Android VisionableSDK v1.3.10 Release Notes (March 26, 2025)

CHANGES/FIXES

Fixed crash during network conditions processing
Fixed issue with Android rotation for Android versions < 12
Enhance NAT traversal logging for IGAudio in IGVideo
Update SDK logging to close file on name change

V1.3.9 RELEASE NOTES
CHANGES/FIXES

Update of dependency libraries.
Network statistics codec profile name fix

V1.3.8 RELEASE NOTES
CHANGES/FIXES

Fixed crash on during join to meeting

V1.3.7 RELEASE NOTES:
CHANGES/FIXES

Correct AudioStreamCondition and VideoStreamCondition member types

V1.3.6 RELEASE NOTES:
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
Fixed issue with displaying camera buffer on Android device rotation
Added configuration capability to Audio and Video

V1.3.3 RELEASE NOTES:
CHANGES/FIXES

No changes, bump version to match other platform SDKs

V1.3.1 RELEASE NOTES:
API CHANGES

Added new callbacks for audio and video network conditions.

```
public void audioConditionUpdate(AudioCondition ac);
```

public void videoConditionUpdate(VideoCondition ac);

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

Minor fixes in jni layer for typing (int32_t and int64_t objects).

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 static int enableImageCapture(String displayName,
                          String directory, String mode)
```

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 static boolean disableImageCapture(int deviceId)
```

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 static boolean imageCapturePutImage(int deviceId,
     byte[] imageBuffer, int width, int height, int size)
```

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 `imageBuffer` parameter is a byte array 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 byte array being passed in. Returns a Boolean indicating whether or not the image was successfully received.

CHANGES/FIXES

Fixed bug that prevented background blurring from persisting when a device is rotated.

KNOWN ISSUES

While the ability to specify a dedicated Looper upon which all delegate methods are invoked, the current Android MeetingSDK does not attempt to create a dedicated Looper if one is not specified. This will result in all delegate method calls being made on the same thread being used to parse low-level audio and video events coming from our audio/video engine.