

C&C Android Change List

3.15.1 -> 3.15.2

- Corrected an issue in HLS parsing causing unexpected NPE for some types of assets.

3.15.0 -> 3.15.1

- Fix issue when adding multiple HLS assets in close sequence where manifest parsing would become deadlocked. (Pulled forward from 3.14.29)

3.14.28 -> 3.15.0

- Fastplay.
- Language filtering for audio, CC, and subtitle content streams.
- Filtering audio streams by codec type.
- New, easy to use, Builder classes streamline asset creation.
- Ability to pause/resume individual asset downloads.
- Ability to add ancillary files to asset download and retrieve them from SDK.
- Improved DRM refresh handled in background tasks including new SDK setting to enable/disable automatic DRM refresh.
- Ability to manually refresh DRM for individual assets.
- Modernize SSL handling in SDK download connections.
- Enhancements to SDK analytics to provide deeper insight into usage.
- Added api methods to access the SDK secure clock information.
- Improved error reporting for asset permissions and manifest parsing.
- Improved asset expiration handled in background tasks.
- Support for large images in SDK notifications.
- Stability fixes around service restarts.
- Added ability to specify content destination path in SDK settings.
- Added ability to configure valid mime types for segments in SDK settings.
- Clarified documentation on SDK settings.

3.14.28 -> 3.14.29

- Fix issue when adding multiple HLS assets in close sequence where manifest parsing would become deadlocked.

3.14.27 -> 3.14.28

- Catch a crash that can occur if http proxy status is checked during service binding.
- Remove mime type checks for initialisation blocks in dash assets.

- Fix issue with default audio streams without separate URI being declared at top of manifest in HLS manifest parsing

3.14.26 -> 3.14.27

- Fixed an issue in backplane permission checking where registering an observer for Queue time permission checks did not ensure the permissions were checked.
- Fixed a timing issue that could cause crashes in the download engine if it was restarted and shutdown by the OS simultaneously. This only occurred when changing the setting for Maximum Download Threads.
- Cleaned up some missing resource releases for the ContentProviderClient.
- Corrected service restart processing to allow a smoother restart flow under conditions where the OS started the process following a force close or crash.

3.14.25 -> 3.14.26

- Fixed an issue where some cookie persistence to transfer cookies between http requests in different processes was causing crashing upon reloading.

3.14.24 -> 3.14.25

- Fixed issue where initial backplane validation on first startup fails if secure clock time has not been established. This was blocking registration from completing successfully in some circumstances.

3.14.23 -> 3.14.24

- Update MPEG DASH parsing to work with multi-period manifests where only start or duration elements are available for each period.

3.14.22 -> 3.14.23

- Update of feature which detects specific HTTP authentication error responses to only be activated upon a setting.
- Added a backup method for fetching time when NTP is blocked by the network or the server is unavailable. This connects to a separate backplane API to fetch the server time.

3.14.21 -> 3.14.22

Features:

- Added a minor feature to detect specific HTTP custom error responses and report a separate error code back to the API.
- Fixed an issue where the downloader can be marked as blocked on cell quota when cell quota is unlimited and the download reaches the storage quota.

3.14.20 -> 3.14.21

Features:

- Fixed a timing issue which can intermittently cause assets being deleted or expired while downloading to show download stopped reasons other than the true cause.
- Fixed the queue observer interface to make the error callback `engineEncounteredErrorDownloadingAsset()` for all permissions related error cases.
- Added the Max copies permission rule to the list of download stopped reasons so that it will be shown in the queue observer callback.

3.14.19 -> 3.14.20

Features:

- Fixed a number of threading and timing issues related to the content provider and settings which could result in intermittent ANRs during startup or background operations.
- Changed backplane sync scheduling logic to ensure a sync occurs at least once with every 24 hour period when the app is opened.
- Fixed issue which occurred if backplane comms were completing during sdk shutdown, which could result in an intermittent crash.

3.14.18 -> 3.14.19

Features:

- Fixed an edge case introduced in 3.14.18 that can cause long parsing times in certain types of manifests.

3.14.17 -> 3.14.18

Features:

- Fixed a number of small errors relating to correctly reporting failure if parsing errors occur while assets are being queued or if an asset is deleted while it is being queued. Also ensured that no artifacts are left in the database in these failure cases.

3.14.16 -> 3.14.17

Features:

- Fixed an error when saving very large HSS or HLS manifests to the database.

3.14.15 -> 3.14.16

Features:

- Corrected an issue where expires don't properly complete and the expiry handler begins to hard-loop attempting to expire the asset.

3.14.14 -> 3.14.15

Features:

- Corrected an issue where some internal error states for network errors could be reported as the max error state in the notifications for the asset.

3.14.13 -> 3.14.14

Features:

- Corrected an issue where some internal error states for network errors could be treated as external policy server permission errors.
- Reduced the restrictions on CnCForegroundServiceHandler::updateNotification() so that the service will accept updates to the cached notification when the service is not currently in foreground.

3.14.12 -> 3.14.13

Features:

- Corrected an issue where subtitles in some DASH manifests caused a parsing exception.

3.14.11 -> 3.14.12

Features:

- Corrected an issue where a security exception was not properly handled. The exception could occur in unusual circumstances when the SDK is starting its own services. In the event of the download service not starting, a method is called on any registered IEngineObserver to inform the main application.
- Corrected an issue where deleting the final item in the download queue whilst the item is blocked was not correctly resetting the download engine state to idle and clearing the notification.
- Added support for ttml+xml being recognized as a valid mime type for download in MPEG DASH manifests.

3.14.10 -> 3.14.11

Features:

- No new features, Corrected an issue where parsing of top level HLS manifests containing audio definitions with no url caused a crash.

3.14.9 -> 3.14.10

Features:

- Corrected additional configuration issues where invalid SSL configuration could be used on older Android devices.

3.14.8 -> 3.14.9

Features:

- Corrected an issue where invalid SSL configuration could be used on older Android devices.
- Corrected an issue where foreground service notifications could still be shown for particular use cases.
- Fixed an NPE error happening on some devices when the playback proxy service took too long starting up.

3.14.7 -> 3.14.8

Features:

- Corrected an issue where closed captions in DASH assets wouldn't download

3.14.6 -> 3.14.7

Features:

- Corrected an issue where service notification may be shown when app restarts
- Corrected an issue where SDK may crash if multiple assets parse simultaneously

3.14.5 -> 3.14.6

Features:

- Adding setting to configure download service to exit foreground mode and remove system notification when the download engine is paused, which defaults to disabled.

3.14.4 -> 3.14.5

Features:

- Added codec filtering support and multiple codec support for MPEG-DASH assets.

3.14.3 -> 3.14.4

Features:

- No SDK Changes. Corrections were made to SDK Demo build configuration.

3.14.2 -> 3.14.3

Features:

- Added ability to select which audio codecs are downloaded when multiple audio codecs are included in the same master HLS manifest.
- Added ability to support multiple simultaneous audio codecs in a master HLS manifest.
- Added new permissions model “max copies of asset per account”.

3.14.1 -> 3.14.2

Features:

- Added API/workflow to properly handle asset creation failures due to app suspension

3.14.0 -> 3.14.1

Features:

- Improved release packaging to support gradle/maven based builds
- Insure setting battery level to 0 completely disables battery checks
- Updates to packaged proguard rules to insure clean building with minify enabled without external rules

3.13.6 -> 3.14.0

Features:

- Improved download engine performance
- Added API for manually reporting play events
- Updates to support latest ExoPlayer versions for DRM support and exposing MediaDrm.OnEventListener via API
- Added support for external policy service permissions model
- Added setting to allow limiting maximum concurrent connections
- Added support for creating downloads directly from Uplynk asset data URLs
- Fix to insure notifications are updated when changes occur while downloads are paused

3.13.1 -> 3.13.6

Android 8 Support:

- Update service to use Foreground service when downloading in the background. Download service switches in and out of foreground mode when necessary. Application needs to provide the notification to display for progress updates.
- Update to use explicit broadcasts throughout SDK to support Android 8.
- Replaced Apache HTTP stack with current default Android URLConnection for main downloader and OkHttp for REST server calls.
- Update monitoring of disk space / memory / CPU usage to support Android 8.

Features:

- Support for subtitles in HSS
- Maximum Asset Download feature to limit maximum number of downloads per asset.
- Increase progress reporting to UI observers for large segments during download.
- Efficiently handle HTTP 416 responses for partial file requests
- Improve local file management to handle HLS files containing multiple folders containing files of identical names.
- Expiry of content during download if necessary.
- Added adaptive multi threading framework for downloader to optimise number of threads used for files with large segment counts and prevent file and network access from throttling download speed. Makes use of additional cores in newer devices. Greatly improved download throughput.

Bug Fixes:

- Improved calculations for fraction complete, available through asset manager.
- Fixed issue with support for really large manifests.
- Improve logging levels throughout SDK to reduce size of log files required for issue reporting.
- Prevent partial download and range requests on key file downloads to prevent encryption issues.
- Handle { and } characters in urls within manifests

3.13 -> 3.13.1

New Features:

The SDK now allows checking of the Max Account Downloads and Max Asset Downloads at the time of adding an Asset to the Queue. The “REQUIRE_PERMISSION_ON_QUEUED” Setting is governed by the backplane and comes down in the Sync.

In order to check the permission at time of addition to queue there is a new Observer Interface defined in IQueue:

```
interface IQueuedAssetPermissionObserver {  
    /**  
     * Callback to report on Asset's permission check when added to the queue.  
     * If the interface is provided to one of the Queue's Add methods then  
     * it will be called regardless of the RequirePermissionWhenQueued Setting.  
     * If an asset was not granted permission then the Asset will have its download status  
     * set to the relevant error. Additional details can be retrieved from the Asset; it  
     * holds the Backplane's permission response.  
     *  
     * @param aQueued True if asset was added to the queue.  
     * @param aDownloadPermitted True if Asset was granted download permission. False otherwise.  
     * @param aAsset The asset which was granted / denied download permission.  
     * @param aAssetPermissionError one of the IAssetPermission.PermissionCode values  
     * @see IAsset#getLastPermissionResponse()  
     * @see IAssetPermission  
     * @see com.penthera.virtuososdk.client.IAssetPermission.PermissionCode  
     * @category 003_queue  
     */  
    void onQueuedWithAssetPermission(boolean aQueued,  
                                     boolean aDownloadPermitted,  
                                     IAsset aAsset,  
                                     int aAssetPermissionError);  
}
```

Overloaded Add methods have also been added to the IQueue interface to allow passing in the IQueuedAssetPermissionObserver. To get the details of the backplane response and a listing of devices which have registered a download for the asset use the IAsset interface and call the getLastPermissionResponse Method.

The IAssetPermission contains an interface that details the account devices and number of Downloads registered against them (IDownloadsPerDevice and IAssetDownloadsPerDevice).

3.13 -

New Features:

The SDK now uses Firebase Cloud Messaging (FCM) instead of Google Cloud Messaging (GCM). The SDK now supports Amazon Device Messaging (ADM). Please see the Setting up Push Notifications section in the [CCClientDevelopersGuideAndroid.pdf](#), that section provides details on what needs to be specified in the manifest. The SDK Demo application also shows how to implement and subclass the SDK components.

Support for the “Maximum Permitted Asset Downloads per Account” (MAD) rule. This rule is governed by the backplane. It can be read from the `com.penthera.virtuososdk.client.IBackplaneSettings` using the method `getMaxDownloadsPerAsset`. This rule controls the maximum number of times an asset can be downloaded across all devices associated with a user.

Support for the “Maximum Permitted Downloads per Account” (MDA) rule. This rule is governed by the backplane. It can be read from the `com.penthera.virtuososdk.client.IBackplaneSettings` using the method `getMaxDownloadsPerAccount`. This rule controls the maximum number of assets a user can have downloaded across all associated devices.

The SDK now allows Segmented Assets to complete with Errored segments. You can access and modify the number of segments which are permitted to have errors through the `ISettings` interface. Look for the methods: `setMaxPermittedSegmentErrors` and `getMaxPermittedSegmentErrors`

The Client HTTP proxy for playing out Segmented files will by default return a HTTP OKAY (200) and a zero length file for segments which have errored out during download. Depending on your player you may need to return a different HTTP code for playout to continue; You can modify the code through the `ISettings` interface. Look for the methods: `setSegmentErrorHttpCode` and `getSegmentErrorHttpCode`.

API Changes:

Class `com.penthera.virtuososdk.client.Virtuoso` The startup method’s signature has changed. The `GCMSenderId` parameter has been removed.

Interface `com.penthera.virtuososdk.client.IPushRegistration` This interface is passed as a parameter into the startup method. The `onRegisterSuccess` method has been removed. The `onRegisterFailed` method has been removed. The `onServiceAvailabilityResponse` method’s signature has been modified. It now takes 2 integer parameters that specify the type of push service and an error code. Please look at the SDK Demo and java docs for full details.

Interface `com.penthera.virtuososdk.client.IEngineObserver` There are now 2 types of setting change callbacks: One for settings provided by the backplane `backplaneSettingChanged` and the other for Settings configurable by the client `settingChanged`. A new set of flags `com.penthera.virtuososdk.Common.BackplaneSettingFlag` has been added for usage with the `backplaneSettingChanged` callback.

Interface `com.penthera.virtuososdk.client.IAssetManager` Signature of the `createFileAsset` method has been modified: the expected size parameter has been removed.

Class `com.penthera.virtuososdk.Common.AssetStatus`

Three new constants have been added to describe the status of an Asset:

- **DOWNLOAD_DENIED_ACCOUNT**: Asset can not be downloaded due to the Max Downloads Per Account rule.
- **DOWNLOAD_DENIED_ASSET**: Asset can not be downloaded due to the Max Downloads Per Asset rule.
- **DOWNLOAD_BLOCKED_AWAITING_PERMISSION**: Asset can not be downloaded until permission has been verified with Backplane.