Release Notes: IBM Aspera HSTS, HSTE, and Desktop Client 3.8.1 for AIX

Product Release: July 12, 2018
Release Notes Updated: July 13, 2018

This release of IBM Aspera High-Speed Transfer Server, High-Speed Transfer Endpoint, and Desktop Client for AIX provides the new features, fixes, and other changes listed below. In particular, the Breaking Changes section provides important information about modifications to the product that may require you to adjust your workflow, configuration, or usage. Additional sections cover system requirements and known problems.

Desktop Client users: Features and issues related to configuration, the Node API, and Watchfolder are not applicable to your product.

NEW FEATURES

General

- Transfers with Microsoft Azure Files are now supported, including using Azure Files access keys.
- Increased server security with upgrades to the OpenSSH SSHD service. (CIM-600)
- Desktop Client no longer requires a license. For upgrades to 3.8.1, existing licenses are overwritten with the unlimited license after a successful upgrade.
- `ascp` and `ascp4` transfers to object storage can now include custom metadata if the object storage supports it (currently S3, Google, Azure, and Swift). Metadata is set using the `--tags` or `--tags64` option with a JSON payload argument. Metadata are applied per session, not per file. (CIM-723)
- New traffic RTT predictor settings offer server configuration options that improve transfer rate stability and maximize FASP transfer performance.
- A new database mode for the Aspera Node Service, `acm_redis`, improves database cleanup and background job management for clusters of Aspera servers.
- A new post-transfer file validation process runs file validation once the transfer is complete, in contrast to existing inline validation methods. Out-of-tranfer validation is also applied to files transferred by HTTP(S) fallback, unlike inline validation. Files that are being processed are reported by Aspera Central (and Faspex and Console) as "validating", and then "complete" once the validation completes.
- `ascp` and `ascp4` logging can now be configured in `aspera.conf`. (CIM-958)
- Improved reporting for package transfers that are intiated by Aspera Central. (CIM-1327)

Ascp

- Transfer sessions that fallback to HTTP now report file IDs in the management stream (as FaspFileID).
- Uploads with a `stdio-tar://` destination can now use transfer tokens for authorization.
- The `stdio-tar://` source file can now specify an offset parameter that indicates where in the destination file the inline raw data should be inserted to overwrite the existing data.
- When using `stdio-tar://` as the source for an `ascp` transfer, the value for "File:" can be a directory and the directory structure is preserved at the destination. Additionally, `stdio-tar://` can now be used as the destination.

Ascp 4

- Ascp 4 and is now supported on AIX. Aspera Ascp 4 is an optimized transfer engine based on FASP technology and is designed for sending extremely large sets of individual files efficiently. The executable, ascp4, is similar to ascp and shares many of the same options and capabilities, as well as options that enable multi-threaded FASP transfer, TCP and UDP stream I/O, memory usage control, and filtering by when a file was last modified.
- The data-streaming capabilities of Ascp 4 are available for High-Speed Transfer Server and High-Speed Transfer Endpoint users.

Node API

- The Node API /ops/transfers now supports `ascp4` transfers. Specify that a transfer should use `ascp4` rather than `ascp` by adding the following line to a JSON request:

  ```
  "use_ascp4" : true
  ```
- The Node API can now pass instructions on how FASP transfers handle symbolic links. If no method is specified, the default policy is now `follow`.
- The logging thread to the kvstore database now times out after the FASP transfer session ends. The timeout period can be configured with a new setting, `<activity_log_queue_timeout>`, in `aspera.conf`.
- Partial files reported by a /files/browse call can now be identified with the new `"partial_file" : true` attribute, allowing them to be processed separately from complete files. To enable this, the `<partial_file_suffix>` must be set in `aspera.conf`. Files with the resume suffix are still filtered out from the dictionary.
- Access keys can now be backed up and restored by using new `asnodeadmin` options.
- POST requests to `/ops/transfers` can now use AES-192 and AES-256 ciphers for encryption of data in transit.
- When a file is deleted with `DELETE /files/{id}`, its preview file is now also deleted.
- The `files_cleanup_interval` setting in `aspera.conf` is now respected.
- Drastically faster response for `/files?sort=name` requests (files sorted by name); depending on the storage and number of files, browsing can be up to 20 times faster.
- Improved metadata rules for `ascp` decrease the load on the node database by only generating file IDs if `activity_event_logging` is enabled.
- `/files/delete` requests can now specify that folders that are not empty should not be deleted; otherwise, all folders are deleted by default.
- Improved Redis database performance by automatic expiration and removal of cached file metadata.
- Disabling `activity_logging` and `activity_event_logging` now turns off all event reporting, including transfer events, filelock events, and permission events.
- The Aspera Node daemon now locates `ascp` and `ascp4` relative to its own path, enabling the Aspera application to be installed in non-default location.
- POST requests to `/streams` can now specify more transfer parameters that are supported by Ascp4, including compression, read and write threads, and minimum transfer rate.
- Filelocks and previews are now cached for faster directory listing and browsing.
- The `/streams` endpoint now accepts bearer token authorization.
- File statistics are now returned by the `/files/`*`file_id`* endpoint even if the user does not have list permission.
- Filelocks are now disabled when `ascp` is run without token authorization.
- Improved access level reporting by the `/permissions` endpoint.
- Filelocks can now be created and removed for files by pathname by using POST requests to `/files/filelock` and `/files/unfilelock`. These endpoints provide an alternative to sending POST, PUT, and DELETE requests to `/files/{id}/filelock`, which require that you specify the file by file ID.
- Transfer settings can now be configured at the access key level. The transfer capabilities are returned in response to calls to `/info`.
- Filelock and permissions operations are now reported by the `/events` endpoint.
- New permissions model allows user-specific permissions for file operations, grouped by access levels (edit, view, or none).
- Node-to-node transfer requests now respect the `xfer_retry` that is specified in the tags, and no longer accept a user-specified `xfer_id` (because Node generates its own `xfer_id` to ensure a unique identifier).
- The `/ops/transfers` endpoint now supports updating the maximum (target) transfer rate, minimum transfer rate, and rate policy.
- POST requests to the `/ops/transfers` endpoint now supports additional `ascp` options in the transfer specification, including excluding files older than or newer than the specified time, preserving timestamps, and moving or deleting files after transfer.

- Faster transfer start up when many transfer sessions are started through the Node API. (CIM-1010)

Sync

- The Sync guide in now included in the High-Speed Transfer Server and High-Speed Transfer Endpoint guides. It includes new instructions for composing an `async` command and an expanded troubleshooting section.
- Improved handling of changing files by continuous Sync sessions when checksum is set to none; files that return a sharing violation error are now retried per `--sharing-retry-max`.
- Access key authentication is now supported by using "Basic: *token_string*" as an argument for the `-W` option.
- The `--dedup` option can now be used in `async` commands even if it is not specified in the first run; however, `--dedup` is rejected if the first run does not use `-k` (calculate file checksums). The dedup index is created if it does not already exist, and if the database is large then this process can take some time.
- Sync can now use multiple scanning threads on the local and remote computers to improve performance by decreasing the time required for directory scanning after the initial scan. Specify the number of threads by using the new command line options `--scan-threads` and `--remote-scan-threads`.
- Sync logging location, level, and size can be configured in `aspera.conf` using new logging settings. Command line options and `<async_log_dir>` take precedence over the new settings.
- Improved Sync logging when `<async_log_dir>` is set in `aspera.conf`, with all logging going to the specified directory.
- Sync sessions with object storage can now include custom metadata if the object storage supports it (currently S3, Google, Azure, and Swift). Metadata is set using the `--tags` or `--tags64` option with a JSON payload argument. (CIM-723)
- File metadata can now be preserved (using `-u -j -t`) when `--dedup=copy`. (File metadata are always preserved when `--dedup=hardlink` or `--dedup=inode`).
- Improved logging about Sync database (`snap.db`) loading.
- Sync can now use a cluster as an endpoint. Specify the remote host with the cluster DNS and provide a unique session name. Aspera recommends creating the session name with the UUID and a descriptive string, for example: `async -N cluster-sync-ba209999-0c6c-11d2-97cf-00c04f8eeAscp45`.
- A new async option, `--ignore-mode`, prevents the file mode from being synced from the source to the destination. Use this option to allow the file to have different modes on the source and destination and to prevent Sync from hanging if the destination permissions change.
- Sync now respects the `files_filelock_enabled` setting in a user's access key, which overrides the server setting in `aspera.conf`.
- A new Sync option `--clean-excluded` can be used to optimize the Sync snapshot database when using `--exclude-dirs-older-than` or `--exclude` by removing directories from the snapshot as they become excluded. The option applies to all Sync directions and the excluded paths are removed from the snapshot database on both endpoints.

Watchfolder and Aspera Watch Service

- Faster directory scanning by the Aspera Watch Service, particulary of directories that contain many (10,000s) subdirectories.
- Watchfolders now supports IPv6 addresses. (CIM-531)
- Watchfolder can delete files from the source as soon as the file is successfully transferred, rather than waiting for the session to complete, by editing the Watchfolder configuration JSON file or by enabling it in Console (In the Console GUI, go to **File Handling > Source deletion** and select **Automatically delete a source file after transfer of this file**). (CIM-493)
- The Aspera Watch Service daemon now uses a single snapshot tree that represents the entire file system and monitors portions of the file system to which users subscribe. This system reduces memory requirements and simplifies watch configuration.

  A user subscribes for file system notifications on a directory, and the Aspera Watch daemon creates a watch for the directory and a subscription ID for the user. The user can unsubscribe from watches or renew a subscription (if it is nearing expiration) by using the subscription ID. If no users are subscribed to the watch, then the watch is automatically deleted, decreasing the load on the Redis database. The subscription system also allows the Redis database to delete snapshots that are no longer needed by any users, for additional database space savings.

When upgrading to 3.8, existing Watch Folders are preserved with existing watches converted into subscriptions. For example, a Watch Folder with one watch becomes a Watch Folder with two subscriptions, one for the watch and one for the Watch Folder itself. See the guide for more information about the new subscription model and preparation for upgrading.

- Watch Folders and the Aspera Watch Service now support cloud storage and URI docroots. Object storage requires that a small scan period be set for the Watch Service subscription because cloud storage does not have a notification API.
- Watch Folders can now be created in "pull" mode, such that a folder on a remote host can be watched and automatically transfer files to the local computer. The remote host can be an Aspera server in object storage.

  The Watch Folder JSON configuration file syntax for the source and target now require that you specify the type of authentication, the port for authentication, and authentication credentials for the remote server (rather than in the `"target"` section). Post-processing is now specified for the source (rather than `"local"`). A new section for watchd configuration enables you to specify the remote watchd service.

  The previous version of the Watch Folder API and JSON configuration is still supported for push Watch Folders, but pull Watch Folders require that the remote server run version 3.8.0 or higher.
- Watch Folders can now use IBM Aspera Shares version 1.9.11 (with patch) as a remote endpoint, authenticated by using Shares credentials.
- Snapshot differentials created through the Watch Service REST API can now be calculated asynchronously for more efficient processing of large differentials.
- Watch Folders can now use access keys for authentication to remote storage. Remote sources (for pull Watch Folders) must have an Aspera Watch Service running. Access key authentication can be used for push Watch Folders with destinations of Aspera Files, Aspera Transfer Service, or Aspera Transfer Cluster nodes.
- Watch Folders can be configured to use a specific Aspera Watch Service.
- Watch Folder-initiated transfers to object storage can now include custom metadata if the object storage supports it (currently S3, Google, Azure, and Swift). Metadata is set in the Watch Folder configuration under "aspera" in a "cloud-metadata" section. (CIM-723)
- The symbolic link handling policy can now be specified in the Watch Folder configuration when creating Watch Folders with `aswatchfolderadmin` or the Watch Folder API.
- The Watch Folder daemon now reports if a Watch Folder license is missing or expired; this information can be retrieved using the API, from the status file, or by running `asrun send -l`.
- Watch and Watch Folder services that are stopped can now be restarted by resending the configuration to the Node service.
- The Aspera Watch Service and Watch Folder daemons are now gracefully shutdown by the Aspera Watch Services Manager (asrund), with improved reporting of daemon status.
- Watch Folders now supports AES-192 and AES-256 encryption.
- Faster drop statistics calculation by storing and updating statistics in the Redis database.
- GET calls to the `/drops` Watch Folder API endpoint now return the last error that occurred in the drop and the last error of a file in the drop. Additionally, a state filter can be specified in the query to limit the results to drops that match the state.
- A new Watch Folder API endpoint, `/schemas/watchfolders/configuration`, returns a JSON schema that provides the default value of each Watch Folder configuration field.
- The `/watchfolders` endpoints support concurrency for calls to the Redis database.

BREAKING CHANGES

If you are upgrading from a previous release, the following changes for this release may require you to adjust your workflow, configuration, or usage.

- Activity event reporting can now be configured with a new `aspera.conf` setting, `activity_event_logging`. Prior to 3.7.4, activity event reporting was always enabled. As of 3.7.4, activity event reporting is disabled by default to improve server performance, and it must be enabled in order to query the

Node API /events endpoint. **Nodes that are added to Aspera Files must have activity event reporting enabled.** To enable it, run the following command:

```
asconfigurator -x "set_server_data;activity_event_logging,true"
```

- Precalculating job size is no longer supported for persistent `ascp` sessions to avoid confusion when a transfer completes before the job size is calculated. (CIM-970)
- OpenSSH 7.0 and newer no longer supports DSA keys. If the client creates connections in version 3.7.3 or older of the GUI, HTTP/S-based connections (such as to Shares or ATS, or authenticated with Node API credentials) to Windows servers version 3.7.4 or newer, or with other OS servers that are using OpenSSH 7.0 or newer, fail to authenticate. Connections that provide a private SSH RSA key are not affected. **Workaround:** Upgrade the Aspera client to version 3.7.4 or newer.
- FASP transfers through IBM Aspera Forward Proxy Server now require that Proxy server self-signed SSL certificates include the hostname, otherwise transfers are refused. The self-signed certificates that are created upon installation must be replaced. For instructions on creating a certificate with a hostname, see "Setting up SSL for your Nodes" in the IBM Aspera High-Speed Transfer Server Admin Guide for Linux.
- Performance enhancements to Ascp 4 required changes that make version 3.8.0+ unable to transfer with versions 3.7.4 and earlier. **Workaround:** Upgrade your server and Ascp 4 clients to 3.8.0 or 3.8.1 to ensure compatibility.
- The `--delete-after` option is no longer supported by Ascp 4. Use `--delete-before` instead.
- The improvements to Watch Folders include several changes to the Watch Folder JSON configuration file syntax and to associated command line utilities:
  - The configuration settings for the Aspera Watch Service and Watch Folders services changed in order to simplify configuration. Individual watches are no longer configured in `aspera.conf`; watches are managed by subscriptions to Aspera Watch Services.
  - The command line option for the Aspera Watch Services Manager for returning information on services changed. A new option, `asrun send --list` (or `asrun send -l`) returns information for all services, equivalent to the behavior of `asrun send -g` or `asrun send --get` in versions 3.7.x. Users can now return information for a specific service using the modified `asrun send --get=`*`service_id`*; the service ID is now required for `asrun send --get` commands.
  - The options available for `aswatchadmin` changed. When subscribing to a Watch service, `--max-snapshots`, `--snapshot-min-interval`, and `--snapshot-min-changes` are no longer supported. The values for `snapshot-min-interval` and `snapshot-min-changes` are read from `aspera.conf`.
  - The use of PUT calls to `/v3/watchfolders/`*`watchfolder_id`*`/drops` has changed. PUT to `/v3/watchfolders/`*`watchfolder_id`*`/drops`, to restart all drops in a Watch Folder, is no longer supported. The drop ID must now be specified, as `/v3/watchfolders/`*`watchfolder_id`*`/drops/`*`drop_id`*.

ISSUES FIXED IN THIS RELEASE

ATT-511 - When both `--overwrite=never` and `--remove-after-tranfer` are used, source files may be deleted even if no transfer occurred. (CIM-932)

ATT-579 - Persistent `ascp` uploads to a server that is configured to skip symbolic links do not report when symbolic links are skipped.

ATT-309 - Content may be mistransferred from a source that is specified with the `stdio://` URL if the session is delayed, which causes illegal, non-sequential reads from the `stdin` source.

ATT-189 - In rare cases, `ascp` keeps running after it encounters a disk read error. (CIM-233)

ATT-98 - If inline validation is configured on the server side, the server does not honor a session timeout if a transfer includes a skipped file.

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

ES-188 - Transfers through Aspera Forward Proxy are rejected if the node user password contains an @ symbol. (CIM-290)

ES-118 (#21517) - Folders created in the Connect Server web GUI may have permissions different from those specified in `aspera.conf`.

NODE-635 - The IAM policy of an AWS access key that uses IAM role authentication must include s3:GetObject permission to authorize the user to browse and upload to the storage.

NODE-244 - A POST request containing an invalid value for "storage_class" returns the wrong error message, "Invalid value for server_side_encryption".

NODE-188 - `alee-admin` times out if a node has more registered access-key entitlements than it can process in 30 seconds.

NODE-177 - [Unix-based OS] `ascp` transfers and `asperanoded` may fail when trying to transfer many (millions) of small files because the Redis db exceeds available number of file descriptors.

WAT-762 - When `async` is run in continuous mode and no initial scan (with `-C --no-scan`), `async` does not receive notifications of new files and directories. (CIM-1238)

WAT-606/WAT-557 - If Sync does not receive a notification that a new folder is created, files in that folder are not synchronized during a continuous Sync session. As a result, a continuous Sync push that is run with the `--scan-dir-rename` option does not synchronize files if the directory is created and then renamed after the Sync session has started.

WAT-594, WAT-589 (#13645), WAT-362 (#24812) - During a Sync session, if `ascp` fails to transfer a file, such as if a file is resized or a directory is renamed, Sync keeps the file as pending rather than as errored and the session does not stop. (CIM-814)

WAT-512 - The `--overwrite older` 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-501 - Some `ascp` sessions started by a Watchfolder may not stop running after synchronization is complete when many (50) large (1000 files of 2 KB to 1 MB) Watchfolders are started at the same time.

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

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-200 - Recently finished Watchfolder drops are not stored and are lost if asrund is restarted.

WAT-169 - When using `top_level_dirs` drop detection with *x* top level directories in Watchfolder, 7(*x*)+ drops are created. The drop count continually increases.

#32553 - When the FASP Session log source file list exceeds 500 bytes and contains multibyte UTF-8 characters, the output is truncated in a manner that creates an invalid UTF-8 sequence.

SYSTEM REQUIREMENTS

Server

- AIX 6.1, 7.1

Client Browsers for Connect-Enabled High-Speed Transfer Server Web UI

- **Linux:** Chrome 62-64, Firefox 56-58, Firefox ESR 52
- **Windows:** Chrome 62-64, Microsoft Edge 39-41, Internet Explorer 11, Firefox 56-58, Firefox ESR 52
- **macOS:** Chrome 62-64, Firefox 56-58, Safari 11, Firefox ESR 52

KNOWN ISSUES

Note: This release contains tickets that were created from different issue-tracking systems. For this reason, the list below uses two different formats for issue numbers.

General

ATT-332 - [Solaris, AIX, Isilon] The remote management stream does not report `ClientMacAddress` or `ServerDocroot` in the `SESSION` section.

ATT-245 (#22726) - Successful transfers might log the error, `Failure Event: -34 - libssh2_channel_wait_closed() invoked when channel is not in EOF state`, particularly downloads in FIPS mode. The error can be safely ignored. (CIM-329)

ES-926 - [AIX] As of 3.8.1, the `aswatchadmin*` and `asconfigurator` man pages are installed in zipped format and cannot be read. **Workaround:** To make these man pages readable:

1. Go to `/opt/aspera/share/man/man1` and unzip each zipped file (files that end with `.gz`). For example, to unzip `aswatchadmin.1.gz`:

   ```
   # gunzip aswatchadmin.1.gz
   ```
2. Go to `/usr/man/man1` and remove the incorrect symbolic links to the zipped files. For example, to delete the symbolic link to `aswatchadmin.1.gz`:

   ```
   # unlink aswatchadmin.1.gz
   ```
3. Create a new symbolic link in `/usr/man/man1` to the unzipped file. For example, to create a symbolic link to `/opt/aspera/share/man/man1/aswatchadmin.1`:

   ```
   # ln -s /opt/aspera/share/man/man1/aswatchadmin.1 aswatchadmin.1
   ```

ES-819 - GID is not preserved for files that are transferred using HTTP fallback. (CIM-1382)

ES-793 - If an HTTP connection cannot be established or errors for another reason, HTTP fallback transfers can take 3 minutes or longer to stop and report the error.

ES-675 - As of 3.7.4, in the rare case when a transfer fails with a "Session open failed" error, the status is not updated in the Aspera Central database from "running". As a result, Console continues to report the session as "RUNNING" until the entry in the central database is manually deleted or updated to "ERROR". (CIM-1072)

ES-664 - As of 3.7.4, `asconfigurator` no longer returns a warning to reload the Aspera Central service in order to activate a change in the server's configuration. (CIM-1037)

ES-610 - The Connect Server web UI displays duplicate headers and does not display symbolic link files if a symbolic link in the same directory is broken. **Workaround:** Correct the broken symbolic link and all files and symbolic links are shown correctly.

ES-323 - When doing a dry run of an `asdelete` (by using the `-d` option), the log shows all the files that were scanned, not the files that would be deleted by the `asdelete` command. (CIM-558)

ES-249 - The aggressiveness setting is applied to Vlinks, rather than only the network rate controller. (CIM-399)

ES-248 - While an `ascp` or `ascp4` transfer of many files is in progress, skipped files are reported as complete. The counters are correct once the entire session is complete. (CIM-398)

ES-216/ASCN-705 - If the Aspera Connect is unable to connect to the server through SSH, a misleading error message, "Failed to authenticate," is reported rather than indicating that it is a connection problem. (CIM-72)

ES-215 - If Aspera Connect is unable to connect to the server by SSH, HTTP fallback is attempted but only after a 15 minute delay. (CIM-320)

#35952 - `asunprotect` cannot decrypt a re-protected file.

#34811 - You are unable to download encrypted files with an incorrect decryption passphrase when you are using HTTP fallback.

#32934 - If the Internet accountability software Covenant Eyes is installed, some HTTP fallback transfers appear to complete but then lose connection with the server and then attempt to retransfer. Covenant Eyes captures the entire HTTP transmission before forwarding it to the server. If the file is so large that this process takes longer than about 20 seconds, the server times out and cancels the session. **Workaround:** Reduce the probability of timeout by increasing the server timeout length. Set `Session Activity Timeout` in `aspera.conf` by running the following command:

```
$> asconfigurator -x "http_server;session_activity_timeout,time_in_seconds"
```

#32517 - Retransfer requests are unencrypted when transfers are encrypted. This change in encryption can cause transfer failures in some scenarios, such as when a network device drops the retransfer request because it detects a bit sequence it considers malicious.

#31791 - Files with the file extension `.aspx` are not transferred. **Workaround:** Edit the `resume_suffix` setting in `aspera.conf` on the client.

#30690 - `ascp` fails with an inaccurate message—`Error: failed to authenticate`—when the server is configured to accept only unsupported ciphers.

#28679 - In some cases, the fallback server cannot accept additional connections, possibly due to too many 'incomplete' requests.

#27056 - `ascmd` does not respect server-side symlink configuration.

#21629 - Connect Server `aspera-dirlist.pl` does not accurately reflect file permissions for user actions.

Ascp

ATT-657 - Multi-session transfers that use `-k1`, or multi-session transfers from cloud storage to local storage, can result in files on the destination that are missing bytes. This occurs when a file is split between sessions; the first session creates matching attributes for the source and destination (partial) file, and the other sessions do not recognize that there are more bytes for transfer. **Workaround:** For multi-session downloads from cloud storage, use `--overwrite=always` (to prevent resuming file transfers) or `--multi-session-threshold=0` (to prevent file-splitting across sessions).

ATT-537 - Downloads that use `ascp --overwrite=always` fail when they are authenticated using ASPERA_LOCAL_TOKEN that specifies a local storage path.

ATT-435 - `save-before-overwrite` is not supported for URI

ATT-395 - When running a persistent `ascp` session, a FaspManager FILEERROR message truncates a filename that is longer than 128 characters to only the first 128 characters.

ATT-361 - `ascp` transfers to S3 fail when the `--symbolic-link=copy` or `--symbolic-link=copy+force` option is used.

ATT-360 and NODE-545 - Directory timestamps are not always preserved on the destination during an `ascp` transfer that uses `-p`.

ATT-226 - If the docroot is a URL path, `ascp` reports incorrect bytes for the sessions that are involved in a multi-session transfer.

ATT-185 - `ascp` does not reconnect to Redis database when `asperanoded` is restarted.

ES-645 - The `ascp -@` option is not supported when the destination is `stdio://`.

ES-626 - As of 3.8.0, `ascp` truncates JSON tags if the tags exceed 4 Kb. With this fix, tag length is checked before the transfer is started and an error is returned if the tags exceed 4 Kb.

ES-359 - `ascp` downloads from SoftLayer do not support `--move-after-transfer`.

ES-267 - Under rare conditions, `ascp` transfers to cloud object storage may be reported as successful even though Trapd reports an error and the content is not in the storage. (CIM-475)

ES-177 - The `range_low` value of a `-@` argument is not respected.

#35010 - If the source path in an `ascp` transfer is a file that is named \ (which is not supported by Aspera), the file is not transferred and an error is generated, but the folder then contains the file and all other files in that folder are transferred.

#32890 - During an `ascp` transfer that uses the `--preserve-xattrs= metafile --remote-preserve-xattrs=metafile` options, the metafile is not transferred.

#32680 - The option to create a directory (`ascp -d`) may create a directory at a destination before an expected session failure.

#30324 - During an `ascp` upload to cloud storage, if a mid-file read failure occurs on the sending computer (which is rare) it can cause the server-side `ascp` to crash and possibly fail to report transfer completion. This read failure can be caused when a source file is truncated during transfer, a drive or file system fails, or a transfer is canceled with `Ctrl+C` or other means.

#28939 - If command line `ascp` neglects to specify a destination host, then the failed transfer (error: "no remote host specified") gets recorded in `SQLite` with `client_node_id NULL`, instead of being populated with the `uuid` of the node. This database error causes an issue with Console.

#26281 - If you run approximately 100 (or a similarly high number) concurrent uploads to S3, intermittent transfer session failures can occur.

#26185 - During an upload to S3 storage, an error may result if `ascp` reports a successful file transfer before the transfer to S3 completes.

Ascp 4

ATT-659 - [Solaris, AIX] Ascp 4 transfers of many small files (for example, 10,000 10-Kb files) can stop abruptly due to running out of available memory on the client. **Workaround:** Use the `--memory=`*`bytes`* (and, if necessary, `--remote-memory=`*`bytes`*) argument to restrict the memory that is available for the process to within the OS limit (the default limit is 512 MB). For example, using `--memory=245M` prevented the error when transferring 10,000 10-Kb files. If you want maximum performance, you can set the LDR_CTRNL environment variable to increase the number of memory segments and override the MAXDATA parameter. For example, to use 3 memory segments, run the following command:

```
$ export LDR_CNTRL=MAXDATA=0x30000000
```

ATT-637 - As of 3.7.4, an empty folder is created on the destination even when the source folder is excluded from the transfer (by using `-E /`*`folder`*`/`*`pathname`*).

ATT-631 - [Solaris, AIX] Ascp 4 transfers stop and return the error "Peer aborted session... failed to close file *pathname*, e=18" when the destination is a symbolic link and the server has a partial file suffix configured (`<partial_file_suffix>` is set in `aspera.conf`).

ATT-621 - As of 3.8.0, Ascp 4 is not backward compatible.

ATT-583 - Ascp 4 does not automatically create a destination folder when the source is a file list and the destination does not exist. Instead, it writes all the files in the file list into one file. (CIM-1198)

ATT-582 - Ascp 4 sessions run with `-d` and a file list do not report an error if the destination already exists and is not a folder. (CIM-1199)

ATT-545 - Ascp 4 downloads all content from an AWS S3 docroot, rather than the specified content, if the docroot contains `?storage-class=REDUCED_REDUNDANCY`.

ATT-485 - Persistent session Ascp 4 downloads from object storage do not report a STOP message to management after the transfer completes.

ATT-477 - When files are transferred to a server with an S3 docroot and quickly retransferred with the `--delete-before-transfer` enabled, some files are deleted from the destination.

ATT-473 - Ascp 4 uploads to object storage that specify `-k 1` (resume if file sizes match) are also sensitive to checksum, such that if a file transfer is resumed and the file has the same size but a different checksum then the entire file is retransferred, rather than resumed from the last successful chunk.

ATT-451 - Ascp 4 does not respect exclude filters if the file path is part of the command line.

ATT-438 - Ascp 4 downloads from object storage fail if the source filename contains special Unicode characters, such as Japanese font.

ATT-428 - During a persistent `ascp4` session, when it a file transferred to a non-existant path and `-d` is used, the file transfers successfully and the destination path is created but the file is renamed to the last element of the destination path instead of being placed inside.

ATT-409 - If a file list contains an invalid path, no error is reported or logged.

ATT-338 - Parallel uploads of several large (>1 GB) files to object or HDFS storage may fail with the error "Peer aborted session" if the number of threads that are specified in the `ascp4` command exceeds the number of jobs that are allowed to run by Trapd. **Workaround:** Open `/opt/aspera/etc/trapd/trap.properties` and set the value for `aspera.session.upload.max-jobs` to one larger than the number of `ascp4` threads. For example,

```
# Number of jobs allowed to run in parallel for uploads.
# Default is 15
aspera.session.upload.max-jobs=50
```

ATT-186 - An `ascp4` multicast session does not fail if the multicast IP address and port is already in use on the receiver.

ATT-2 (#32295) - The default minimum transfer rate set in `aspera.conf` is not respected.

ES-247 - Console-initiated `ascp4` transfers fail if the docroot on the source is a UNC path (for example, `\\localhost\SHARE`), returning the error `ERR Source base/path is not a valid directory/file (doesn't match any source path).` (CIM-397)

ES-151 - `ascp4` does not recognize the UNC-path docroot of a Console transfer user. (CIM-197)

Node API

ES-505 - If the Aspera Central database cannot be reached by a Reliable Query request to `/services/rest/transfers/v1/sessions`, the response only includes a 500 Internal Server error and does not describe the error. (CIM-895)

NODE-686 - GET requests to /files/{id} return duplicate file entries for files with mixed-case file names (for example, aspera.txt is not duplicated, Aspera.txt is duplicated).

NODE-674 - `asnodeadmin`, `ascp`, and `asperanoded` hang when the Redis database schema is not up-to-date, rather than closing with an error.

NODE-626 - The response to a GET request to /ops/transfers does not return all stream-specific information for a transfer that is started by a POST request to /streams. The response also includes parameters that are not relevant to streams transfers.

NODE-619 - The Node API does not clean up transfer session information if the session was submitted with an invalid SSH port. **Workaround:** Clean the jobs manually by running the following command on the server:

```
$> asnodeadmin --transfer-log-del transfer_id
```

NODE-610 - The response to a GET request to /ops/transfers does not include the key-value pair `"use_ascp4":"`*`boolean`*`"`.

NODE-572 - As of 3.8.0, when you try to create an access key with a path to a subdirectory that does not exist, the error response does not report the invalid path.

NODE-571 - The Aspera Noded service cannot process more than 4,000 tags in a transfer request, which limits the number of files that can be moved and copied between folders in Aspera Files to about 50. (CIM-925)

NODE-492 - For transfers started by the Node API, the target directory is always created if it is not present, even when `"create_dir"` is set to false. (CIM-995)

NODE-481 - The Node service sometimes returns invalid JSON when Console polls `async` jobs. **Workaround:** Recreate the Redis database to resolve the issue. (CIM-988)

NODE-469 - When managing files with a request to `/files/{id}/files`, if the system user under whom the Node service runs does not have write permissions to the docroot, a 500 internal error is returned, even if the node user is attached to a system user who has write permissions.

NODE-466 - A POST request to `/ops/transfers` that contains an invalid hostname returns "waiting for 300 seconds" rather than an accurate error message.

NODE-463 - `ascp` transfers that use `--remove-after-transfer` do not report `file.deleted` events to the Node service.

NODE-460 - The Redis database grows large quickly when reporting Sync sessions to Console that frequently synchronize large directories. (CIM-936) **Workaround:** Reduce the number of files that are reported to Console or the retention time of data in the database.

NODE-442 - `/ops/transfers` can return a value of "null" for `files_failed`, which can prevent transfers from being displayed in Console. (CIM-864) **Workaround:** Enable the Console database to handle the "null" value by logging into the Console database and running the following command:

```
ALTER TABLE fasp_sessions CHANGE COLUMN files_failed files_failed INTEGER
 null;
```

NODE-437 - Transfers with object storage, particularly with buckets that contain a lot of data, become slow when `<files_filelock_enabled>` in `aspera.conf` is set to **true** (in order to enable the filelock feature in the Node API `/files` endpoint). The default setting is **false**.

NODE-433 - The value for `xfer_retry` that is submitted in a `POST` request to `/ops/transfers` is not respected. Transfers that retry but ultimately fail take a long time to be reported as inactive.

NODE-405 - The `max_rate_kbps` in the output of a `/events` call is incorrectly reported as zero.

NODE-392 - PUT requests to `/access_keys/`*`id`*`/storage` cannot locate the specified access key. **Workaround:** Submit updated storage specifications as a PUT request to `/access_keys/id`.

NODE-257 - Reports sometimes fail if the Node API temporarily reports an impossibly large value for `bytes_transferred`.

NODE-236 - Transfers with a status of "waiting" cannot be canceled.

NODE-231 - When a node-to-node transfer fails due to a transfer authentication error, the `GET /ops/transfers` response does not provide error information.

NODE-139 - The `--token-key-length` option in `asnodeadmin` allows invalid token key lengths.

NODE-137 - A Node API `/ops/transfers` call reports the incorrect values for `files_completed` and `files_failed`.

#33206 - `/ops/transfers` might briefly report pending transfers as `failed` when transfers are retried.

#32669 - When a directory is symbolically linked from a subdirectory, it does not appear in the search result for a `/files/search` request in the Node API.

Watchfolder and Aspera Watch Service

AC-517 - Pull Watch Folders are not visible in IBM Aspera Console because it uses an older version of the Watch Folder API.

WAT-812 - A pull Watch Folder becomes inactive when the remote Watch service hangs and cannot recover because the Watch service does not have a timeout.

WAT-810 - As of 3.8.1, when a Watch Folder is configured to pull from cloud storage, the Watchd service that is associated with it can crash under heavy transfer loads.

WAT-804 - If db_spec is set in both the <watch> and <rund> sections of aspera.conf, the setting for rund is not respected and the one for watch is used instead.

WAT-758 - The transfer token that is used for pull Watch Folders expires and is not automatically replaced, causing transfers to error. **Workaround:** Restart (disable and enable) the Aspera Watch Folder service (asperawatchfolderd) that is associated with the Watch Folder user.

WAT-674 - As of 3.8.0, some Watch Folder API endpoints still use "local" and "remote" terminology instead of "source" and target", which are used in the 3.8.0 Watch Folder configuration.

WAT-567 - A Watch Folder configured for growing files reports a "Healthy" state and shows bytes are written at the destination despite having an invalid password and no transfer occurring.

WAT-559 - Watchd allows users to create a watch on the root folder, which can overload the Redis database and cause Watchd to coredump. (CIM-662)

WAT-501 - Some ascp sessions started by a Watch Folder may not stop running after synchronization is complete when many (50) large (1000 files of 2 KB to 1 MB) Watchfolders are started at the same time.

WAT-314 - asperawatchfolderd must be running in order to delete a Watch Folder.

WAT-174 - Watch Folders uses excessive memory when it watches 10 million files.

WAT-159 - If one file in a Watch Folder transfer fails or a drop is aborted, the other files in the package are reported as aborted but ascp is not stopped and the transfer continues.

Sync

Async on AIX, Solaris, Mac OSX, does not support continuous PUSH or BIDI modes.

WAT-768 /ES-455 - A continuous Sync session might log several error messages stating "add watch failed" because it tries to access the service when one is already in use. This issue does not affect file synchronization. (CIM-806)

WAT-759 - In continuous synchronization mode, the UID and GID are not preserved even when --preserve-uid and --preserve-gid are used.

WAT-742 - When filelocks are enabled on the server and a push or bidi Sync session is run with basic token authentication, a local file deletion is not propagated to the server.

WAT-737 - Sync can hang on a fatal error, such as if the disk with the Sync database runs out of space, rather than aborting the session. (CIM-1075)

WAT-715 - The initial synchronization of directories in object storage is very slow.

WAT-700/ES-92 - Sync reports incorrect counts for 'deleted_paths', 'deleted_bytes', and 'cumulative_deleted_bytes'.

WAT-644 - If the local directory to synchronize is a symlink, the Sync session succeeds the first time but then fails the second time with the error, "ERROR: Error reading from peer (disconnected)". (CIM-897)

WAT-573 - Under rare conditions, Sync crashes after reporting the error message, "ERROR: Failed to reconcile with peer snap db".

WAT-550 (#29686) - A continuous Sync push to S3 storage does not update the object in S3 when the source file is renamed or deleted while it is transferred.

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

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 you sync 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.

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

#27621 - Hidden, temporary, or transient files, such as temporary files created by Microsoft Office products, can cause Sync to report conflicts.

#20906 - `async` cannot 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 - If you use the `-R log dir` from Unix-like OSes to Windows and there are spaces in the directory path, the path is truncated at the first space in the path.

#19945 - `asyncadmin` creates SHM and WAL files for read-only operations. Once asyncadmin 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 issue is due to a bug in SQLite.

#16911 - Characters in the `async` session option that are not preceded by a "`-`" or "`--`" are ignored and no error message is reported. Any session options that are specified (such as `-l` or `-a`) after the string of characters that are not preceded by a "`-`" or "`--`" are also ignored. The session runs using the default values, and does not notify you that the command line settings were ignored.

#13761 - If file names contain "\" or new line, `async` transfer fails, causing the internal transfer queue to become full and the synchronization to stall.

PRODUCT SUPPORT

For online support, go to the IBM Aspera Support site at https://www.ibm.com/mysupport/. To open a support case, log in with your IBMid or set up a new IBMid account.