

C&C Android Change List

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.