

Aspera Sync Admin Guide 3.7.3

AIX

Revision: 3.7.3.142285 Generated: 04/20/2017 12:16

Contents

Introduction.....	3
Overview.....	3
Synchronization and Direction Modes.....	4
Aspera Sync FAQ.....	5
Installation & Configuration.....	7
System Requirements.....	7
Setup.....	7
Configuring Sync.....	8
Viewing Sync Transfers in the Aspera GUI.....	10
Advanced Symbolic Link Options.....	11
Server-Side Symbolic Link Handling.....	11
Multiple Databases (Snap DB).....	12
Running async.....	15
Aspera Sync Command Reference.....	15
Examples of Sync Commands and Output.....	24
Include/Exclude Filtering Rules.....	25
Include/Exclude Filtering Examples.....	28
Bi-directional Example.....	29
Sync to S3 on AWS Cloud Object Storage.....	30
Working with the Aspera Watch Service.....	32
Starting the Aspera Watch Service.....	32
asperawatchd Configuration.....	34
Aspera Watch Service Session Examples.....	36
Logging and Reporting.....	38
asynadmin Command-Line Options.....	38
Logging.....	40
Appendix.....	41
Hardlinks.....	41
Creating SSH Keys.....	41
rsync vs. async Uni-directional Example.....	42
Error Troubleshooting.....	44
Technical Support.....	46
Legal Notice.....	47

Introduction

Overview

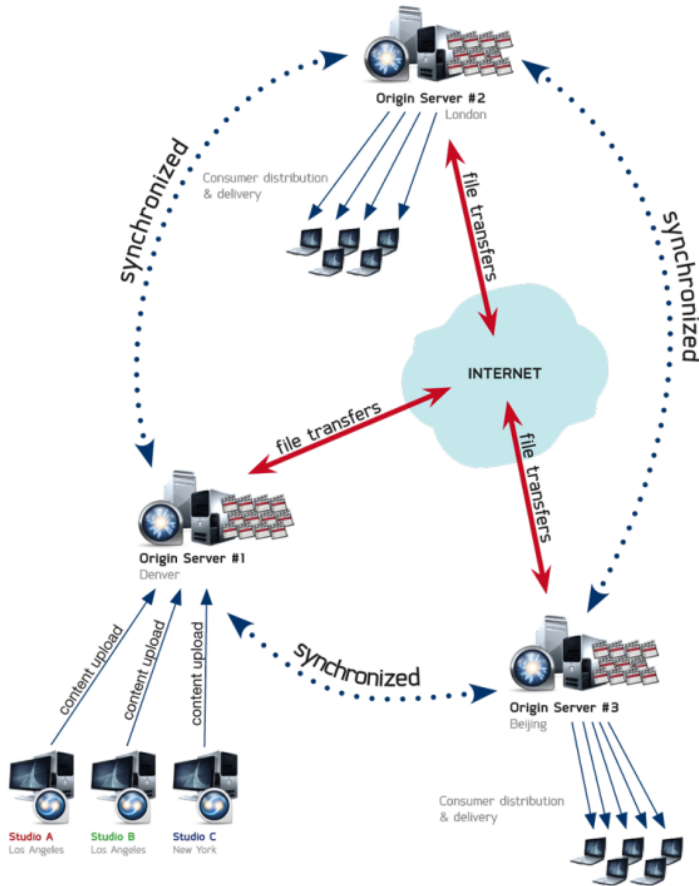
Aspera Sync is a software application that provides high-speed and highly scalable multi-directional, file-based replication and synchronization. Aspera Sync is designed to fill the performance gap of uni-directional file synchronization tools like `rsync`, which are often slow for synchronizing large files and large sets of files over the WAN. Additionally, Aspera Sync extends the capability of uni-directional synchronization tools with full support for synchronization that is bi-directional and multi-directional.

Aspera Sync offers the following key capabilities:

- Utilizes high-speed Aspera FASP transport for moving data at maximum speed over the WAN, whereas traditional synchronization tools are built on TCP. Aspera Sync transfers new data between remote hosts at full bandwidth capacity, regardless of round-trip delay and packet loss, and does not degrade in performance for large file sizes.
- Compares against a local snapshot, thereby avoiding the process of making a comparison against the remote file system over the WAN, which is used by most traditional tools and can be slow.
- Recognizes file system changes (such as moves and renames) and propagates these changes to peers. Traditional tools treat these operations as deletion of old data and then recreate or retransfer the new data, which can lead to costly data copying over the WAN.
- Supports bi-directional and multi-directional synchronization topologies, where files are changing on multiple nodes. For a bi-directional synchronization, Aspera Sync runs with a bi-directional option. For a multi-directional synchronization, one session is run for each peer to remain synced. Any topology that has an acyclic graph topology between peers is supported.
- Uses file system notifications for change notification, when available.
- Monitors file contents and waits for files to be stable (no longer changing in `md5sum`) before transferring. The wait period is configurable and is designed to avoid transferring only partially complete files.

Aspera Sync is a command-line program called `async` that, like `rsync`, uses an SSH connection to establish connectivity with its remote peers and is spawned as an SSH subsystem binary on the remote system. The program can be run one time or periodically (through a `cron` tab scheduled job) on file systems that do not provide asynchronous change notification, or in a continuous mode on file systems that do support asynchronous change notification. Aspera Sync is designed to process files and transfer new data in a continuous pipeline for maximum speed, even when running in scan-only mode (when no file system change notification is available).

Sample Sync Deployment Diagram



Synchronization and Direction Modes

Aspera Sync offers two modes of operation: one-time ("on demand") synchronization and continuous synchronization, as well as three direction modes: uni-directional, bi-directional, and multi-directional.

One-time vs. Continuous Synchronization

One-time synchronization

In this mode, `async` performs synchronization of the endpoints, and exits. If an existing snapshot database exists from a previous run, `async` uses the existing snapshot to determine changes, unless specifically instructed to drop the snapshot and scan the file system again (see the `-x` option in [Aspera Sync Command Reference](#) on page 15).

This mode should be used for `one_time` operations, or for periodic, scheduled synchronizations where file systems do not support event-based change notification. For the latter, `async` can be scheduled as a cron job to run periodically.

One-time synchronization is supported between all operating systems.

Continuous synchronization

In this mode, `async` performs synchronization of the endpoints and continues running. As file system updates occur (i.e., files or directories are added, deleted or modified), `async` detects these changes and synchronizes with the peer endpoint.

Continuous mode is supported only when the file source is Windows or Linux. See the table below for the operating system requirements for the Sync server and client for the different Sync directions.

Continuous Sync Direction	Supported Sync Client OS	Supported Sync Server OS
PUSH	Linux, Windows	All
PULL	All	Linux, Windows
BIDI	Linux, Windows	Linux, Windows

Sync Direction Modes

Uni-directional

Similar to `rsync`, the uni-directional mode supports replication of files and directories, and any updates to these (including deletions, renames, moves, and copies) from a source to a target. The direction of replication can be specified as a "push" or "pull" operation, relative to the initiating host. Once a snapshot is taken after the first replication, all file system updates are recognized against this snapshot, and no comparison of source to target over the WAN is performed (as in `rsync`). Aspera Sync supports most of the same uni-directional synchronization options as `rsync`, such as include/exclude filters, overwrite only if newer, symbolic link handling, and preservation of file system ownership and timestamps.

Bi-directional

Bi-directional mode supports the replication of all file and directory updates between the peers. For any case in which the most recent version of an update cannot be reliably determined, or when a file changes on both endpoints concurrently, Aspera Sync flags the update as a conflict and leaves the peer file systems in their present state (and in conflict). Files in conflict can be reviewed using the `asynadmin` command-line tool (see [asynadmin Command-Line Options](#) on page 38). In this version, it is up to the operator to resolve conflicts manually.

Multi-directional

Multi-directional modes support the synchronization of multiple endpoints, including the following:

- Hub-and-spoke synchronization: A core source (hub) server or cluster replicates to N-number of endpoints (spokes). This configuration can support distributed workflows, content delivery networks, remote- or branch-office replication, and any other scenario requiring concurrent replication to multiple endpoints.
- Collaborative file-based working: Aspera Sync can be used to synchronize remote file stores over the WAN where the file content is being updated concurrently. It is ideal for sync workflows that rely on large unstructured files or file sets, for which traditional caching/compression or de-duplication strategies offer no significant benefit.

In multi-directional scenarios, one Aspera Sync session (one `async` process execution) is required for each remote peer with which synchronization is needed. Any number of `async` processes can be run concurrently, and any number of peers can be synced concurrently. The model assumes that peers are synchronizing are in an acyclic graph topology, meaning that a downstream peer is not configured to sync "back" in a loop to an upstream peer.

Aspera Sync FAQ

What does Aspera Sync actually do?

Aspera Sync synchronizes new and modified files and directories between remote endpoints. It moves, deletes, renames, and transfers new file contents as needed. For example:

- Moving a file out of the synchronized directory results in deletion at the remote peer.
- Moving a file into the synchronized directory results in a copy at the remote peer.
- Renaming a file in a previously synchronized directory renames the file at the remote peer; moving a file in a previously synchronized directory results in the same move operation at the peer.

For additional details on Aspera Sync's core features, see the [Introduction](#) on page 3.

How does Aspera Sync differ from rsync?

Aspera Sync is a high-speed replacement for `rsync` in uni-directional mode, and is designed to be a drop-in replacement with similar command-line options ([async Command-Line Options](#)). Aspera Sync also supports bi-directional and multi-directional synchronization. The following key capabilities distinguish it from `rsync`:

- Uses Aspera's high-speed FASP transport technology, while `rsync` transfers over traditional TCP.
- Operates in push, pull and bi-directional modes.
- Circumvents the typically slower comparison of the local system to the remote system over the WAN, and instead, it efficiently compares the current file system state to a *snapshot* of the last sync.
- Detects and implements file or directory moves and renames to avoid unnecessary transfers over the network.
- Waits for the systems to become stable (that is, it detects whether files are still being modified) before performing synchronization.

To view an example of `async` options versus `rsync` options, see [rsync vs. async Uni-directional Example](#).

How is one-time mode different from continuous mode?

As described in [Synchronization and Direction Modes](#) on page 4, Aspera Sync offers two modes of operation: one-time ("on-demand") synchronization and continuous synchronization. When running in one-time mode, it synchronizes once and exits. In continuous mode, on the other hand, it offers constant synchronization between file systems.

Continuous mode can only be used where file system change notification (that is, *inotify*, which monitors file system events) is available on the systems that are running `async`. NFS-mounted file systems do not support *inotify* change notification for updates made by remote NFS clients, so in these scenarios, `async` should be run in one-time mode (which can be scheduled through cron). The Sync scan mode is designed for maximum speed and is fully pipelined with transfer, so as to allow for maximum performance even in one-time mode.

In what directions does Aspera Sync work?

Sync works in multiple directions: push, pull, and bi-directional. For more information about Sync direction modes, see [Synchronization and Direction Modes](#) on page 4.

- Aspera Sync supports *pushing* files/directories from the local system to a remote system (as does `rsync`), and *pulling* files/directories from a remote system to the local system.
- Bi-directional synchronization occurs between two endpoints, which means that file system changes occurring at either end (local or remote) are replicated on both sides.
- Multi-directional synchronization syncs between multiple endpoints in a hub-and-spoke model or a collaborative model.

How are conflicts handled in bi-directional mode?

A conflict situation can arise in bi-directional mode when a file or directory changes content, an entity is renamed before synchronization has completed, or the change occurs on both endpoints concurrently such that the "newer" version cannot be reliably determined. Aspera Sync reports such conflicts and does not modify either file system, leaving the file systems in conflict. To resolve this type of conflict, delete the file on one side and run `async` again.


How much space is required for an Aspera Sync snapshot?

Snapshots require up to 1 GB of disk space for every 1 million files, and an additional 1 GB for cleanup purposes. For optimum performance, Aspera recommends that the file system have at least 2 GB free per 1 million files, and 3 GB free per 1 million files on Windows (due to the poor performance of Windows NTFS when more than half of the available disk space is occupied).

Installation & Configuration


System Requirements

The following requirements apply to all Aspera Sync endpoints:

- AIX 6.1 or 7.1
 - For all non-Windows environments: libc version GLIB 2.5 or higher
 - Aspera Enterprise Server, Connect Server, Point-to-Point, or Desktop Client version 3.5 or above with a Sync-enabled license. The `async` and `asyncadmin` executables are installed with these products. Updates to Sync are available as patches that can be downloaded from the Aspera website: <http://www.asperasoft.com/en/downloads/23>.
 - Firewall: Allow SSH connections on a non-default, configurable TCP port (for example, TCP/33001).
 - Sync uses the `ascp` binary to perform data transfer. For this reason, you should verify performance and test transfer speeds by running an `ascp` transfer between the endpoints (pushing and/or pulling) that you plan to sync.
-  **Note:** Peer machines must be running compatible versions of Aspera Sync. The 3.5 and newer versions of Sync require peers with Sync 3.5 or newer, or Sync 1.5.6 (bundled with installations of Enterprise/Connect Server, Point-to-Point, and Client 3.4.6).

Setup

IBM Aspera Sync (the `async` and `asyncadmin` executables) is automatically installed with Enterprise Server, Connect Server, Point-to-Point, or Desktop Client version 3.5 or above, but requires a Sync-enabled license to run. IBM Aspera Sync must be installed on both the source and destination machines. For more information, see [System Requirements](#) on page 7. You must have permission to access the directories and files that are to be synchronized using Sync.

-  **Important:** If you attempt to sync a directory without the proper read/write permissions, the directory and files it contains are *not* marked with an ERROR flag in the file directory status output. However, the error will be noted in the log file.

Updates to Sync are available as patches that can be downloaded from the Aspera website. To install the patch, follow the instructions below.

1. Download the update patch from Aspera from the following location:

<http://www.asperasoft.com/en/downloads/23>

If you need help determining your firm's access credentials, contact your Aspera account manager.

2. Extract Sync files and binaries .

```
# tar -xvf aspera-sync-version-operating_system.tar
```

Replace the existing files and binaries in `/opt/aspera` with the extracted files.

3. Verify the `async` and `asyncadmin` installations.

Run the following commands in a terminal window:

```
# async -A
# asyncadmin -A
```

Messages similar to the following messages appear if installation was successful:

```
# async -A                                <!-- Check Aspera product license -->
Aspera Enterprise Server version version
async version version
Operating System: platform
FIPS 140-2-validated crypto ready to configure
License max rate=(unlimited), account no.=1, license no.=12345. Expiration
date: expiration_date

# asyncadmin -A                            <!-- Display software version -->
Aspera asyncadmin version version
```

Configuring Sync

Many parameters of Sync can be configured by using the `asconfigurator` tool from the command line or by editing the `file_system` and `default` sections of `aspera.conf`. To configure Sync using `asconfigurator`, use this general syntax:

```
# /opt/aspera/bin/asconfigurator -x "set_node_data;option,value"
```

Possible options and values are described below.

To manually edit `aspera.conf`, open it in a text editor with administrative privileges.

```
/opt/aspera/etc/aspera.conf
```

See an example of the `asperawatchd` configuration in `aspera.conf` below.

You must validate `aspera.conf` after manually editing it to confirm that your configuration is correct. Run the following command:

```
# /opt/aspera/bin/asuserdata -v
```

To view the current settings, run the following command:

```
# /opt/aspera/bin/asuserdata -a
```

Sync Configuration Options

The table below provides a reference of many of the configuration options for use with Sync.

asconfigurator option aspera.conf setting	Description and Value Options
directory_create_mode <directory_create_mode>	Specify the 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. This option is applied only when the server is a Unix-based receiver. Value is a positive integer (octal). (Default: undefined)
directory_create_grant_mask <directory_create_grant_mask>	Specify the mode for newly created directories if <code>directory_create_mode</code> is not specified. If specified, directory modes are set to their original modes plus the grant mask values. This option is applied only when the server is a Unix-based receiver and when <code>directory_create_mode</code> is not specified.

asconfigurator option aspera.conf setting	Description and Value Options
	Value is a positive integer (octal). (Default: 755)
<pre>preserve_acls preserve_xattrs <preserve_acls> <preserve_xattrs></pre>	<p>Specify if the ACL access data (<code>acls</code>) or extended attributes (<code>xattrs</code>) from Windows or Mac OS X files are preserved. Three modes are supported. (Default: none)</p> <p><code>native</code>: <code>acls</code> or <code>xattrs</code> are preserved by using the native capabilities of the file system. If the destination does not support <code>acls</code> or <code>xattrs</code>, <code>async</code> generates an error and exits.</p> <p><code>metafile</code>: <code>acls</code> or <code>xattrs</code> are preserved in a separate file. The file is in the same location and has same name, but has the added extension <code>.aspera-meta</code>. The <code>.aspera-meta</code> files are platform-independent, and files can be reverted to native form if they are synced with a compatible system.</p> <p><code>none</code>: No <code>acls</code> or <code>xattrs</code> data is preserved. This mode is supported on all file systems.</p> <p>ACL preservation is only meaningful if both hosts are in the same domain. If a SID (security ID) in a source file does not exist at a destination, the sync proceeds but no ACL data is saved and the log records that the ACL was not applied.</p> <p>The <code>aspera.conf</code> settings for <code>acls</code> or <code>xattrs</code> can be overwritten by using the <code>--preserve-acls</code> or <code>--preserve-xattrs</code> options, respectively, in a command-line <code>async</code> session.</p>
<pre>async_db_spec <async_db_spec></pre>	<p>Value has the syntax <code>sqlite:lock_style:storage_style</code>. (Default: undefined)</p> <p><code>lock_style</code>: Specify how <code>async</code> interfaces with the operating system. Values depend on operating system.</p> <p>Unix-based systems have the following options:</p> <ul style="list-style-type: none"> • <code>empty</code> or <code>unix</code>: The default method that is used by most applications. • <code>unix-flock</code>: For file systems that do not support POSIX locking style. • <code>unix-dotfile</code>: For file systems that do not support POSIX nor flock-locking styles. • <code>unix-none</code>: No database-locking mechanism is used. Allowing a single database to be accessed by multiple clients is not safe with this option. <p><code>storage_style</code>: Specify where Sync stores a local database that traces each directory and file. Three values can be used:</p> <ul style="list-style-type: none"> • <code>undefined</code> or <code>disk</code>: The default option. The database reads and writes to disk. This provides maximum reliability and no limitations on the number of files that can be synced. • <code>lms</code>: The database is loaded from disk to memory at startup, changes during the session are saved to memory, and the database is saved to disk on exit. This option increases speed but all changes are lost if <code>async</code> stops abruptly, and the number of synced files is limited by memory availability.

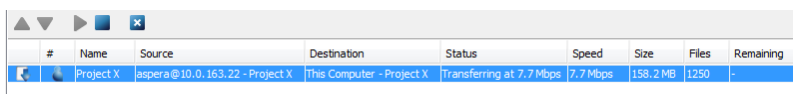
asconfigurator option aspera.conf setting	Description and Value Options
	<ul style="list-style-type: none"> memory: The database is stored completely in memory. This method provides maximum speed but is not reliable because the database is not backed up to disk.
async_connection_timeout <async_connection_timeout>	<p>The number of seconds <code>async</code> waits for a connection to be established before it terminates.</p> <p>Value is a positive integer. (Default: 20)</p>
async_session_timeout <async_session_timeout>	<p>The number of seconds <code>async</code> waits for a non-responsive session to resume before it terminates.</p> <p>Value is a positive integer. (Default: 20)</p>
async_log_dir <async_log_dir>	<p>Specify an alternative location for the <code>async</code> server's log files. If unspecified, log files are saved in the default location or the location that is specified by the client with the <code>-R</code> option. For information on the default log file location, see Logging on page 40.</p>
async_log_level <async_log_level>	<p>Set the amount of detail in the <code>async</code> server activity log. Valid values are <code>disable</code>, <code>log</code> (default), <code>dbg1</code>, or <code>dbg2</code>.</p>

Example Sync Configuration in aspera.conf

```
<file_system>
...
<directory_create_mode> </directory_create_mode>
<directory_create_grant_mask>755</directory_create_grant_mask>
<preserve_acls>none</preserve_acls>
<preserve_xattrs>none</preserve_xattrs>
...
</file_system>
...
<default>
...
<async_db_spec> </async_db_spec>
<async_connection_timeout>20</async_connection_timeout>
<async_session_timeout>20</async_session_timeout>
<async_log_dir>AS_NULL</async_log_dir>
<async_log_level>log</async_log_level>
...
</default>
```

Viewing Sync Transfers in the Aspera GUI

The Aspera Enterprise Server, Point-to-Point Client, and Desktop Client GUI shows `async`-initiated transfers if Sync is run on the machine (as client) by default, whereas server `async` transfers are not shown. In the example below, transfers associated with a Sync job in which the remote user, `aspera`, is pushing files to the server folder for Project X.



#	Name	Source	Destination	Status	Speed	Size	Files	Remaining
1	Project X	aspera@10.0.163.22 - Project X	This Computer - Project X	Transferring at 7.7 Mbps	7.7 Mbps	158.2 MB	1250	-

You can configure the server and client reporting to the Aspera GUI with the following options.

Server reporting:

Server reporting is disabled by default. To enable the server to report Sync-initiated transfers, run the following command on the server:

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

Restart the Aspera NodeD service to activate the change. Run the following commands:

```
$ /etc/rc.d/init.d/asperanoded restart
```

Client reporting:

Client reporting is enabled by default. To disable the client from reporting Sync-initiated transfers, run the following command on the client machine:

```
# asconfigurator -x
"set_client_data;async_management_activity_logging,false"
```

You do not need to restart the Aspera NodeD service for this change to take effect.

Advanced Symbolic Link Options

Client-side handling of symbolic links is configured from the following IBM Aspera Sync command line:

```
SHORT-FORM OPTION: # async -n option
LONG-FORM OPTION: # async --symbolic-links=option
```

The following section describes the possible configuration options:

Configuration Options

Option	Description
copy	Create (or update) the link at the destination. (Not valid for Windows source or destination.) This is the default option.
follow	Follow symbolic links and transfer the linked files. (Skip, if the source is Windows.)
skip	Ignore the symbolic link.

Server-Side Symbolic Link Handling


Aspera handles symbolic links in IBM Aspera Sync based on settings configured in the `aspera.conf` file, found in the following location:

```
/opt/aspera/etc/aspera.conf
```

Configuration Options

The following configuration options are set in the `<file_system>` section of the `aspera.conf` file:

```
<file_system>
  <symbolic_links>comma-separated_options</symbolic_links>
</file_system>
```

 **Note:** If no option is specified, the configuration defaults to create, follow.

Option	Description	Client Behavior	Server Behavior
create	Create symbolic links with arbitrary targets. This is the default.	Disallowed if not configured. Note: Servers running on a Windows platform will always reject such a request, regardless of the option configured.	Symbolic links are always copied to the server if the client requests.
follow	Follow symbolic links with targets inside docroot. If at any point the path goes outside the docroot, Aspera Sync will not complete the transfer. This is option set by default.	Symbolic links are always copied to the server if the client requests. Note: If the docroot is a symbolic link and is specified as the source or destination: As the receiver, follow the target widely (no docroot constraint) and unconditionally, regardless of the symbolic link actions that are configured/requested.	
follow_wild	Follow symbolic links with arbitrary targets, even if the targets are outside the docroot.		Disallowed if not configured, except for Windows servers. Follow symbolic links anywhere without docroot constraint. Note: If the docroot is a symbolic link and is specified as the source or destination: As sender, follow the target unconditionally, regardless of the symbolic link actions that are configured/requested.
none	Take no action with symbolic links.	Take no action with symbolic links.	Take no action with symbolic links.

Multiple Databases (Snap DB)

The database directory is specified from the command line using `-b` (local) and `-B` (remote). Snap DB (snapshot database) does not work on CIFS, NFS, or other shared file systems mounted on Linux; therefore, `-B` and `-b` must specify a directory on a file system physically local to the endpoint host.

`async` creates a private directory (`.private-asp`) to store the Snap DB and in-progress transfers (before the files get moved to their final location). This produces a `.private-asp` directory in the sync directory (used for transfer cache) and a `.private-asp` directory in the `-b/-B` directory that stores the Snap DB.

Multiple sessions can sync the same directory or specify the same Snap DB directory (`-b/-B`), so `async` creates a subdirectory in `.private-asp` for each session (with the name specified by `-N`). To allow the session name to be used as a directory name, names can only use standard alphanumeric characters and `"_"` and `"-"` characters. This subdirectory is created on both sides, if needed, depending on the direction of the transfer and Snap DB location (see below for details).

Each `async` session must have a unique name. If multiple sessions sync the same directory or specify the same Snap DB directory (`-b/-B`), then the session names *must* be unique. Snap DB records the session name and the sync directories (local and remote). If you initiate a second `async` session with the same name, using the same sync

directory on one end but a different directory on the other end (or both directories are different, but the database location is the same), `async` will fail.

Example 1: Bi-directional `async`

```
# -N ex1 -b /var/db -B /opt/aspera/var -d /data/users -r root@server:/
storage/users -K bidi
```

The above command creates the following:

On the local computer (initiator):

- `/var/db/.private-asp/ex1/snap.db`
- `/data/users/.private-asp/ex1` (for transfer cache)

On the remote computer (reactor):

- `/opt/aspera/var/.private-asp/ex1/snap.db`
- `/storage/users/ex1` (for transfer cache)

Example 2: Uni-directional `async`

```
# -N ex2 -b /var/db -B /opt/aspera/var -d /data/users -r root@server:/
storage/users -K push
```

The above command creates the following:

On the local computer (initiator):

`/var/db/.private-asp/ex2/snap.db`

On the remote computer (reactor):

`/opt/aspera/var/.private-asp/ex2/snap.db`

`/storage/users/ex2` (for transfer cache)

Changing Direction Between Runs

Changing direction between runs is not allowed. `async` will fail with an error message and you must run it with `--reset` or provide a new database directory.

Starting up when a snapshot DB is missing

An `async` session cannot start when only one of the Snap DBs is present. If this is the case, you must either clean the Snap DB or start `async` with `--reset`. The Snap DB is located under the regular private directory by default, although its location can also be specified by startup parameters. Behavior with respect to presence of the private directory and Snap DB file are as follows:

On the client:

1. If the private directory doesn't exist, create it.
2. If the database directory doesn't exist (meaning the database doesn't exist), create both, and remember "there was no DB."
3. If the database file doesn't exist, create it and remember "there was no DB."
4. Tell the remote whether the DB existed.

On the remote:

1. Same as client 1, 2, and 3.
2. Determine if "there was no DB."

If there is a discrepancy (local Snap DB exists, but no remote; or, remote exists but no local), tell the client there is an error. If a subsequent run specifies different DB directories, the above behavior is preserved. In other words, even if a

Snap DB exists in the regular private directory or elsewhere, it assumes "there was no DB" if there isn't a database in the specified directory.

Deleting the snapshot DB during synchronization results in undefined behavior

To recover, stop `async`, delete the DB on the other side as well, and restart.

Running async

Aspera Sync Command Reference

Basic syntax:

```
# async [instance_options] -N pair -d ldir -r [user@host:rdir]
   [session_options] ...
```

- 👉 **Note:** Transfers started by `async` can be controlled from the Enterprise Server or Connect Server GUI. Canceling an `async` transfer from GUI shuts down `async`.

Required Command Options

Naming the async session: `-N pair`

`-N pair` is required in `async` commands. The value for `pair` is a name that uniquely identifies the Sync session and is visible in Aspera Console. `-N pair` must follow any instance options and must precede all session arguments. Names can only use standard alphanumeric characters, plus "_" and "-" characters.

Specifying filepaths and filenames: `ldir` and `rdir`

`ldir` specifies the local directory to be synchronized and `rdir` specifies the remote directory to be synchronized. File paths and filenames must follow these rules:

- The drive letter is required in Windows paths, unless the server's `aspera.conf` file has a `docroot` defined for the user. If no drive letter is included when syncing with a Windows machine and `docroot` is not defined for the user, `async` displays the error message: "Failed. Peer error: Remote directory is not absolute."
- You can synchronize Windows, Linux, Mac OS X, and other Unix-based endpoints and servers, but must take care with path separators. The path separator "/" is supported on Windows and other platforms. The path separator "\" is platform-agnostic *only* for the options `-d/r/L/R/B/b` and `--keep-dir-local/remote`. In [Include/Exclude Filtering Rules](#) on page 25, however, "\" is exclusively a quoting operator and "/" is the only path separator recognized.
- File names may not contain `\n`, `\r`, or `\.`. Files with these in their names are skipped.
- When scanning or monitoring a file system for changes, `async` skips over files with names that end in one of the special suffixes specified in `aspera.conf` with `<resume_suffix>` and `<partial_file_suffix>`. To disable this behavior, you can set these values to the empty string. `<resume_suffix>` defaults to `.aspx`. The `<partial_file_suffix>` tag defaults to the empty string, but is often set to `.partial`.

Specifying the direction of the sync: `-K direction`

Sync has three modes of synchronization: `push`, `pull`, and `bidirectional`.

- `push`: The contents of `ldir` are synchronized to `rdir`, with the `ldir` content overwriting the `rdir` content, by default (unless the overwrite options are specified otherwise, e.g. to only overwrite if newer, or never overwrite).
- `pull`: The contents of `rdir` are synchronized to `ldir`.
- `bidirectional` (bi-directional): The contents of `ldir` and `rdir` are synchronized, with newer versions of files and directories overwriting older versions in either `ldir` or `rdir`, by default.

Using continuous mode: `-C`

Continuous mode is supported only when the file source is Windows or Linux. See the table below for the operating system requirements for the Sync server and client for the different Sync directions.

Continuous Sync Direction	Supported Sync Client OS	Supported Sync Server OS
PUSH	Linux, Windows	All
PULL	All	Linux, Windows
BIDI	Linux, Windows	Linux, Windows

One-time synchronization is supported between all operating systems.

The following tables are complete command-line options references. View an abbreviated version from the command line by running:

```
# async -h
```


For examples of `async` commands and output, see [Examples of Sync Commands and Output](#) on page 24.

Instance Options



Option (- Short form / -- Long form)	Description
-A --version	Display the <code>async</code> version information and license information.
-h --help	Display help for command-line options.
-D[D..]	Debug level. Default is 0.
-q --quiet	Disable progress display.
-L <i>log_dir</i> --alt-logdir= <i>log_dir</i>	Specify a logging directory on the local host. If the directory doesn't exist, <code>async</code> creates it for you.
--watchd= <i>datastore:host:port[:domain]</i>	Use <code>asperawatchd</code> connected to the specified Redis for the transfer session. <code>redis</code> is the only accepted <i>datastore</i> for this option. For example: <pre>--watchd=redis:localhost:31415</pre> The optional <i>domain</i> argument allows you to specify if the domain is other than the default root. For more information see Working with the Aspera Watch Service on page 32.
--apply-local-docroot	Prepend the local <code>docroot</code> to the local directory.


Session Options


Option	Description
-N <i>pair</i>	Name for the synchronization pair, which can be any string used to identify the connection. This value is stored in the session cookie and can be used in

Option	Description
<pre>--name=pair</pre>	<p>Aspera Console to identify the transfer session's name. It can contain only ASCII alphanumeric, hyphen, and underscore characters.</p> <p>-N must precede any combination of the options below.</p>
<pre>-d ldir --local-dir=ldir</pre>	<p>Set the local sync directory to <i>ldir</i>. You can use <code>--create-dir</code> (described below) to create the remote directory if it does not already exist.</p>
<pre>-r rdir --remote-dir=rdir</pre>	<p>Set the remote sync directory to <i>rdir</i>, where <i>rdir</i>=[<i>user@host:path</i>]. You can use the <code>--create-dir</code> option (described below) to create the remote directory if it does not already exist.</p> <p> Caution: For a local synchronization, do not specify a remote (destination) directory that is located inside your source directory.</p>
<pre>--host=host</pre>	<p>The name of the remote host. When <i>host</i> is specified with this option, the characters "@" and ":" are no longer treated specially in the argument to <code>-r</code>, so may appear in any such name. If <code>--host=host</code> is specified, the remote username cannot be extracted from the argument for <code>--remote-dir</code> and must be supplied by a <code>--user=name</code> option or in the environment variable <code>\$user</code> (on Windows, <code>%USER%</code>). Allowed forms are as follows:</p> <pre>--remote-dir user@host:/rootdir # (old method) --user user --remote-dir host:/rootdir --host host --user user --remote-dir /rootdir --remote-dir host:/rootdir # (uses \$user) --host host --remote-dir /rootdir # (uses \$user)</pre> <p>The following means the same as the first three lines above:</p> <pre>-r /rootdir --user=user --host=host</pre> <p>For backward compatibility, <code>-r A:/rootdir</code> for any single letter <i>A</i> is still taken as a Windows path, not as <code>--host A -r /rootdir</code>. To specify a one-letter host name <i>A</i>, use an explicit <code>--host=A</code>.</p>
<pre>--user=user</pre>	<p>The name of the remote user. When <code>--user=user</code> is specified, the character "@" is no longer treated specially in the argument to <code>-r</code>, so may appear in any such name.</p>
<pre>-w pass --pass=pass</pre>	<p>Set the passphrase string to the user's password.</p>
<pre>-W token_string --token=token_string</pre>	<p>Set the authorization token. The token type (sync-push, sync-pull, or sync-bidi) must match the direction (push, pull, or bidi) of the requested transfer. The token path must match the remote directory of the requested transfer. If an invalid token is provided, the requested transfer will be denied.</p>
<pre>-i file --private-key-path=file</pre>	<p>The SSH private key file. For information on creating a key pair, see Creating SSH Keys on page 41.</p>
<pre>--check-sshfp=fingerprint</pre>	<p>Compare <i>fingerprint</i> to remote host key hash, fail on mismatch.</p>
<pre>-K direction --direction=direction</pre>	<p>Set the transfer direction. <i>direction</i> can be push, pull, or bidi (bi-directional). (Default: push)</p>

Option	Description
-k <i>type</i> --checksum= <i>type</i>	Set the checksum type. <i>type</i> can be sha1, md5, sha1-sparse, md5-sparse, or none. A value of none is equivalent to a size check only and <code>async</code> will not detect a change in timestamp. (Default: sha1-sparse for local storage, none for object storage)
-P <i>port</i> --tcp-port= <i>port</i>	Set the TCP port for SSH. <i>port</i> must be valid numeric IP port. (Default: 22)
-O <i>port</i> --udp-port= <i>policy</i>	Set the UDP port used by FASP for data transfer. (Default: 33001)
-a <i>policy</i> --rate-policy= <i>policy</i>	Set the transfer rate policy. <i>policy</i> can be fixed, fair, high, or low. (Default: fair)
-l <i>rate</i> --target-rate= <i>rate</i>	Set the maximum transfer rate. <i>rate</i> =integer G/g, M/m, K/k, or bps. (Default: 10 Mbps)
-m <i>rate</i> --min-rate= <i>rate</i>	Set min transfer rate. <i>rate</i> =integer G/g, M/m, K/k, or just bps. (Default: 200 Kbps)
-g <i>size</i> --read-block-size= <i>size</i>	Set block size for reading. <i>size</i> =integer K, M, or just bytes. (Default: 64MB)
-G <i>size</i> --write-block-size= <i>size</i>	Set block size using for writing. <i>size</i> =integer K, M, or just bytes. (Default: 64MB)
-H <i>val</i> --scan-intensity= <i>val</i>	Set the intensity for scanning work. <i>val</i> can be one of vlow, low, medium, high, or vhigh. vlow minimizes system activity. vhigh maximizes system activity by continuously scanning files without rest. (Default: medium)
-R <i>rem_log_dir</i> --remote-logdir= <i>rem_log_dir</i>	Specify a logging directory on the remote host.
-Z <i>mtu</i> --datagram-size= <i>mtu</i>	Specify the datagram size (<i>mtu</i>) as an integer. (Default: detected-path MTU)
-X <i>size</i> --rexmsg-size= <i>size</i>	Set the <i>size</i> (in bytes) of a retransmission request (Maximum: 1440).
-c <i>cipher</i> --cipher= <i>cipher</i>	Set the encryption algorithm. <i>cipher</i> can be none, AES128, AES192, or AES256. (Default: AES128)
-b <i>ldbdir</i> --local-db-dir= <i>ldbdir</i>	Local database directory. (Default: <code>.private-asp</code> at the root level of the synchronized directory) You may relocate the snapshot database to a different location than the default one under the <i>ldir</i> specified with <code>-d</code> . This allows placing the database away from the main data files. This is useful for performance tuning. It is also useful when <code>-d ldir</code> is located on a network share

Option	Description
	volume that does not reliably support database locking. For further usage information, see Multiple Databases (Snap DB) on page 12.
<pre>-B rdbdir --remote-db-dir=rdbdir</pre>	Remote database directory. Similar to <code>-b</code> above, but for the remote database. For further usage information, see Multiple Databases (Snap DB) on page 12. (Default: <code>.private-asp</code> at the root level of the synchronized directory)
<pre>-I file --include-from=file</pre>	Scan and include paths specified in the filter <i>file</i> . For more information on setting filters, see Include/Exclude Filtering Rules on page 25.
<pre>-E file --exclude-from=file</pre>	Skip paths specified in the filter <i>file</i> . For more information on setting filters, see Include/Exclude Filtering Rules on page 25.
<pre>--include=pattern --exclude=pattern</pre>	Include and exclude paths that match <i>pattern</i> . For more information on how to set include and exclude patterns, see Include/Exclude Filtering Rules on page 25.
<pre>--exclude-dirs-older- than=MTIME</pre>	<p>After the initial Sync scan, do not scan directories during subsequent syncs if they or their parents have a recursive modified time older than the specified value. The recursive modified time of a directory is the most recent modification time of it or any of its children (file or directory). This option is used to avoid rescanning directories that are known to have not changed since the previous Sync, such as a monthly archive directory structure in which only the most recent subdirectory is being modified.</p> <p><i>MTIME</i> may be specified in any one of the following ways:</p> <ul style="list-style-type: none"> As a positive number of seconds since 1970-01-01 00:00:00, for Unix and POSIX-compliant operating systems. <p> Note: Some file servers, such as Windows NT, use a different epoch for the recursive modified time. In this case, <i>MTIME</i> should be specified as a duration relative to present or UTC timestamp.</p> <ul style="list-style-type: none"> As a UTC timestamp with the format YYYY-MM-DDTHH:MM:SS, such as 2015-01-01T08:00:00. As a duration formatted as DDd HH:MM:SS or WWw DDd HHh MMm SSs. Directories whose "mtime" is older than Now minus <i>MTIME</i> are not scanned. Input requirements: Leading zero fields and spaces may be omitted. The leftmost fields are optional, but fields to the right of the largest unit specified are required. For example, to exclude directories older than 24 hours, you could specify 1d 0:0:0, 24:00:00, or 24h 00m 00s, but not 1d. <p>This option does not apply to the root directory.</p> <p> Note: Sync fails if the first run of <code>async</code> and the next run do not use the same <code>--exclude-dirs-older-than</code> option. If the first run specifies <code>--exclude-dirs-older-than</code>, then the next run must use this option, too. If the first run does not include <code>--exclude-dirs-older-than</code>, then the next run fails if this option is specified.</p>
<pre>-n action --symbolic-links=action</pre>	Treat symbolic links per the specified <i>action</i> . Permitted values are:

Option	Description
	<p><code>copy</code> - create (or update) the link at the destination. (Not valid for Windows source or destination.)</p> <p><code>skip</code> - ignore the link altogether.</p> <p><code>follow</code> - treat the link as if it were the file (or directory) it points to, so that at the destination, what was a link will now be a copy of the file (or directory). (Functions as <code>skip</code> if source is Windows.)</p> <p>(Default: <code>copy</code>)</p>
<p><code>-C</code> <code>--continuous</code></p>	<p>Run continuous synchronization. (Default: disabled)</p> <p> Note: Continuous mode is supported only when the file source is Windows or Linux. Continuous pulls can be run from any operating system if the source is Windows or Linux. Continuous push can be run only by Windows or Linux. Continuous bidi requires that both the Sync client and server are Windows or Linux.</p> <p>If you receive an <code>inotify</code> error when attempting to run continuous synchronization, see Error Troubleshooting on page 44.</p> <p>If a file is open, <code>async</code> cannot transfer the file due to sharing violations and may ignore the file if it is closed without changes. To specify the maximum number of retries after a sharing violation, use the <code>--sharing-retry-max=N</code> option. To enable periodic scans that detect when an opened file has been closed and can be transferred, use the <code>--scan-interval=duration</code> option.</p>
<p><code>--scan-interval=duration</code></p>	<p>Enable periodic scans during a continuous Sync (a session run with the <code>-C</code> option). <i>duration</i> defines the interval between periodic scans. <i>duration</i> may be specified as DDd HH:MM:SS.mmm or WWw DDd HHh MMm SSs XXms XXus. Leading zero fields may be omitted. Spaces may be omitted. A plain number XX will be interpreted as SSs (seconds).</p>
<p><code>--remote-scan-interval=duration</code></p>	<p>Like <code>--scan-interval</code> but scanning the remote machine.</p>
<p><code>-t</code> <code>--preserve-time</code></p>	<p>Preserve file timestamps. (Default: disabled)</p>
<p><code>--preserve-access-time</code></p>	<p>Set the access time of the destination file to the same value as that of the source file. (Default: disabled)</p>
<p><code>--preserve-modification-time</code></p>	<p>Set the modification time of the destination file to the same value as that of the source file. (Default: disabled)</p>
<p><code>--preserve-creation-time</code></p>	<p>Set the creation time of the destination file to the same value as that of the source file. Only valid on Windows machines. (Default: disabled)</p>
<p><code>-u</code> <code>--preserve-uid</code> <code>-j</code> <code>--preserve-gid</code></p>	<p>Preserve file owner's <i>uid</i> or <i>gid</i>. This preserve option requires that <code>async</code> is running as root. (Default: disabled)</p>
<p><code>--write-uid=uid</code> <code>--write-gid=gid</code></p>	<p>Write files as the user <i>uid</i> or the group <i>gid</i>. <i>uid</i> and <i>gid</i> may be numeric, or by name. If by name, the name is looked up on the host actually performing the write. Failure to set the <i>uid</i> or <i>gid</i> is logged, but is not an</p>

Option	Description
	<p>error. The <i>uid</i> or <i>gid</i> is set after <i>ascp</i> completes and before moving the file from the staging directory to the final location.</p> <p><code>--write-uid=<i>uid</i></code> conflicts with <code>--preserve-uid</code> and <code>--write-gid=<i>gid</i></code> conflicts with <code>--preserve-gid</code>.</p>
<code>--scan-dir-rename</code>	Enable detection of renamed directories and files on the initial scan. Results are only relevant if the nodes for your file system are stable.
<code>--scan-file-rename</code>	Enable detection of renamed files on the initial scan. Results are only relevant if the nodes for your file system are stable.
<code>-o <i>policy</i></code> <code>--overwrite=<i>policy</i></code>	<p>Set the overwrite policy to <code>always</code>, <code>older</code>, or <code>conflict</code>. Use with <code>-K push</code> and <code>pull</code>. (Default: <code>always</code> for <code>-K push</code> and <code>pull</code>; <code>conflict</code> for <code>-K bidi</code>)</p> <p> Note: When syncing with object storage, only file size (<code>--checksum=none</code>) can be used to compare files. Thus, using <code>--overwrite=always</code> only overwrites files whose sizes have changed. If the content of a local file is different from a file with the same name in object storage but the files are the same size, the file in object storage is not overwritten. To overwrite files in this case, use <code>--overwrite=older</code>.</p> <p><code>--overwrite=older</code> is only accurate if the user also specifies <code>--preserve-time</code> (preserve timestamps).</p> <p>To resolve <code>conflict</code> and <code>error</code> situations in a uni-directional sync, “touch” the problem files on the source and run <i>async</i> with <code>--overwrite=always</code>. This clears all <code>conflict</code> and <code>error</code> states as the problem files are synced.</p>
<code>-x</code> <code>--reset</code>	Clear the <i>async</i> snapshot database and rescan the synchronized directories and files to create a fresh snapshot. (Default: disabled)
<code>--assume-no-mods</code>	Assume the directory structure has not been modified. If a directory's modification time has not changed compared to the last snapshot, <i>async</i> in non-continuous mode skips scanning the directory. This option speeds up scanning static directory structures.
<code>--ignore-delete</code>	Do not copy removals to the peer. This option is used mostly with uni-directional syncs. In bi-directional sync, a deletion on one side is ignored but the next time <i>async</i> is run, the file is recopied from the other end. In continuous mode, the file is not recopied until either <i>async</i> is restarted or the file is changed (<i>touched</i>).
<code>--delete-delay</code>	Postpone the actual deletion of files or directories until the end of the sync session.
<code>--keep-dir-local=<i>dir</i></code>	Move deleted files into <i>dir</i> . Note that <i>dir</i> must exist (it is not created by <code>--create-dir</code>), and must be outside the synchronization directory (or excluded from the sync using <code>--exclude</code> or <code>--exclude-from</code>), but on the same file system.
<code>--keep-dir-remote=<i>dir</i></code>	Move the server's deleted files into <i>dir</i> . Note that <i>dir</i> must exist (it is not created by <code>--create-dir</code>), and must be outside the synchronization directory (or excluded from the sync using <code>--exclude</code> or <code>--exclude-from</code>), but on the same file system.

Option	Description
<code>--local-mount-signature=signature file</code>	Verify that the local file system is mounted by the existence of this file. Use of this feature increases the time to synchronize deletes.
<code>--remote-mount-signature=signature file</code>	Verify that the remote file system is mounted by the existence of this file. Use of this feature increases the time to synchronize deletes.
<code>--create-dir</code>	Create the remote directory if it does not exist. This option is used with the <code>-d</code> and <code>-r</code> options (<code>async</code> creates directories if they do not exist, rather than reporting an error and quitting).
<code>--dedup [=mode]</code>	<p>Specify the action to take when Sync detects duplicate files on the source, even if they have different pathnames. This option requires specifying the checksum type with the <code>-k</code> option.</p> <p>Available modes are <code>hardlink</code>, <code>inode</code>, or <code>copy</code> (Default: <code>hardlink</code>).</p> <ul style="list-style-type: none"> <code>hardlink</code> - When two or more source files are duplicates, a hardlink is created between them on the target. This saves storage by preventing multiple copies of the same file from accumulating on the target. The files on the target have the same inode, even if the source files have different inodes. The target must be running a Unix-based operating system. <code>inode</code> - When two or more source files have matching inodes, a hardlink is created between them on the target and the target files have matching inodes. This is only supported between Unix-based platforms. If <code>--dedup=inode</code> is used in a continuous sync, Aspera recommends using the <code>scan-interval</code> option. <code>copy</code> - After a file is synced on the target, the synced file is copied to the duplicate. This saves bandwidth by not transferring duplicate files. This mode is useful when the target is Windows. <p>Without the dedup option, all duplicate files are synced.</p> <p>Duplicates may still be synced, rather than hardlinked or copied, if one of the duplicates has not yet been synced on the target.</p>
<code>--no-scan</code>	Skip initial scanning.
<code>--cooloff=sec</code>	Number of seconds to delay the start of the transfer. For example, if <code>--cooloff=5</code> , <code>async</code> waits 5 seconds before copying a file. If <code>--cooloff=0</code> transfers start immediately. Note that both peers get the same cooloff period. The permitted value for <code>sec</code> is any integer from 0 to 60. (Default: 3)
<code>--pending-max=N</code>	This option acts as a buffer to ensure that files to be transferred do not exceed the maximum. (Default: 2000)
<code>--sharing-retry-max=N</code>	Maximum number of times to retry after a sharing violation. The interval between retries is the number of seconds specified by <code>--cooloff</code> . (Default: 3) Not supported
<code>--preserve-acls=mode</code>	<p>Preserve the ACL access data from Windows or OS X files if the preservation <code>mode</code> is <code>native</code> or <code>metafile</code>. (Default: <code>none</code>)</p> <p>If <code>mode</code> is <code>native</code>, <code>async</code> expects that the destination supports the same native ACL format as the source; if not, it generates an error and exits.</p> <p>If <code>mode</code> is <code>metafile</code>, <code>async</code> writes the access data to a separate file in the same location; the file has same name but with the added extension <code>.aspera-meta</code>. Metafile data is in a generic format that can reside on any</p>

Option	Description
	<p>platform and also be reconverted to native form if the file is again synced with a system that supports it.</p> <p>If <i>mode</i> is <i>none</i>, no ACL data is preserved at all.</p> <p>This feature is only meaningful if both hosts are in a common security domain. If a SID (security ID) in a source file does not exist at a destination, the sync proceeds but no ACL data is saved and the log records that the ACL could not be applied.</p> <p>Note: Both <code>--preserve-acls</code> and <code>--remote-preserve-acls</code> must be specified in order for the target side of the pull to apply the <code>acls</code>.</p>
<code>--preserve-xattrs=mode</code>	<p>Preserve OS X extended attributes data (xattr) if preservation <i>mode</i> is <i>native</i> or <i>metafile</i>. (Default: <i>none</i>)</p> <p>If the sync is run by a regular user, only user-level attributes are preserved. If run as superuser, all attributes are preserved.</p> <p>If <i>mode</i> is <i>native</i>, <code>async</code> expects that the destination supports the same native xattr format as the source; if not, it generates an error and exits.</p> <p>If <i>mode</i> is <i>metafile</i>, <code>async</code> writes the xattr data to a separate file in the same location; the file has the same name with the added extension <code>.aspera-meta</code>. The <code>aspera-meta</code> files are platform-independent and can be copied between hosts without loss of data. They can be reconverted to the native form if the file is again synced with an OS X system.</p> <p>If <i>mode</i> is <i>none</i>, no xattr data is preserved at all.</p>
<code>--remote-preserve-acls=mode</code>	<p>Like <code>--preserve-acls</code> but used when ACLs are stored in a different format on the remote host. Defaults to the value of <code>--preserve-acls</code>.</p> <p>Note: Both <code>--preserve-acls</code> and <code>--remote-preserve-acls</code> must be specified in order for the target side of the pull to apply the <code>acls</code>.</p>
<code>--remote-preserve-xattrs=mode</code>	<p>Like <code>--preserve-xattrs</code> but used attributes are stored in a different format on the remote system. Defaults to the value of <code>--preserve-xattrs</code>.</p>
<code>--compression=mode</code>	<p>Specify method used to compress a file before transfer. <i>mode</i> can be <code>zlib</code> or <code>none</code>. The default value is <code>none</code>.</p>
<code>--transfer-threads=N[:size]</code>	<p>Specify intervals for dedicated transfer threads. <i>N</i> corresponds to the number of threads used to process files smaller or equal to the specified size. The number of threads should not exceed the number of available CPU cores (the lower value of the client and server machines). If no size is specified, infinity is used as an upper bound.</p> <p>For example, to use 2 transfer threads to transfer files with a size smaller than 128 bytes and all other files with 6 transfer threads, use the following options:</p> <pre data-bbox="643 1696 1382 1759">--transfer-threads=2:128 --transfer-threads=6</pre>
<code>--remove-after-transfer</code>	<p>Remove source files after they are successfully synchronized.</p>
<code>--no-preserve-root-attrs</code>	<p>Disable preservation of attributes on Sync root.</p>

Examples of Sync Commands and Output

Sync Command Examples

1. **Continuous synchronization of a daily archive of large files on a Windows machine to Linux machine, preserving Windows acls, run as an async pull on the Linux machine:**

```
$ async -L /sync/logs -N backup -d /sync/backup -r
alligator@everglades.company.com:"C:\data\" -i /.ssh/lion_private_key
-K pull --remote-scan-interval=4h --preserve-acls=metafile --remote-
preserve-acls=metafile -C --exclude-dirs-older-than=1w0d0h0m0s
```

Details:

- Logs are stored on the Linux machine in the specified location.
- The user, lion, authenticates with an SSH key using the `-i` option
- Because the files in the backup are large, `remote-scan-interval` is used to scan the Windows machine every 4 hours, which forces an additional scan in case any notifications are missed.
- In order to preserve Windows acls in the backup, both `preserve-acls=metafile` and `remote-preserve-acls=metafile` must be specified.
- Since the archive directory creates a new directory for each day, use `exclude-dirs-older-than=1w0d0h0m0s` to avoid scanning directories that are no longer changing (older than a week).

2. **High performance push sync of many (10,000s) of small files (<10 KB) between Windows machines:**

```
> async -L c:/logs:200 -q -N small-files -c none --pending-max=10000 --
preserve-acls=native --transfer-threads=4 -R c:/logs:200 -d c:/data/ -r
bobcat@192.168.4.24:"C:\data\" -K push -l 500m
```

Details:

- Specifying the logging locations (`-L` and `-R`) is optional. Adding `:200` to the end of the log directory value allows the logs to reach 200 MB before being rotated.
- If the connection is secure, disabling encryption using `-c none` may boost performance.
- Increase the number of pending files from the default of 2000 using `--pending-max=10000`.
- The `--preserve-acls=native` option preserves Windows ALCs.
- Using more FASP threads to move the data can improve performance, set with `--transfer-threads=4`. The number of threads should not exceed the number of CPU cores (the lower value of the client and server machines).
- The user must enter the password at the prompt because it is not provided in the command. Aspera recommends using SSH keys for authentication, but this is not required.

3. **Non-continuous bidirectional sync of directories containing a mix of large and small files in which small files are synced using one thread and large files use another, run on a Linux machine to a Mac OS X machine:**

```
$ async -L /sync/logs -q -N sync-2017-01-01 -d /images --
user=gazelle@company.com --host=10.4.25.10 -r Library/data/images -i /
lion/.ssh/lion_private_key -R Library/sync/logs --transfer-threads=2:100k
-K bidi
```

Details:

- Logs are saved in the specified locations on both machines.
- The user authenticates with an SSH key using the `-i` option.
- The user and host are specified as separate options, rather than as part of the destination folder, so that the username with an `@` can be used (`@` is reserved in an `-r` argument for specifying the host).

- The sync uses two threads, one for files larger than 100 KB and one for files less than or equal to 100 KB, specified with the `--transfer-threads` option.

Sync Output Example

When Sync is run in interactive mode, the status of each file in the synchronized directory is displayed in a list similar to the following:

```
/file1          SYNCHRONIZED
/file2          SYNCHRONIZED (exs)
/file3          SYNCHRONIZED (skp)
/file4          SYNCHRONIZED (del)
```

The status may be one of the following options:

- SYNCHRONIZED: file transferred
- SYNCHRONIZED (skp) : file skipped
- SYNCHRONIZED (del) : file deleted
- SYNCHRONIZED (ddp) : dedup (duplicate files present)
- SYNCHRONIZED (exs) : file exists
- SYNCHRONIZED (mov) : file has changed (renamed, moved, or different attributes)

Include/Exclude Filtering Rules

Notes on Separators and Case

- You can synchronize Windows, Linux, Mac OS X, and other Unix-based endpoints and servers, but must take care with path separators. The path separator "/" is supported on Windows and other platforms. The path separator "\" is platform-agnostic *only* for the options `-d/r/L/R/B/b` and `--keep-dir-local/remote`. In [Include/Exclude Filtering Rules](#) on page 25, however, "\" is exclusively a quoting operator and "/" is the only path separator recognized.
- Case always matters, even if the scanned file system does not enforce such a distinction. For example, on Windows FAT / NTFS file systems and Mac OS X HPFS+, if the user writes "DEBUG", it matches files "Debug" and "debug". *In filter rules, however, the comparison is always exact.* To pick up both "Debug" and "debug", the pattern must be "[Dd]ebug".

Specifying Rules on the Command Line

Rules may be specified on the command line (for example, "`--exclude PATTERN`") or read from a file (for example, "`--exclude-from=FILE`") to make `async` skip or include specific directories and files. The following command-line options can be freely intermixed:

- `--exclude`
- `--include`
- `--exclude-from`
- `--include-from`

Rules are applied in the order that they are encountered, and the first matching rule (whether *including* or *excluding*) takes precedence. For example, if the file `aspera-test` specifies rules `W`, `X` and `Y`, then the following options apply rules `V`, `W`, `X`, `Y`, `Z` (in that order).


```
--include V --exclude-from aspera-test --include Z
```

- 👉 **Note:** Directories and files are visited in strict depth order, so that with the command-line options `--exclude /above/ --include /above/below` the file `/above/below` is never considered. The

directory `/above/` is not scanned; thus, `"/above/below"` has no opportunity to be checked against any rules.

Specifying Rules in `aspera.conf`

Rules may be specified in `aspera.conf`, as they are for `ascp` sessions, to apply to sessions run by a specific user or all users. These rules are applied first, then any command line filters are applied.

 **Caution:** Filters set in `aspera.conf` apply to both `ascp` and `async` sessions. If you do not want `async` filtering rules to apply to `ascp` sessions, set the rules for a specific user and use that user for `async` sessions. If you notice your `async` sessions are being filtered in unexpected ways, search `aspera.conf` for `<filters>` to determine what rules have been configured. You can find `aspera.conf` in the following location:

```
/opt/aspera/etc/aspera.conf
```

To specify an inclusion, start the filter pattern with '+' (+ and a whitespace, such as `+ *.jpg`). To specify an exclusion, start the filter pattern with '-' (- and a whitespace, such as `- *.png`).

To set filters for a specific user, run the following `asconfigurator` command:

```
# asconfigurator -x
"set_user_data;user_name,username;file_filters,|filter1[|filter2]"
```

To set filters for all users, run the following:

```
# asconfigurator -x "set_node_data;file_filters,|filter1[|filter2]"
```

The separator "|" is not required if only one filter is set.

Individual Matching Rules

Rules applied to directory and file names are limited regular expressions (referred to as "globs" in UNIX terminology). Rules use standard globbing syntax and globbing extensions, as described in the tables below.

Standard Globbing Syntax

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

Examples of Standard Globbing

Standard Globbing	Example	Matches	Does Not Match
/	abc/def/xyz	abc/def/xyz	abc/def
\	abc\?	abc?	abc\? abc/D abcD
*	abc*f	abcdef abc.def	abc/f abc/def abc/.def
?	abc??	abcde	abcdef abc/d abc/.
[...]	[abc]def	adeb bdef cdef	abcdef ade
[^...]	[^abc]def	zdef .def 2def	adeb bdef cdef /def /.def
[!...]	[!abc]def	zdef .def 2def	adeb bdef cdef /def /.def
[x-x]	[a-z]def	cdef ydef	Adef 2def .def

Globbing Extensions

Globbing Extensions	Description
/**	Like * except that it also matches the / character.
* or /** at end of rule	Matches both directories and files.
/ at end of rule	Matches directories only.
no / at end of rule	Matches files only.
/ at start of rule	Matches from the system's root directory (absolute path) only; that is, the entire string must be matched. Note: The / means the system's root, not the docroot, and not from the top level specified for the transfer set.

Examples of Globbing Extensions

Globbing Extensions	Example	Matches	Does Not Match
/**	abc/**/def	abc/def abc/x/def abc/.wxy/ def abc/wxy/tuv/def	abc/xyz/def/ zabc/wxy/def
* at end of rule	abc*	abc/file abc/dir	
/** at end of rule	abc/**	abc/.file abc/dir abc/ wxy/.dir abc/wxy/tuv/file	abc/
/ at end of rule	abc*/	abc/dir	abc/file
no / at end of rule	file	file	dir
/ at start of rule	/abc/def	/abc/def	.../abc/def

Rule Composition

Example	Expected Behavior
<code>--include A --include B</code>	Matches all file or directory names that match A or B.
<code>--include A --exclude B</code>	Matches all file or directory names that match A; of the rest, <i>excludes</i> all that match B.
<code>--exclude B --include A</code>	Excludes all file or directory names that match B; of the rest, matches those that match A.
<code>--exclude-from=FILE1 --include-from=FILE2</code>	<p>Read filter specifications from FILE1 and FILE2</p> <ul style="list-style-type: none"> Rules in the file are appended as if from a series of <code>--include A --exclude B</code> options Lines that start with <code>+</code> or <i>plus, space</i>, (e.g., <code>+ A</code>), are equivalent to the option <code>--include A</code> Lines that start with <code>-</code> or <i>minus, space</i> (e.g., <code>- B</code>), are equivalent to the option <code>--exclude B</code> Lines that start with <code>.+</code> or <i>dot, plus, space</i> (e.g., <code>.+ F</code>), are equivalent to the option <code>--include-from=F</code> Lines that start with <code>.-</code> or <i>dot, minus, space</i> (e.g., <code>.- G</code>), are equivalent to the option <code>--exclude-from=G</code> Lines that start with <code>.</code> or <i>dot, space</i>, name a file to read for more rules, but without a default (i.e., each line must start with one of the above commands). Other lines default to include (for <code>--include-from</code>) or exclude (for <code>--exclude-from</code>) rules Leading white space, blank lines and comment lines (<code># comment</code>) are ignored


A file or directory name that does not match any rule is still tracked, as if by a final `"*"` and `".*"`. Also, to reliably exclude all unmatched files, add two final rules: `"- *`" and `"- .*"`.

Semantics

Excluded new files are invisible to `async`. Files that have already been synced continue to be tracked even when they have, or get, a name that is now excluded. When run with `--exclude FILE3`:

Local event	Effect on peer (previously synced)	Clean start
<code>mv FILE4 FILE3</code>	<code>mv FILE4 FILE3</code>	<code>rm FILE4</code>
<code>rm FILE3</code>	<code>rm FILE3</code>	(ignored)
<code>cp FILE4 FILE3</code>	<code>cp FILE4 FILE3</code>	(ignored)
<code>mv FILE3 FILE4</code>	<code>mv FILE3 FILE4</code>	new file FILE4

Include/Exclude Filtering Examples

-  **Note:** You can synchronize Windows, Linux, Mac OS X, and other Unix-based endpoints and servers, but must take care with path separators. The path separator `"/"` is supported on Windows and other platforms. The path separator `"\"` is platform-agnostic *only* for the options `-d/r/L/R/B/b` and `--keep-dir-local/`

remote. In [Include/Exclude Filtering Rules](#) on page 25, however, "\"" is exclusively a quoting operator and "/" is the only path separator recognized.

1. Include files under top-level directories Raw and Jpg. Exclude all others.

```
# async ... --include='/Raw/**' --include='/Jpg/**' --exclude='*' \
--exclude='.*' ...
```

2. Same as Example (1), except also include directories starting with ".", at any level.

```
# async ... --include='./' --include='/Raw/**' --include='/Jpg/**' \
--exclude='*' --exclude='.*' ...
```

3. Same as Example (2), except exclude regular files ending in "~" or ".thm".

```
# async ... --include='./' --exclude='.*~' --exclude='*~' \
--exclude='*.thm' --exclude='*.thm' --include='/Raw/**' \
--include='/Jpg/**' --exclude='*' --exclude='.*' ...
```

4. Same as Example (3), except include only certain directories under Jpg.

```
# async ... --exclude='.*~' --exclude='*~' --exclude='*.thm' \
--exclude='*.thm' --include='./' --include '/Raw/**' \
--include='/Jpg/Big/**' --include='/Jpg/Med/**' \
--exclude='*' --exclude='.*' ...
```

Note that the long sequence in Example (4) can also be represented as a file (as shown below).

```
# async ... --exclude-from=- <<EOF
# no regular files with ~ suffix, dot or otherwise:
  .*~
  *~
# similarly for ".thm" suffix files:
  *.thm
  *.thm
# include directories starting with "."
+ ./
# include everything else found under top-level Raw :
+ /Raw/**
# and under Big/ and Med/ in Jpg:
+ /Jpg/Big/**
+ /Jpg/Med/**
# but nothing else:
*
.*
EOF
```

Bi-directional Example

This section describes typical `async` usage for bi-directional synchronization.

- 👉 **Note:** You can synchronize Windows, Linux, Mac OS X, and other Unix-based endpoints and servers, but must take care with path separators. The path separator "/" is supported on Windows and other platforms. The path separator "\" is platform-agnostic *only* for the options `-d/r/L/R/B/b` and `--keep-dir-local/remote`. In [Include/Exclude Filtering Rules](#) on page 25, however, "\"" is exclusively a quoting operator and "/" is the only path separator recognized.

Example Options:

- Pair name = "asyncTwoWay"
- Local directory is /fio/S

- Remote directory and login is admin@192.168.200.218:d:/mnt/fio/S (Windows machine)
- Password is v00d00
- Target rate = 100,000 Kbps or 100 Mbps
- No encryption
- Transfer policy = fair
- Read-block size = 1048576 or 1MB
- Write-block size = 1048576 or 1MB
- Continuous transfer
- Bidirectional transfer

Example Command:

```
$ async -N asyncTwoWay -d /fio/S -r admin@192.168.200.218:d:/mnt/fio/S -w
v00d00 -l 100M -c none -a fair -g 1M -G 1M -C -K BIDI
```

Example Output:

```
/ SYNCHRONIZED
/a SYNCHRONIZED
/b SYNCHRONIZED
/c SYNCHRONIZED
/DIR1 SYNCHRONIZED
/A1 SYNCHRONIZED
/DIR2 SYNCHRONIZED
/A2 SYNCHRONIZED
/REMOTE_DIR1 SYNCHRONIZED
/REMOTE_DIR2 SYNCHRONIZED
/REMOTE_DIR1 SYNCHRONIZED
SYNCHRONIZED (del)
/DIR1/a SYNCHRONIZED
/DIR1/b SYNCHRONIZED
/DIR1/c SYNCHRONIZED
[idle ] Found/Synced/Pending/Error/Conflict=9/9/0/0/0
```

Sync to S3 on AWS Cloud Object Storage

Sync can be used to sync files when the source or destination is S3 on AWS Cloud Object Storage.

Capabilities:

- Non-continuous PUSH, PULL, and BIDI work fully between a local disk and S3 on AWS.
- Non-continuous PUSH between S3 storages is supported.
- Continuous PULL and BIDI when S3 is the content source; requires the `--scan-interval` option.

Requirements:

- An Aspera On Demand instance in S3 on AWS, or Enterprise Server for Linux version 3.6.0 or later installed on a virtual machine instance in AWS with Trapd enabled. For instructions on configuring a non-On Demand instance, see the [Enterprise Server Admin Guide for Linux: Enabling EC2 on AWS/S3 on AWS Using the Command Line](#).
- The S3 instance must have an On Demand entitlement and a Sync-enabled license.
- The `async` binary must be installed on both the source and destination server.
- Configure the S3 instance as described in the following steps:

1. SSH into your instance as root by running the following command.

The command is for Linux but also works for Mac. Windows users must use an SSH tool, such as Putty.

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

2. Elevate to root privileges by running the following command:

```
# sudo su -
```

3. Set an S3 docroot for the system account user that will be used to run Sync.

```
# asconfigurator -x "set_user_data;user_name,username;absolute,s3://s3.amazonaws.com/bucketname"
```

If you are not using IAM roles, then you must also specify the S3 credentials in your docroot:


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

4. Set database and log directories for async by running the following commands:

```
# asconfigurator -x "set_node_data;async_db_dir,/mnt/ephemeral/data/db"
# asconfigurator -x "set_node_data;async_log_dir,/mnt/ephemeral/data/log"
```

This specifies that the logs and Sync database are stored in no-cost ephemeral storage associated with your instance.

Sync Examples:

 **Note:** If the client is on the cloud storage host, the following options are required:

- The log directory and local database directory should be specified using the following options:

```
-L /mnt/ephemeral/data/log -b /mnt/ephemeral/data/db
```

- For PUSH and BIDI modes, async must use the `apply-local-docroot` option.

A one-time (non-continuous) sync with a local disk pushing to an S3 bucket using SSH keys (for more information on using SSH keys, see [Creating SSH Keys](#) on page 41):

```
# async -N sync-to-s3 -d /data/data-2017-01 -r bobcat@192.0.4.24:/data -i /bobcat/.ssh/private_key -K push -B /mnt/ephemeral/data/db
```

A one-time bidirectional sync run from the S3 client to a local disk:

```
# async -L /mnt/ephemeral/data/log --apply-local-docroot -N bidi_london -d /data -r bear@192.0.12.442:/data -K bidi -b /mnt/ephemeral/data/db -B /async/log
```

Working with the Aspera Watch Service

Starting the Aspera Watch Service

The Aspera Watch Service (`asperawatchd`) is a file system change detection and snapshot service for speed, scale and distributed sources. It discovers changes (such as new files and directories, deleted items, and renames) in source file systems as they occur, eliminating the need to scan the file system. It can be used on any local or shares (CIFS, NFS) host.

When used in conjunction with `ascp` commands, the Aspera Watch Service allows for fast detection and send of new and deleted items. By comparing snapshots of the file directory it is watching, `asperawatchd` generates file lists for `ascp` commands for transfers and delete requests.

Aspera Sync can be configured to use `asperawatchd` for fast synchronization of very large numbers of files without scanning the directory. Aspera Sync can push files from a local directory, pull files from a remote directory, or create a bi-directional session between two directories (as long as `asperawatchd` is properly configured to monitor both directories). For more information on using Sync, see the [IBM Aspera Sync User Guide](#).

To **push** files to a remote host using Aspera Watch Service, configure `asperawatchd` on the local host. The remote host does not need to be configured. To **pull** files from a remote host, configure `asperawatchd` on the remote host. The local host does not need to be configured.

To start `asperawatchd`, follow these steps:

1. Configure a docroot for the user who will run the service.

The docroot is a security feature that allows you to restrict the area `asperawatchd` can access. If you need to access the entire file system, you can set the docroot path as `/`, but you cannot leave it empty.

Set the docroot for the user using the `asconfigurator` utility:

```
# asconfigurator -x "set_user_data;user_name,username;absolute,docroot"
```

For example:

```
# asconfigurator -x "set_user_data;user_name,root;absolute,/"
success
user_name: root
```

The `asconfigurator` command adds the following configuration to the `<aaa>` section of `aspera.conf`:

```
<aaa>
  <realms>
    <realm>
      <users>
        <user>
          <name>root</name>
          <file_system>
            <access>
              <paths>
                <path>
                  <absolute>/</absolute>
                </path>
              </paths>
            </access>
          </file_system>
        </user>
      </users>
```



```

    </realm>
  </realms>
</aaa>

```

2. Ensure the user has permissions to write to the default log directory if no directory is specified.
3. To **pull** files from a remote host, on the remote host set the data storage for either default or for specific users.

Setting data storage under default uses `asperawatchd` for all pull requests to the server whereas setting data storage for a specific user means that `asperawatchd` is used only for pull requests by that user. You can use `asconfigurator` to set either of these, or you can edit `aspera.conf` manually. To add the data storage to the default section, run the following command:

```

#asconfigurator -x
"set_node_data;async_watchd,redis:hostname:31415[:domain]"

```

This adds the following line to `aspera.conf`:

```

<default>
  <async_watchd>redis:hostname:31415:domain</async_watchd>
</default>

```

To add the data storage to a specific user, run the following command:

```

#asconfigurator -x
"set_user_data;user_name,username;async_watchd,redis:hostname:31415[:domain]"

```

This adds the following line to `aspera.conf`:

```

<aaa>
  <realms><realm>
    <users>
      <user>
        <name>username</name>
        <file_system>
          <access><paths><path>
            <absolute>/data</absolute>
          </path></paths></access>
        </file_system>
        <async_watchd>redis:hostname:31415[:domain]</async_watchd>
      </user>
    </users>
  </realm></realms>
</aaa>

```

Validate the `aspera.conf` file using the `asuserdata` utility:

```

# /opt/aspera/bin/asuserdata -v

```

4. Configure the `asperawatchd` service to run under your user.

The following command adds the `asperawatchd` service to the `asperarund` database; `asperarund` automatically starts and preserves services in its database. The `asperarund` service must be running.

```

# /opt/aspera/sbin/asperawatchd --user username

```

5. Verify that `asperawatchd` is running under the given user.

Use the `aswatchadmin` utility to retrieve a list of running daemons. Daemons are named with the username you passed in when starting the service. For example, if you used the root user to run your services, you should see the root daemon listed when you run the following commands:

```

# /opt/aspera/bin/aswatchadmin query-daemons

```

```
[aswatchadmin query-daemons] Found a single daemon:
root
```

6. Create a watch.

A watch is a directory watched by the `asperawatchd` service. To create a watch, run the following command, where *daemon* is the username used to start the `asperawatchd` service and *filepath* is the directory to watch.:

```
# /opt/aspera/bin/aswatchadmin create-watch daemon filepath
```

You can configure `asperawatchd` to watch multiple directories by creating more than one watch, but Aspera recommends specifying a single watch at the top level directory. Though you can specify as many watches as needed, (for example, one for each directory), specifying a single watch is the most efficient use of memory. For example, to monitor both the `/data/D1` and `/data/D2` directories, specify `/data` as the path for the watch.

With the Watch Service configured, you can now monitor watches and manage their files as they change.

asperawatchd Configuration

Many aspects of `asperawatchd` can be configured using the `asconfigurator` tool from the command line or by editing the server section of `aspera.conf`. To configure `asperawatchd` using `asconfigurator`, use this general syntax:

```
# /opt/aspera/bin/asconfigurator -x "set_server_data;option,value"
```

Options and values are described below.

To edit `aspera.conf`, open it in a text editor with administrative privileges.

```
/opt/aspera/etc/aspera.conf
```

For an example of the `asperawatchd` configuration in `aspera.conf`, see the sample below. Possible options and values are described below.

Remember to validate `aspera.conf` after any editing to confirm your configuration is correct. Run the following command:

```
# /opt/aspera/bin/asuserdata -v
```

To view the current settings, run the following command:

```
# /opt/aspera/bin/asuserdata -a
```

asperawatchd Configuration Options

The table below provides a reference of the configuration options for `asperawatchd`.

asconfigurator option aspera.conf setting	Description	Default
watchd_max_directories <max_directories>	The maximum number of directories that can be watched (combined across all watches). This is only used on Linux machines to overwrite the system value <code>/proc/sys/fs/inotify/max_user_watches</code> . To overwrite the system value with the	1000000

asconfigurator option aspera.conf setting	Description	Default
	aspera.conf value, run the setup procedure in the admin tool: <pre># aswatchadmin setup</pre>	
watchd_max_snapshots <max_snapshots>	The number of snapshots stored in the database before the oldest are overwritten.	10000
watchd_snapshot_min_interval <snapshot_min_interval>	The maximum amount of time between snapshots. If this period has passed without the minimum number of changes occurring to trigger a snapshot, a new snapshot is taken.	3s
watchd_snapshot_min_changes <snapshot_min_changes>	The minimum number of changes required to trigger a snapshot. If this number is reached before the snapshot minimum interval has passed, a new snapshot is taken.	100
watchd_log_dir <log_dir>	Specify the log directory.	The Aspera logging file
watchd_log_level <log_level>	Set the log level.	log
watchd_db_spec <db_spec>	Define how the local machine connects to the Redis database.	redis:127.0.0.1:31415:domain
watchd_scan_period <scan_period>	The amount of time between asperawatchd assessments of the watches (from end of one to start of the next). asperawatchd assesses watches for change independent of the snapshot minimum interval and snapshot minimum changes to ensure changes are captured.	30m

Example WatchD Configuration in aspera.conf

```
...
<server>
  <watchd>
    <max_directories>1000000</max_directories>
    <max_snapshots>10000</max_snapshots>
    <snapshot_min_interval>3s</snapshot_min_interval>
    <snapshot_min_changes>100</snapshot_min_changes>
    <log_dir></log_dir>
    <log_level>log</log_level>
    <db_spec>redis:host:31415:domain</db_spec>
    <scan_period>30m</scan_period>
  </watchd>
```

```
</server>
```

Aspera Watch Service Session Examples

Push Example

The asperawatchd service must first be properly configured and running on the local host to push files with asperawatchd. For more information about configuring and starting the Aspera WatchD service, see [Starting the Aspera Watch Service](#) on page 32.

To push files, start an Aspera Sync session with the `--watchd datastore:host:port[:domain]` option. For example:

```
async --watchd redis:localhost:31415:root -N watch_push -d /data/D1 -r
adminuser@10.0.0.1:/data/R1
```

Pull Example

The asperawatchd service must first be properly configured and running on the remote host to pull files with asperawatchd. For more information about configuring and starting the Aspera WatchD service, see [Starting the Aspera Watch Service](#) on page 32.

Aspera Sync uses the remote host's `aspera.conf` file to determine whether or not to use asperawatchd for the session. To pull files, start an Aspera Sync session with the `-K pull` option. For example:

```
async -N watch_pull -d /data/D1 -r adminuser@10.0.0.1:/data/R11 -K pull
```

Bi-Directional Example

The asperawatchd service must first be properly configured and running on both the remote and local hosts to start a bi-directional session with asperawatchd. For more information about configuring and starting the Aspera WatchD service, see [Starting the Aspera Watch Service](#) on page 32.

To start a bi-directional session, start an Aspera Sync session with the `--watchd datastore:host:port:domain` option and the `-K BIDI` option. For example:

```
async --watchd redis:localhost:31415:root -N watch_session -d /data/D1 -r
adminuser@10.0.0.1:/data/R11 -K BIDI
```

Remote from ascp Example

If you are using CIFS or NFS mounted storage, you must configure and run the asperawatchd service on the host running the NFS server, but neither the local host nor the remote host need to be configured.

On the NFS server, you must also set the Redis database to a non-loopback address by configuring Redis with a modified configuration file including the correct port and host address bindings. For example, if your host address is "10.54.44.194":

```
# Accept connections on the specified port, default is 6379.
# If port 0 is specified Redis will not listen on a TCP socket.
port 31415

# If you want you can bind a single interface, if the bind option is not
# specified all the interfaces will listen for incoming connections.
#
bind 10.54.44.194
```

Save your configuration file and then run the the asperaredisd service with the location of your configuration file.

```
# /opt/aspera/sbin/asperaredisd /filepath/redis_configuration.conf.
```

Point asperawatchd to the new Redis location by running the following command on your server:

```
# asconfigurator -x
"set_node_data;watchfolderd_db_spec,redis:redis_host:redis_port:domain"
```


For example,

```
# asconfigurator -x
"set_node_data;watchfolderd_db_spec,redis:10.54.44.194:31415:root"
```

Restart the asperawatchd service.

```
# /opt/aspera/bin/asperawatchd --user username
```


You can now start an Aspera Sync session from any client mounting NFS storage from that NFS server.

 **Important:** The path of your mounted directory must match the path of the directory on the NFS server. For example, if the directory is found at /data/D1 on the NFS server, you must mount it at /data/D1.

Start an Aspera Sync session with the local directory (-d) pointing to the mounted storage and the --watchd option pointing to the remote Redis monitored by asperawatchd. For example:

```
async --watchd redis:10.54.44.194:31415 -N watch_remote -d /data/D1 -r
adminuser@10.0.0.1:/data/R11 -K BIDI
```

In this example, the client on Host A starts the Aspera Sync session. The asperawatchd service on Host B (10.54.44.194) scans the /data/D1 directory mounted by Host A and passes the snapshot to Sync. Sync transfers the relevant files from the mounted storage to the target directory remote Host C (10.0.0.1). In this example, only Host B needs to be configured for asperawatchd.

 **Note:** These examples are all one-time sessions, but you can run any of these sessions in continuous mode (if the source machine is Windows or Linux) by using the -C option. In continuous mode, any changes you make to a monitored directory are detected by asperawatchd. Changes are propagated through Aspera Sync.

Logging and Reporting


asynccadmin Command-Line Options

Administrators can use the `asynccadmin` command to view the status of the current synchronization, as well as the latest snapshot. This includes the number of files in each state and any changes that may be incomplete on the remote endpoint. `asynccadmin` also includes troubleshooting options for deleting file records from a snapshot by path globbing match or filename. `asynccadmin` definitions, allowable values, and defaults are described below.

General `asynccadmin` usage:


```
# asynccadmin -d path [-N name] [options]
```

The `-N name` option is required if multiple Sync sessions are running; you must specify the name of the session to which the `asynccadmin` command should apply.

 **Note:** When records are deleted using the `-M` or `-E` options, Sync recalculates file counters for the entire database. This may take a while, depending on the size of the database.

Session Options


Option (short and long form)	Description
<code>-h</code> <code>--help</code>	Show <code>asynccadmin</code> command-line option help.
<code>-A</code>	Display the <code>asynccadmin</code> version.
<code>-d path</code> <code>--local-dir=path</code>	Choose the location of the local sync directory.
<code>-b path</code> <code>--local-db-dir=path</code>	Choose the location of the local database directory. The default location is the local sync directory.
<code>-N name</code> <code>--name=name</code>	Choose a source-destination pair from the snapshot database by name.
<code>-l</code> <code>--list</code>	List snapshot databases found in the database directory.
<code>-m</code>	Report meta details.

Option (short and long form)	Description
--meta	
-s --summary	Report the number of files in each state. When <code>-s</code> is used alone, a quick summary from the <code>async</code> database's <code>counters</code> table is reported back (same as the cached counters as in the <code>-t</code> option).
-s -v	When <code>-s</code> is used with <code>-v</code> , every record in the <code>async</code> database is counted.  Important: This should only be used when <code>async</code> is not running.
-f --file-info	Report the status of all files.
-j --journal	List changes that may be incomplete remotely.
-x --init	Delete all file system snapshot records.
-C --clean	Delete problem records (records with statuses of <code>CONF</code> , <code>PCONF</code> , and <code>ERR</code>).
-M <i>pattern</i> --match= <i>pattern</i>	Delete file records by path globbing.
-E <i>number</i> --erase= <i>number</i>	Delete file record by number.
-q --quiet	Emit only the information requested and do not abbreviate file names in <code>-f</code> / <code>--file-info</code> listing
-F --force	Allow changes while database is already in use.
-p --pause	Pause when printing a large amount of data (for example, <code>-f</code>).
-t <i>num</i> --tail= <i>num</i>	Report status of last <i>num</i> files.
-v	Increase the verbosity of summary (<code>-s</code>) or file info (<code>-f</code>).

Option (short and long form)	Description
<code>--verbose</code>	
<code>-T</code> <code>--terminate</code>	Shut down <code>async</code> if it is running. This option cannot be used if the storage style set for <code><async_db_spec></code> is LMS and outputs an error message.
<code>-O</code> <code>--compact</code>	Compact the database file.
<code>--touch=path</code>	Change the recursive mtime of the node and all its parents to current time if they are older. This option is only applied if <code>async</code> has been run using the <code>--exclude-dirs-older-than</code> option.

Logging

By default, Aspera Sync logs all file system synchronization events and transfers to `syslog`, including any errors that were encountered by synchronizing hosts.

-  **Important:** If you attempt to sync a directory without the proper read/write permissions, the directory and files it contains are *not* marked with an ERROR flag in the file directory status output. However, the error will be noted in the log file.

Appendix

Hardlinks

On Unix-based systems, it's possible to encounter multiple files with the same inode. The most common case of this is a hardlink. `async` is agnostic as to whether two files with multiple inodes are hardlinks or if they are actually different. It assumes that directories have unique inodes.

Handling Hardlinks

- One or more hardlinks at the source become regular files at the destination.
- In continuous mode, if a file with multiple links changes, only that file is replicated at the destination (even though all links at the source changed).
- In scan mode (one-time and continuous startup), all files for that link are replicated at the destination.

Handling Moves in Scan Mode

- If a new file has only one link, `async` checks whether it is a move.
- If a new file has two or more links, `async` does not check whether it is a move (regardless of whether the other links are inside or outside the sync directory).
- For directories, `async` checks whether or not it is a move.

Handling Moves in Continuous Mode

For all files and directories, notifications tell `async` the old and new paths; thus, a move is recognized in all cases.

Creating SSH Keys

To log in to other Aspera servers with public key authentication, create key pairs from the command line. Follow the instructions below.

1. Create an `.ssh` folder in the home directory if it does not already exist.

```
$ mkdir /home/username/.ssh
```

Go to the `.ssh` folder and continue:

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

Run the following command in the `.ssh` folder. The program prompts you for the key-pair's filename. Press ENTER to use the default name `id_rsa`. For a passphrase, you can either enter a password, or press return twice to leave it blank:


```
# ssh-keygen -t rsa
```

3. Retrieve the public key file.

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


```
/home/username/.ssh/id_rsa.pub
```

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

 **Note:** You must have the server administrator install your public key file on the server in order to successful sync with it.

4. Use the key in an async session.

Use the option `-i private_key_file`, instead of `-w password`, as in the following example:

 **Note:** Your private key and public key must be located in the same directory.

rsync vs. async Uni-directional Example

Below are examples of `rsync` commands and their `async` equivalents for uni-directional synchronization.

Example 1

Options:

- Recursively synchronize the contents of a directory, `/media/` to the remote system directory `/backups/media`
- Preserve access and modification time stamps on files
- Preserve the owner and group ID
- No encryption
- Transfer policy = fair
- Target rate = 100,000 Kbps (100 Mbps)
- One-time transfer (not continuous)

`rsync` command:

```
# rsync --stats -v -r -u -l -t -o -g -p /media/
editor@docserver:/backups/media
```

`async` equivalent:

```
# async -N Oneway -u -t -j -d /media/ -r editor@docserver:/backups/media -l
100M -w d0c5 -K push -c none
```

Example 2

Options:

- Recursively synchronize the contents of the directory `/media/wmv/`
- Exclude "." files within the directory
- Exclude all other directories
- Preserve the owner and group ID
- Preserve access and modification time stamps on files
- No encryption
- Transfer policy = fair
- Target rate = 100,000 Kbps (100 Mbps)
- One-time transfer (not continuous)

`rsync` command:

```
# rsync --stats -v -r -u -l -t -o -g -p /media/ --include="/media" --
include="/media/wmv" --exclude="/media/.*" editor@docserver:/backups/media
```

async equivalent:

```
# async -N Oneway -u -t -j -d /media/ --include="/media"
--include="/media/wmv" --exclude="/media/*.*" -r
editor@docserver:/backups/media -w d0c5 -K push -c none
```

Options Comparison Table

rsync Option	async Option	Description
--stats	Enabled by default	Display file transfer status
-v, --verbose	Enabled by default	Increase verbosity
-q, --quiet	-q, --quiet	Disable progress display
-r, --recursive	Enabled by default	Recurse into directories
-u, --update	If a file exists at the destination with the same name, then async's default behavior is to do nothing if the files are the same (size and checksum), and overwrite if the file is different.	Skip files that are newer on the receiver
-l, --links	Linux and Mac OS X: <code>-n copy</code> , <code>--symbolic-links=copy</code> Symlinks are skipped in Windows.	Copy symlinks as symlinks (Linux and Mac OS X only)
-t, --times	-t, --preserve-time (must have Enterprise Server, Connect Server, or Point-to-Point 3.1+)	Preserve modification times
-o, --owner	-u, --preserve-uid	Preserve owner
-g, --group	-j, --preserve-gid	Preserve group
-p, --perms	With regard to directory attributes, if the source mode doesn't have sufficient owner permissions, then the destination will add: owner rwx.	Preserve permissions
--version	-A, --version	Print version number
-h, --help	-h, --help	Show help
--include-from= <i>file</i>	-I, --include-from= <i>file</i>	Include filter (text file with paths for inclusion). See Include/Exclude Filtering Rules on page 25.
--exclude-from= <i>file</i>	-E, --exclude-from= <i>file</i>	Exclude filter (text file with paths for exclusions). See Include/Exclude Filtering Rules on page 25.
--include= <i>pattern</i>	--include= <i>pattern</i>	Include paths that match <i>pattern</i> . See Include/Exclude Filtering Rules on page 25.
--exclude= <i>pattern</i>	--exclude= <i>pattern</i>	Skip paths that match <i>pattern</i> . See Include/Exclude Filtering Rules on page 25.
	-c <i>none</i>	<code>rsync</code> , as a protocol, does not encrypt on its own; however, <code>rsync</code> can enable/disable

rsync Option	async Option	Description
		the SSH encryption protocol (using option <code>-e ssh</code>).

Error Troubleshooting

Error returned when you attempt a continuous sync

If you attempt to run a continuous Sync from a client that does not support continuous mode, you receive the following error:

```
Failed. File system change notification not supported by platform
(code=45112)
```

If you attempt to run a continuous Sync to a server that does not support continuous mode, you receive the following error:

```
Failed. [PEER] File system change notification not supported by platform
(code=45112)
```

Solution: You must run your Sync session to or from a machine with an operating system that supports continuous mode. See the table below.

Continuous Sync Direction	Supported Sync Client OS	Supported Sync Server OS
PUSH	Linux, Windows	All
PULL	All	Linux, Windows
BIDI	Linux, Windows	Linux, Windows

Error returned when you attempt to monitor a Linux directory in continuous mode

If you attempt to monitor a directory on a Linux system (i.e., you are running `async` in continuous mode), you might receive the following error:

```
Failed to set up directory change notification - reached the per-user limit
on number of inotify watch descriptors.
```

Cause: You have exceeded the per-user limit imposed by the OS on the number of directories that may be monitored (i.e., the number of inotify watch descriptors).

Solution: You must modify the kernel parameters on the Linux machine to increase the maximum number of user watches. The following procedure may differ between Linux versions; consult your operating system Administrator's guide for more information.

On the Linux machine, open `/etc/sysctl.conf` in a text editor and increase the maximum number of user watches. Enter a value that exceeds the maximum number of directories ever expected to exist in the docroot that is monitored by `async`. For example,

```
fs.inotify.max_user_watches=1000000
```

Save your changes and run the following command to load the configuration changes:

```
# sysctl -p
```

To confirm that the changes have been applied, run the following command:

```
# sysctl -a | grep max_user_watches
fs.inotify.max_user_watches=1000000
```

The Sync client displays failure to start sync error

When the `async` binary on the remote computer cannot initialize, the `async` client gets a generic error similar to the following:

```
Failed to start sync session
```

Cause: There are several possible causes.

- `async` binary doesn't exist (or is not in the path and `sshd` cannot find/execute it).
- `async` binary cannot be run.
- `async` binary cannot initialize properly (e.g., if the system is out of memory or socket resources).
- `async` binary cannot create its log files, if specified with `-R` (bad path, bad permissions).

Solution: Check your system for the possible causes above.

Error returned when you attempt sync from Linux to Windows.

When you attempt to perform a sync from Linux to Windows, you receive the following error:

```
Failed. Peer error: Symlink policy copy not supported on Windows peer.
```

Solution: You must specify `-n skip` or `--symbolic-links=skip` when performing the sync.

Technical Support

Support Websites

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

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

Personalized Support

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

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

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

Legal Notice

© 2011-2017 Aspera, Inc., an IBM Company. All rights reserved.

Licensed Materials - Property of IBM

5725-S60

© Copyright IBM Corp. 2011, 2017. Used under license.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Aspera, the Aspera logo, and FASP transfer technology are trademarks of Aspera, Inc., registered in the United States. Aspera Connect Server, Aspera Drive, Aspera Enterprise Server, Aspera Point-to-Point, Aspera Client, Aspera Connect, Aspera Cargo, Aspera Console, Aspera Orchestrator, Aspera Crypt, Aspera Shares, the Aspera Add-in for Microsoft Outlook, and Aspera Faspex are trademarks of Aspera, Inc. All other trademarks mentioned in this document are the property of their respective owners. Mention of third-party products in this document is for informational purposes only. All understandings, agreements, or warranties, if any, take place directly between the vendors and the prospective users.