

Release Notes: IBM Aspera Sync 3.7.3 for AIX

Product Release: March 21, 2017

Release Notes Updated: March 21, 2017

This release of IBM Aspera Sync for AIX provides the new features, fixes, and other changes listed below. Additional sections cover system requirements and known problems.

NEW FEATURES

- Sync is now documented for AIX. Below are highlights of new Sync functionality. For more information, see the [IBM Aspera Sync Admin Guide 3.7.3 for AIX](#).
- On machines with Unix-based operating systems, a new value, `inode`, is available for the `--dedup` option. In this mode, 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.
- Sync with S3 is now supported and documented.
- Improved Sync shutdown process when a Sync session is stopped.
- Sync sessions can now be filtered using the `<filter>` configuration in `aspera.conf`. Command line filters are applied after `aspera.conf` settings. If filtering is not always desired, configure filters for one user. Sync sessions run by that user are filtered while sync session run by other users are not.
- Sync now supports access key authentication. Access keys provide an alternative to the security credentials of a node user or system user. Because an access key is restricted to its own storage (local or cloud), it allows access control and usage reporting to be segregated by storage. This offers significant benefits to multi-tenant service providers and enterprise installations with multiple departments. This also allows `async` to run on Aspera Files and for other REST-ful applications.
- A new `asyncaadmin` option prints file attributes (mtime, recursive mtime for directories, and checksum).
- Sync now preserves POSIX attributes in the event that file attributes change but file contents do not (times, UID, GID).
- The new recursive `mtime` option enables the exclusion of directories and files older than a configured timestamp.
- The new `--delete-delay` option allows the delete operations in a unidirectional sync to be delayed until the end of the synchronization.
- When using the `--keep-dir-local` or `--keep-dir-remote` options, locating the `keep-dir` under the same directory as the sync directory is no longer an error condition as long as the `keep-dir` has been added to the exclude list.
- The `Async` library now takes a license as a parameter. If the library doesn't get a license on initialization, it can still search for one on the disk, as it does currently.
- `async` now accepts external cookies, tags, and reports through management.
- Sync has the following new options:
 - `--local-mount-signature=signature_file`**
 - `--remote-mount-signature=signature_file`**
Verify local/remote file system is mounted by the existence of this file. NOTE: Use of this feature will increase the time to synchronize deletes.
 - `--pending-max=number_of_files`**
The maximum number of pending files to allow. Default is 2000.
 - `--compression=mode`**
Specify method used to compress a file before transfer. `mode` can be `zlib` or `none`. The default value is `none`.
 - `--transfer-threads=number[:size]`**
Specify intervals for dedicated transfer threads. `number` corresponds to the number of threads used to process files smaller or equal to the specified `size`. If no `size` is specified, infinity is used as an upper bound.
 - `--remove-after-transfer`**
Remove source files after they are transferred correctly.
 - `--scan-interval=duration`**
Enable periodic scans. `duration` defines the interval between periodic scans. `duration` may be specified as `DDd HH:MM:SS.mmm` or `NNw NNd NNh NNm NNs NNms NNus`.
 - `--remote-scan-interval=duration`**
Enable periodic scans on the remote machine.

- Sync also has three new options for preserving timestamps:
 - `--preserve-access-time`
Preserve file access time.
 - `--preserve-modification-time`
Preserve file modification time.
 - `--preserve-creation-time`
Preserve file creation time (Windows only).

ISSUES FIXED IN THIS RELEASE

Note: This release contains tickets created from different tracking systems. For this reason, the ticket IDs may reflect different numbering formats.

ES-131 - The installer changes permission on system-level directories. (CIM-208)

WAT-449 - Sync does not respect the cooloff option.

SYSTEM REQUIREMENTS

AIX: 6.1, 7.1

KNOWN ISSUES

Note: This release contains tickets created from different tracking systems. For this reason, the ticket IDs may reflect different numbering formats.

Sync

Async on AIX does not support continuous PUSH or BIDI modes.

ES-238 - When a client-side `asynsession` is forced to quit, in rare cases the server-side `asynprocess` may not stop. (CIM-364)

ES-92 - Sync reports incorrect counts for 'deleted_paths', 'deleted_bytes', and 'cumulative_deleted_bytes'.

ES-50 - When using local storage for Sync with Files on Unix-based systems, Sync fails if the access key storage does not start with "/". For example, `datastorage` fails but `/datastorage` is successful.

WAT-512 - The `--overwrite` option does not recognize modified files if the size has not changed and they are in storage that does not support sparse checksums, such as Azure cloud storage, or when `async` is run with the `-checksum none` option.

WAT-465 - Sync hangs following a TCP impairment that produces a `libssh2` timeout or error.

WAT-377 (#27391) - [Unix-based OS] A continuous `async` session configured to follow symlinks will not sync a symlink file after it has been modified.

WAT-362 (#24812) - If a file's size decreases during a continuous Sync push, the file remains pending and is never synced.

WAT-288 (#27311) - An `--apply-local-docroot` pull copies the local docroot path into the same path. For example, `/home/user1/sync` is copied into `/home/user1/sync`.

WAT-9 - When the `scan-file-rename` option is used with `asperawatchd`, moved files should be detected and renamed at the destination, not deleted and replaced by a transferred, renamed file.

#29038 - Using `overwrite=always` when syncing with cloud storage does not overwrite the file. The default checksum behavior with S3 (as with any cloud storage) is "none". An existing file on S3 is considered identical to the local file when their sizes are equal. Therefore the file on S3 is not overwritten, even when the content of S3 differs from the content of the local file.

#28887 - When a file or directory is moved using the `--exclude` command in Sync continuous mode, the old file or directory is not removed from the remote side.

#28817 - The Sync log entry for `SYNCERROR_DELAY` does not include information describing the file name and path.

#27621 - Sync conflicts due to handling of hidden, temporary, or transient files (such as those created by Microsoft Office products).

#25915 - When the sync function is set to continuous, overwriting an existing file on the initiating node may result in the deletion of the previous version of the file on the responding node. **Workaround:** Perform the sync manually to recreate the deleted file.

#24805 - For a synced directory, while all of the files and subdirectories under the source folder will have ACL preserved at destination, the ACL for the source folder itself will not be replicated to the remote destination folder.

#23954 - When the `--preserve-acls` or `--preserve-xattrs` option is used, `async` will not preserve the `acl` or `xattr` when either file `acl` or `xattr` is modified and (a) file content is unmodified or (b) file content is unmodified and the file is renamed.

#23400 - [Unix-based OS] As of Sync version 1.5+, the user is permitted to sync to the root directory.

#23004 - Console does not draw a line between the involved nodes for an `async` transfer. This issue occurs when using two managed nodes with two interfaces on two different networks.

#22633 - Sync does not support large `xattr/ResourceForks`.

#22044 - `asynadmin` reports all sessions locked after the last actual running session.

#21014 - When creating a file with `vi` during a sync, the swap file is in conflict.

#20906 - `async` can't create a watch on an unreadable directory; therefore, it does not get notified when permissions change. In addition, `async` treats an unreadable directory as "skip" rather than reporting an error or conflict.

#20767 - `-R log dir` creation from Linux to Windows with spaces in the path will only get as far as the first space in the path.

#20347 - `Async` reports errors and conflicts for deep directory depth when the sync is from Linux to Windows.

#19945 - `asynadmin` creates SHM and WAL files for read-only operations. Once `asynadmin` is run as the root, `async` run by the user does not have permission to access the existing SHM and WAL files and thus `async` fails. This is due to a bug in SQLite.

#16911 - Characters in the async session option that are not preceded by a "-" or "--" are ignored, therefore failing to trigger an error message. Any session options specified (such as -lor -a) after the string of characters not preceded by a "-" or "--" are also ignored. The session will still run, using the default values, without notifying the user that the command line settings have not been applied.

#13761 - If filenames contain "\" or new line, async will try to transfer but fail, causing the internal transfer queue to become full and the synchronization to stall (or hang after all the other files have completed).

#13645 - When a directory is renamed during transfer, Sync continues running and never completes.

PRODUCT SUPPORT

For online support resources for Aspera products, including raising new support tickets, please visit the [Aspera Support Portal](#). Note that you may have an existing account if you contacted the Aspera support team in the past. Before creating a new account, first try setting a password for the email that you use to interact with us. You may also call one of our regional [support centers](#).