# Android VisionableSDK v1.2.5 Release Notes (September 4, 2024)

**API CHANGES**

None

**CHANGES/FIXES**

Fixed bug that prevented the “blurring” flag from being properly passed when enabling video preview.

**V1.2.3 RELEASE NOTES (NO v1.2.4 RELEASE FOR ANDROID)**
*API CHANGES*

None

*CHANGES/FIXES*

Fixed bug that mistakenly allowed H.261 as a video send resolution option.

**V1.2.2 RELEASE NOTES:**
*API CHANGES*

None

*CHANGES/FIXES*

Further fixes to reporting accuracies of network quality for all participants

**V1.2.1 RELEASE NOTES:**
*API CHANGES*

None

*CHANGES/FIXES*

Miscellaneous fixes in underlying audio/video engines.

More accurate reporting of network quality for all participants

**V1.2 RELEASE NOTES:**
*API CHANGES*

None

*CHANGES/FIXES*

Fixed bug that might cause incoming screen shares to not be displayed properly (no frames rendered)

**V1.1.1 RELEASE NOTES:**

*API CHANGES*

None

*CHANGES/FIXES*

Fixed obscure bug that could have caused some problems when implementing the participantNetworkQuality delegate method.

**V1.1 RELEASE NOTES:**

*API CHANGES*

None

*CHANGES/FIXES*

When notifying the RTN service about the devices available to the user in a meeting, properly communicate whether or not a video device is enabled and what its encoding is.

Miscellaneous fixes in underlying audio and video engines

**V1.0.2 RELEASE NOTES:**

**API CHANGES**

None

**CHANGES/FIXES**

Fixed a bug that prevented the ModeratorSDK::sendPTZCommand() from functioning properly when remote PTZ commands are allowed.

Fixed a generic problem with sending JSON payloads to the RTN service

**KNOWN ISSUES**

none

**V1.0.1 RELEASE NOTES:**

*API CHANGES*

none

*CHANGES/FIXES*

Fixed a crash that resulted when attempting to enable a video preview.

*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.

**V1.0 RELEASE NOTES:**

*NOTE*:
This is the first major release of the Visionable SDK. It contains three main modules:

- MeetingSDK
- ModeratorSDK
- VisionableAPI

These are all represented as separate singletons bundled in an Android aar file named MeetingSDK.aar. All APIs are documented in our readme.io-based documentation repository. The release notes below will cover changes made to the original MeetingSDK singleton; please consult the readme.io repository for information on new singletons (ModeratorSDK, VisionableAPI)

*API CHANGES*

Please consult the Android MeetingSK v1.0 Porting guide for a complete list of changes to APIs from the last v0.x release.

```
public static boolean enableVideoPreview(String deviceName, String
resolution, Boolean enableBlurring)

public static boolean enableVideoPreview(String deviceName, String
resolution)
```

A new, overloaded version of the enableVideoPreview API call has been provided that takes a third argument. This argument is a boolean which, if true, will enable background blurring on preview frames. If you do not wish to have blurring on preview frames, either pass `false` for this parameter OR use the original 2-argument version.

*CHANGES/FIXES*

Any functionality associated with calling Visionable's web service APIs has been moved into the new ModeratorSDK singleton.

When connecting to a V3 or later server, the SDK now establishes a web socket connection to Visionable's RTN service for the purposes of information updates from the Visionable back-end as well as out-of-band commands that may be sent to the client from a meeting Moderator or for the purposes of receiving PTZ commands (for supported cameras) from remote users.

*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.