Android MeetingSDK v0.5.4 Release Notes. (October 30, 2022)

NOTE:
Version 0.5.x of the Visionable Anddroid MeetingSDK is a transitional release that begins to expose functionality available in Visionable's "V3" server architecture. As this architecture is not expected to be in production until 4Q2022, any 0.5.x releases of the MeetingSDK are likely to be unstable and change frequently.

If you are looking to write an application that interfaces with Visionable's V2 architecture, you should remain with v0.4 of the MeetingSDK.

We expect that stability will be established against the V3 architecture with v0.6 of the SDK due when the V3 architecture officially goes into production.

OVERARCHING CHANGES (this text present for ALL v0.5.x releases)
Starting with v0.5, the Visionable MeetingSDK has been re-architected to rely on a layer of cross-platform C++-based code to manage the parsing of XML objects coming from our audio/video engine, for establishing model objects representing Meetings and Participants, and for firing "delegate methods/callbacks" notifying your application of changes in state for the current meeting. Prior to v0.5, these functions were individually implemented per-platform supported in that platform's native language. Moving all of this functionality into a common, C++ codebase should result in consistent behavior when dealing with Visionable back-end servers.

CONNECTING TO V3 SERVERS (this text present for ALL v0.5.x releases)
In Visionable's V3 architecture, a special token (referred to as an MJWT token) is required to join a meeting. There are two types of MJWT tokens: a *guest* MJWT token that doesn't correspond to Visionable user and an *authenticated* MJWT token that is obtained by passing a JWT token obtained from Visionable's authentication system (not covered here). To retrieve an MJWT token, use the new `initializeMeetingWithToken` API call *instead of* the original `initializeMeeting` API call (which is used only with V2 servers).

```
public static void initializeMeetingWithToken(
        String token,
        String server,
        String uuid,
        IInitializeMeetingCompleteCallback initDoneCallback
)
```

This function still takes a `uuid` and a `server` name but now also takes a `token` parameter that is either `NULL` if you wish to obtain a guest MJWT or it contains a JWT token if you want to obtain an authenticated MJWT.

The completion routine for `initializeMeetingWithToken` now is called with a second parameter (String) that contains the MJWT token (guest or authenticated).

Once you obtain an MJWT, you now join the meeting with a call to joinMeeting (same call for either V2 or V3 servers):

```
public static int joinMeeting(
        String name,
        String userUUID,
        IJoinMeetingCompleteCallback joinDoneCallback)
```

The `initializeMeetingWithToken` function caches the `server` name you are connecting to, the `uuid` for the meeting and the MJWT needed to connect to the meeting. When calling `joinMeeting`, you only need to pass the `name` of the user to be shown in the meeting as well as the optional `userUUID` to be associated with this user (pass an empty string to have the SDK generate a UUID for this user). This behavior is slightly different than the APIs in the MeetingSDK for non-Android platforms, and the Android MeetingSDK will likely be refactored in the future to behave in a manner similar to other platforms.

Using these two calls will allow you to connect to a V3 meeting. Once connected, all other SDK functionality is the same as with V2 servers.

CONNECTING TO V2 SERVERS

The APIs for connecting to V2 servers are the same as they were for the v0.4 release of the Android MeetingSDK.

See previous release notes in the v0.5.x series for API changes that were new in previous releases. The rest of these release notes pertain only to the v0.5.3 release.

API CHANGES

New APIs have been added for supporting video preview. The following MeetingSDK API calls have been added:

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

Starts a video preview session with the specified camera. The camera `deviceName` parameter should be one of the values returned by `getVideoDevices`. The `resolution` parameter is one of the standard supported resolutions (same ones used for the `enableVideoStream` API call). When the video preview has started, you will be notified via a new `INotificationCallback` routine, documented below.

Note: While, in theory, it should be possible to begin a video preview session while you are already in a meeting, the behavior or attempting to preview the back camera while the front camera is being used for a meeting (or vice versa) has not been verified.

```
public static boolean disableVideoPreview(String deviceName)
```

Stop video preview events from the device specified which should be one of the strings returned by the `getVideoDevices` API call.

The following new `INotificationCallback` method has been added:

```
default void previewVideoViewCreated(VideoView videoView) {}
```

Called whenever a new preview video view is ready to be used. The `VideoView` object passed to this interface method is one that can be embedded in your user interface.

Additionally, all interface methods in `INotificationCallback` have been given default definitions (which do nothing) which means that you only need to implement the methods in `INotificationCallback` that you are interested in.

CHANGES/FIXES

Underlying fixes were made to memory management with respect to playing sounds with the `playSound` API call.

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.