4.2.40

- Allow DRM with mismatched CIDs to continue playing, but with warning.
- Better handling of poorer quality NTP servers.

4.2.39

- Log NTP server and address when trusted time is received.
- Use a rotating queue of NTP time servers for lookups, retry if invalid time received.

4.2.38

- Disallow successful, but invalid NTP results for time source.

4.2.36-37

- Better automatic handling for AVContentKeySession-based DRM with 3rd party players.

4.2.35

- Properly handle embedded forward slashes in URL query parameters when calculating manifest hash

4.2.34

- Provide interface to access downloaded ancillary files via HTTP proxy URL.

4.2.33

- Fast-fail and reset download if continuing previous download via byte-range request results in a different expected file size than previously reported.

4.2.32

- Improves download start times for non-segmented assets
- Fixed issue accessing FairPlay methods on simulators

4.2.31

- Disable MD5 validation if HTTP etag header doesn't contain valid MD5 hash string.

4.2.30

- Disable manifest hash calculations for non-segmented assets
- Corrects race condition occasionally leading to multiple downloads of the same segment.

4.2.29

- Properly handle embedded forward slashes in URL query parameters in proxy requests and manifest rewriting

4.2.28

- Make all access of HTTP headers case insensitive

4.2.27

- Fixes edge case VirtuosoSecureClock issue resulting in certain devices being unable to authenticate with Penthera Cloud.

4.2.26

- Fixes an edge case causing segment size mismatches and md5 issues during some segment downloads. Some improvements to download performance.

4.2.20-25

- Validate segments on disk against server hashes, when possible.

4.2.19

- Expose the VirtuosoAsset used for DRM licensing to custom AVAssetResourceLoaderDelegate implementations via VirtuosoLicenseManager. Formally deprecated asset and loadPersistentLicense properties in the VirtuosoAvAssetResourceLoaderDelegate protocol.

4.2.18

- Catch exceptions bubbling up from AVContentKeySession under certain error conditions.

4.2.14-17

- Optimization, performance improvements, and bug fixes when transitioning downloads between background and foreground operation.

4.2.13

- Fixed a race condition introduced in 4.2.12 in VirtuosoClientHTTPServer that may prevent back-to-back playback sessions from starting the second playback properly.

4.2.12

- Updated to support FairPlay licensing servers that do not provide unique skd:// urls per asset.

4.2.11

- Updated to support third party players that deeply embed AVContentKeySession FairPlay handling.

4.2.10

- Updated PlayAssure parser not to skip GROUP manifest entries without URI
- Added fallback to use GET requests when calculating manifest hash

4.2.9

- Added filters to reduce noise in analytics data.
- Corrected some analytics measurements.

4.2.8

- Added M1 Simulator slice to the build. No other changes.

4.2.7

- Fixed crash when push notices are received that are not for this SDK
- Improved FastPlay performance and analytics

4.2.6

- Fixed download performance issues with HLS segmented assets and separate audio tracks.
- Improved VirtuosoDownloadEngineNotificationsManager API

4.2.5

- Fixed unhandled exception when processing HLS manifests in a particular format.

4.2.4

- Fixed memory access bug when reporting playback event

4.2.3

- logPlayback has been removed from the SDK and will be automatically invoked
- Engine enabled state maintained
- Timestamp to track playback added to LogAssetsViewes
- Handle gzip content-encoding and content-length errors
- Calculate estimatedSize correctly for single-segment assets

4.2.2

- Fixed bug in fastPlay for HLS single segments
- New Asset class method to retrieve assets for a DRM license path (assetsWithDrmLicensePath)
- Rendition selection delegate provides VirtuosoAsset including new Example17 tutorial
- Improved manifest hash calculation for play_session_stats event

4.2.1

- play_session_stats event
- Master manifest added to download_requested event
- Fixed bug to ensure callback for deleteAssetOnComplete called after asset deleted.
- Introduced new mechanism for filtering downloads
- Removed need for asset property in default AVAssetResourceLoader
  Relaxed requirement that IFRAME RESOLUTION match selected rendition resolution
- Fixed bug where Subtitle and Language video rendition settings can create unplayable download
- Prefer average-bandwidth for bitrate selection
- Add ability to select rendition by resolution/quality rather than just bitrate

4.2.0.3

- Correct use of I-FRAME when IFRAME segments are not contained within the main video binary

- Added an option to override SDK video rendition selection process by allowing a delegate for manual rendition selection.

4.2.0.2
- Asset configuration filter for allowable resolutions
- Change I-FRAME selection with differing video resolutions
- Use AVERAGE-BANDWIDTH when available for selecting video rendition

4.2.0.1
- Fix AVContentKeySessionDelegate check

4.2.0
- Playlist API changes to introduce new Playlist type (FastPlay)
- App_launch event changed to indicate whether SDK build was debug or release.

4.1.4
- Fixes race condition where FairPlay license metadata does not get saved to asset and fix assets with missing license metadata.

4.1.3
- Fixes Playlist append issue when Playlist was not created via PlaylistManager

4.1.2
- Changes Engine startup to remove all assets when user, keys, or backplaneURL changes. Previously only removed all assets when user changed.
- Playlist Tutorial updated to show append examples

4.1.1
- Asset deleteAll now will trigger any pending auto-downloads, previously it did not.
- Playlist support for expiry after download if the asset was played.
- Tutorial updates

4.1
- Playlists
- Swift package manager support
- Reduced memory footprint
- Automatic injection of play events (play_start, stop, progress)
- Support for Akamai AMD token authentication
- DASH Asset deprecation on iOS
- SAM Beta
- The names of the release products have changed to reflect updated build configurations. The VirtuosoClientDownloadEngine.xcframework supports bitcode, but lacks Widevine support. This is the recommended version for most customers. The VirtuosoClientDownloadEngineWidevine.xcframework version supports

Widevine but not bitcode, and *should only be used if you require Widevine*.

4.0.3

- Lazy initialization of all subsystems to minimize footprint. VirtuosoSubscriptions must now be specifically enabled using setting: [VirtuosoSettings.instance setSubscriptionsEnabled:TRUE];
- Asset.resolution field added and will be populated with value of video resolution (if present in the manifest). Format: asset.resolution = "800x400" where 800 represents the width, and 400 the height.
- AVOD support for Verizon/Freewheel
- Asset method clearRetryCountOnComplete renamed clearDownloadRetryCountOnComplete
- Asset class property retryLimit renamed downloadRetryLimit
- Asset instance property maximumRetriesExceeded renamed maximumDownloadRetriesExceeded
- Added new "delete list of assets" API to allow efficient and immediate reporting of batched deletions. This API should be used whenever more than one asset is being deleted concurrently.

4.0.2

- Explicitly set file protection for NSSQLiteStoreType to NSFileProtectionNone to enable background downloads to properly continue when app has Data Protection set.

4.0.1

- Fastplay download improvements

4.00

- New API's make Asset creation easier
- Playlists for related Videos supporting smart-downloads
- Ad Supported Videos (Beta)
- Easy to use Delegates for Engine status notifications
- Improved threading cooperation
- Download performance improvements
- Support for additional HLS tags
- New Tutorials to help boot-strap
- Improved analytics reporting
- Logging improvements
- Simplified Engine startup options
- Device externalID configurable
- HLS, HSS, DASH language filtering
- Reparse downloaded asset to include additional language options
- Vastly simplified implementation requirements for AppDelegate
- All settings now use private store, previously used NSUserDefaults

3.15.15

- Fixed infrequent crash due to mutation of a collection during downloading.

3.15.14

- Added support for additional CMAF asset metadata in HLS manifests. Properly parse byte-range for EXT-X-MAP segments. Properly include manifest metadata that may appear after the EXT-END-LIST tag.
- Fixed edge-case issue causing long "finalization" processes for certain types of HLS asset downloads (byte-range-based HLS).
- Fixed infrequent crash due to mutation of a collection during indexing.

3.15.13

- Added support for DRM subtypes, which allows different DRM license sources for different assets in the platform.
- Added FairPlay DRM support for proprietary players.

3.15.12

- Changed internal method handling Segment IO errors to fail gracefully and recover.
- Changed network state change listener to correct logic error that was causing UI events to be processed out of sequence.
- Corrected incorrect download size reporting on HLS Single Segment files
- Corrected Device unregister to post notification when errors happen

3.15.10/11

- Fixed an error where the download engine may get too aggressive retrying downloads if unexpected responses to download permissions requests are received.

3.15.9

- Fixed final internal CoreData concurrency issue.

3.15.8

- Fixed internal CoreData concurrency issues.
- Fixed an issue that could cause some single-file HLS style assets to incorrectly report the asset as completely downloaded after transitioning to foreground download from background download.
- Fixed an issue where failure to retrieve the registered device list for an account could cause the SDK authentication to report as invalid.

3.15.7

- Added option to disable built-in advertising integration with AVPlayer. To disable this integration, add "[NSUserDefaults.standardUserDefaults setBool:TRUE forKey:@"VFM_DisableAVPlayerIntegration"];" to the beginning of your AppDelegate applicationDidFinishLaunching method.
- Extended VirtuosoLicenseProcessingDelegate API to pass the VirtuosoAsset object the licensing request is for in all methods.

3.15.6

- Add additional FairPlay built-in licensing customization API.

3.15.5

- Catch all exceptions during backplane sync and log analytics event for failure.

3.15.4

- Updated to support iOS 13.1 builds with latest XCode 11.
- Don't ask FairPlay DRM handler to extract CKC if server returned errors, when configured with built in Virtuoso FairPlay delegate.

3.15.3

- Added new API to allow external ID changes without unregistering and re-registering the device.
- Added additional information to analytics events to provide better system behavior analysis.
- Support new setting to require DRM license retrieval prior to starting asset parsing.
- Added some guards to prevent potential for multiple instances of AVAssetResourceLoaderDelegate subclasses during DRM processing.

3.15.2

- Added additional analytics events to track FastPlay usage.

3.15.1

- Don't refresh Widevine DRM licenses once a good license has been received. FairPlay DRM licenses will auto-renew according to the design spec.
- Added undocumented option ([NSUserDefaults.standardUserDefaults boolForKey:@"com.virtuoso.globo.isSimultaneousAccessLimitCheckEnabled"]) to treat HTTP 401 errors as a unique kVirtuosoDownloadEngineErrorTooManySimultaneousDownloads error instead of a generic network error.

3.15.0

- Support for ancillary files (custom files which can be added to be downloaded with the video).
- Merged VirtuosoClientSubscriptionManager framework into VirtuosoClientDownloadEngine framework.
- Overall system improvements to improve main thread performance during active downloading.
- Improved automatic DRM refresh
- Added option to disable automatic DRM refresh
- Overall performance and stability improvements
- New (optional) delegate callback patterns to support easier integration
- Support for HLS IFRAMES
- Improved/added analytics event information.

3.14.17

- Extended the VirtuosoLicenseManager API to allow use of custom headers

3.14.16

- Limit the number of assets that can parse simultaneously in order to improve performance when user starts download of many assets in quick succession.
- Corrected an issue that could cause downloads to stall at the end of the download process.
- Corrected an issue where network failures during asset parsing might cause a download to fail but appear successful, resulting in the inability to play back the asset. Such errors are now reported in the parse completion callback.

3.14.15

- Fixed a minor performance issue that slowed down asset parsing a bit.
- Corrected an issue that caused critical memory issues when using FairPlay assets in the simulator.

3.14.14

- Fixed an issue with certain types of assets where iOS temporary download files would not get immediately cleaned up upon asset deletion in some use cases.

3.14.13

- Adjusted the (optional) API that allows developers to modify URLs to be used to download sub-manifests, encryption keys, and segments during asset parsing.

3.14.12

- Fixed Release build to work with Google Widevine Release libs
- Made mechanism for enclosing application to change URLs at parsing more versatile.

3.14.11

- Disabled packager, in favor of new optimized background download engine.
- Added optional mechanism for the enclosing application to add additional parameters to segments created during parsing.

3.14.10

- Catch exceptions due to settings in VirtuosoSettings being changed while remote wipe or unregister is occurring.

3.14.8-3.14.9

- Restart any existing proxies when the app enters the foreground so that any players that were running with offline content can be resumed without issues.

3.14.7

- Weak linked the widevine libraries to make them optional.

3.14.6

- Fixed parse issue when master manifest contains query strings.

3.14.5

- Separated MPEG-DASH support into separate build product. If you require MPEG-DASH, bitcode cannot be enabled due to security restrictions. If you are not using MPEG-DASH, you can use the "-Bitcode" version of the frameworks, which disables MPEG-DASH support and has bitcode enabled.

3.14.2-3.14.4

- Corrected some edge-case issues with playback of offline content
- Fixed an issue where remote unregister request may not be honored if sent from a different device than the one being unregistered.

3.14.1

- Corrected some memory related issues that could occur during indeterminate network states.

3.14.0

- Improved performance of asset "delete" methods.
- Improved error notifications when asset parsing issues are encountered.
- Added support for downloading only desired audio codecs.
- Added support for only downloading desired audio/CC languages.
- Added optional support for HLS IFRAME playlists.
- Added support for system-wide "maximum copies of asset per account" download permissions logic.
- More efficient processing of non-packager background downloads.

3.13.49-3.14.50

- Removed libstdc++ from required library list.
- Minor updates to fix some URL encoding issues encountered when master HLS manifest contains relative URLs and URL parameters.

3.13.48

- Fix for issue where occasionally download will not resume when transferring from non-packager background download to foreground download

3.13.47

- Support for parsing/handling Uplynk asset references directly
- Skip content length checks for any segments returned with compressed content encodings

3.13.42-3.13.46

- Minor internal bug fixes

3.13.41

- Internal modifications to pass the error code and message from an external policy service through to the enclosing app. The external policy server response is now passed back in the userInfo dictionary of the error that is posted.
- Increased priority of task queue that handles starting download requests in order to prevent stalling of downloads when other system functions are under load