Apple VisionableSDK v1.3.6 Release Notes (January 8, 2025)

CHANGES/FIXES

Update logging to open file once and flush content after full log message has been written to filestream

V1.3.5 RELEASE NOTES:
CHANGES/FIXES

Updated Audio and Video library dependencies
Corrected Network Stats types and added streamId
Added configuration capability to Audio and Video

V1.3.4 RELEASE NOTES:
CHANGES/FIXES

Fix imports for Audio and Video conditions

V1.3.3 RELEASE NOTES:
CHANGES/FIXES

Fix for device info updates during join to meeting
Relative zoom instead of absolute for CAM520

V1.3.1 RELEASE NOTES:
API CHANGES

Added new callbacks for audio and video network conditions.

```
public func audioConditionUpdate(audioCondition: AudioCondition)
public func videoConditionUpdate(videoCondition: VideoCondition)
```

Added dedicated objects for conditions data:

`AudioCondition` – represents general audio network condition and contains data for audio streams
`AudioStreamCondition` - represents audio stream specific network conditions
`VideoCondition` - represents general video network condition and contains data for video streams
`VideoStreamCondition` - represents video stream specific network conditions

CHANGES/FIXES

No changes

KNOWN ISSUES

Same as in v1.3.0

V1.3.0 RELEASE NOTES:
API CHANGES

Added APIs to allow for an "Image Capture" device. This is a device that the application "creates" with an API call by specifying a directory to which image files can be written to (via a new API call) and from which the underlying video engine can read image files to be sent up into a meeting.

```
public func enableImageCapture(displayName: String,
                    directory: String, mode: String)->Int32
```

Asks the SDK to create a new image device. The `displayName` parameter is the name that will appear in the corresponding `VideoInfo siteName` field for this stream. The `directory` parameter is an absolute path to a directory on the local device that can be written to. This absolute path **must** contain a trailing directory separator. The `mode` parameter is a screen sharing mode to be used for this stream (such as "BEST SCREEN").

Returns an integer ID to be used with other API calls that need to reference this device. Successful execution of this API call will generate an immediate `participantVideoAdded` callback for this user in all applications connected to the meeting.

```
public func disableImageCapture(deviceId: Int32) -> Bool
```

Disables a previously created image capture device. The `deviceId` parameter is the identifier returned by the corresponding call to `enableImageCaptureDevice` (which created this capture device). Returns a `boolean` indicating whether or not the call was successful.

```
public func imageCapturePutImage(deviceId: Int32, data: Data,
                width: Int32, height: Int32, size: Int32) -> Bool
```

Send a YUV420P image into the meeting for the specified device. The `deviceId` parameter is the identifier returned by the call to `enableImageCaptureDevice` the application used to create the capture device being used. The `data` parameter is a Data (NSData) object containing an unpadded YUV420P image. The `width` and `height` parameters are the width and height of the image, respectively. The `size` parameter is the size of the Data object being passed in.

Returns a Boolean indicating whether or not the image was successfully received.

CHANGES/FIXES

Miscellaneous Audio/Video engine fixes

KNOWN ISSUES

When screen sharing from iOS and placing the application in the background, iOS may suspend the app (and cause screen sharing to be paused) if the iOS device is running low on system resources.

When sharing a window into the meeting, the remote user may see the share freeze if the shared window is resized while being actively shared.

The new `previewVideoUpdated` delegate method may *not* be called when a device rotates. This will be resolved in a future SDK release.

In support of the new threading model, all delegate methods are executed on a serial OperationQueue that is created by the SDK. Future versions will allow you to specify an OperationQueue that you create (or use the main queue)