

iOS CLIENT SDK

GETTING STARTED GUIDE

REVIEW HISTORY

DATE	VERSION	REVIEWERS	STATUS
7-JUN-2022	1.5.0	George Sealy, Gaurav Rihan, Nishanth Shettar, Ankita Rani, Ruairi Griffin	Complete
4-MAY-2022	1.4.0	George Sealy, Gaurav Rihan, Nishanth Shettar, Ankita Rani, Ruairi Griffin	Complete
10-FEB-2022	1.3.0	George Sealy, Gaurav Rihan, Nishanth Shettar, Ankita Rani, Ruairi Griffin	Complete
9-DEC-2021	1.2.0	George Sealy	Complete
1-NOV-2021	1.1.0	George Sealy, Gaurav Rihan, Nishanth Shettar	Complete
20-SEPT-2021	1.0.0	George Sealy, Gaurav Rihan, Nishanth Shettar	Complete

Table of Contents

OVERVIEW.....	3
PRE-REQUISITES.....	3
WHERE DO I START?	3
INTEGRATING THE SDK	3
MANUAL INTEGRATION	3
SPM INTEGRATION.....	3
HOW TO UPGRADE FROM A PREVIOUS VERSION OF THE SDK	4
EXPLORING THE SDK	4
USE CASES	4
SUPPORTED FEATURES	4
OBSERVING SDK STATE.....	6
ACCESSING VIDEO STREAMS	7
CUSTOMIZATIONS.....	7
WHAT ABOUT FEATURES SUPPORTED IN THE DESKTOP/MOBILE VERSION OF BLUEJEANS?	8
LOGGING	8
SAMPLE APPS	9
TRACKING & ANALYTICS	9
KNOWN CAVEATS	9
LOOKING FOR ADDITIONAL HELP?	9
CONTRIBUTING	9
LICENSE	10
LEGAL REQUIREMENTS.....	10
FREQUENTLY ASKED QUESTIONS	10



Overview

The BlueJeans iOS Client SDK provides a collection of frameworks that expose APIs for audio and video conferencing meetings, together with easy to follow sample apps and extensive documentation. With this SDK you can bring audio, video, chat, screen share and much more to your native iOS apps and create brand new use cases for your customers.

Pre-requisites

SDK Version	1.5.0
OS Support	iOS 13 and later
Hardware Support	Any iOS device running iOS 13 or later
Supported Languages	Swift 5.4 and higher (Xcode 12.5 and up)
Simulator Support	Yes. M1 Macs need to run under Rosetta

XCode 12.5 or later is required to build an app using this SDK. The SDK is built with module stability to ensure compatibility with future updates to Swift and Xcode.

Where do I start?

To get going, you need two things; this SDK and a valid BlueJeans meeting ID / passcode. If you do not yet have a meeting, you can:

- Sign up for a BlueJeans account, either a [trial](#) or [paid](#) account.
- Once you have an account, you can schedule a meeting via <https://bluejeans.com> or using [direct API calls](#). To enable API calls on your account, please [Contact Support](#).

Integrating the SDK

Full documentation, README and Sample apps can be found on GitHub [here](#). Once you have a meeting ID, follow the steps below.

Manual Integration

- Download the binary `xcframeworks` from [here](#).
- To integrate the SDK, simply embed the BlueJeans `xcframeworks` in your app – a step by step example is given in the README file.

SPM Integration

- Use the Swift package manager (SPM), adding the package (version 1.5.0) from <https://github.com/bluejeans/ios-client-sdk>.
- You have a choice to add either a device-specific or simulator specific package (due to differences in the frameworks used). A single package that supports both device and simulator will be available in a future release.



Some additional setup that you must be aware of:

- Bitcode must be disabled in your build settings.
- Permission strings for microphone and camera access must be added to your `Info.plist` file.
- Additionally, if you require your app to remain active while in the background, you need to add the Background Modes -> Audio capability.
- To add simulator support, some of the frameworks will need to be conditionally added to your app, only for device builds. Details are in the README file and a helper script for this is also available in the Scripts folder of the SDK (*Scripts/embed-device-only-framework.sh*).

How to upgrade from a previous version of the SDK

Updates to new versions of this SDK will be published on GitHub with change logs. Upgrading is as simple as replacing the frameworks with newer versions or updating via SPM.

Exploring the SDK

Use Cases

- Inheriting BlueJeans Meeting experience inside your iOS app.
- Allowing users with or without a BlueJeans account to join meetings on your iOS app.

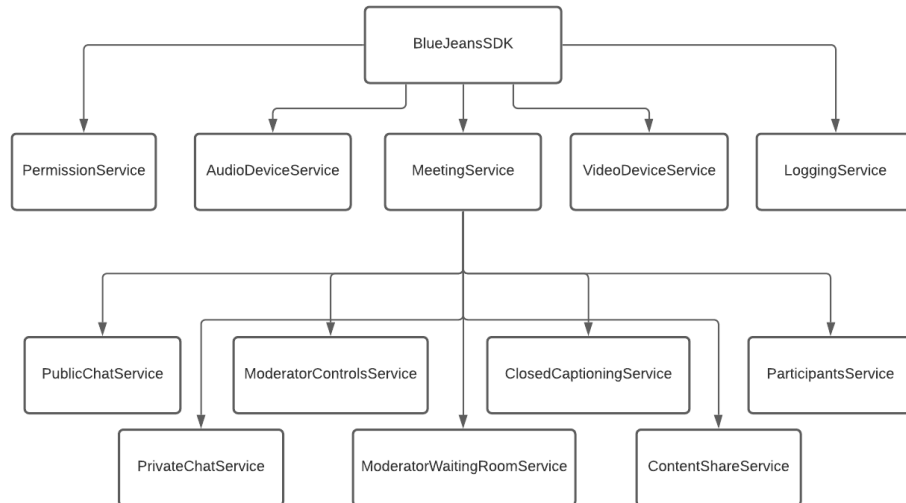
Supported Features

Supported Video Resolution	Send up to 720p video and receive up to 720p.
Supported Frame Rate	Up to 30fps for video, 5fps for content.
Join Meeting	Join a meeting with meeting id and moderator/participant passcode.
Leave Meeting	Leave a meeting (and optionally end the meeting as a moderator).
Self Audio/Video Mute	Participant can mute / unmute either their audio or video.
Select Audio/Video Device	Toggle between front and back camera. Audio devices can be selected through standard iOS UI, including Bluetooth and standard external microphones / headphones.
Initiate Screen Share	Participants can share their screen.
Receive Content/Screen Share	Content / Screen shares received from other end points can be displayed.
List of Participants	A list of all participants currently in the meeting and their current audio/video states. Information about the active speaker.
Layout Support for Remote Video	Ability to change between different styles of layout. See “Accessing Video Streams” below.
Supported Meeting Capacity	Number of participants supported in a meeting is based on the plan and differs for Trial, E-COMM & Enterprise customers. More details here .
Video Rotation	Video rotation between portrait and landscape orientations is handled automatically.
Retrieve SDK Version	Allows a developer to programmatically retrieve the SDK version, for logging or support purposes.

Public & Private Chat	Ability for meeting participants to send text messages during the meeting, either publicly or privately.
Moderator Controls	<p>As a moderator of the meeting, user can –</p> <ul style="list-style-type: none"> a) Start/Stop recording. By default, announcements will play on all endpoints when recording is started or stopped. b) Mute/Un-Mute Audio of other participants / all participants c) Mute/Un-Mute Video of other participants / all participants d) Remove a participant from the meeting <p>End meeting for all immediately or after a certain configurable delay (in seconds)</p>
Waiting Room support	<p>As a participant, ability to observe, that I'm being put into the waiting room and transition in to/out of the meeting, based on moderator's choice.</p> <p>As a moderator, ability to moderate meeting's waiting room with following functionality -</p> <ul style="list-style-type: none"> a. Turn the waiting room on/off. b. Admit a waiting room participant into the meeting. c. Admit all waiting room participants into the meeting. d. Deny a waiting room participant entry into the meeting. e. Deny all waiting room participants entry into the meeting. <p>Demote a meeting participant into the waiting room.</p>
Closed Captioning	Ability to turn-on or turn-off transcription of the meeting audio as it occurs.
Enable/Disable Media Stream	<p>An ability for a user to disable their video, audio and content streams. Once disabled, only the user can enable their video, audio and content streams and the moderator of the meeting does not have control over it.</p> <p>This can be used to power scenarios where user is provided a choice, to move into low bandwidth consumption in case of travel or low network zone, or to provide a content only mode.</p>

The SDK provides a single top-level class, *BlueJeansSDK*, that provides access to all available functionality. Within this class, APIs are grouped into Service objects for convenience.

BlueJeansSDK exposes an *initialize()* method that can be used to configure options such as support for 5x5 remote video views in Gallery layout (see below). Note that it is *not* necessary to call *BlueJeansSDK.initialize()*, but if you wish to alter the default configuration this must be done by calling *BlueJeansSDK.initialize()* before any other SDK methods. See the Customizations section below for more detail.



SDK Architecture and listing of various services

APIs are exposed through service objects like *MeetingService*, *LoggingService*, *PermissionService* and so on.

Properties or methods of these service objects can be accessed directly, for example:

```
if BlueJeansSDK.meetingService.meetingState == ...
```

It may also be convenient to maintain local references to these objects, for example:

```
let meetingService = BlueJeansSDK.meetingService
...
if meetingService.meetingState == ...
```

Observing SDK State

The current state of your BlueJeans meeting is exposed as a collection of properties that are observable. Some examples include:

- `MeetingService.meetingState` // Reflects the connection status of a meeting
- `MeetingService.audioMuted` // Is the user's microphone muted?
- `PublicChatService.chatHistory` // An array of chat messages received
- `LoggingService.loggingMode` // How much detail are we logging?

App level logic or UX can subscribe to changes in this state as follows:

```
meetingService.videoMuted.onChange {  
    print("Video muted changed to ",meetingService.videoMuted.value)  
}
```

More examples can be found in the detailed README and sample apps.

Accessing Video Streams

Video Streams can be requested from the VideoDeviceService:

- `VideoDeviceService.getSelfView()` returns a `UIView` containing the local camera view.
- `VideoDeviceService.getContentShareView()` returns a `UIView` containing any received content (screen shares) from other participants.
- `VideoDeviceService.getRemoteVideoController()` returns a `UIViewController` containing remote video views. It supports several layouts:
 - **Speaker** – just the currently active speaker is shown.
 - **People** – one large video for the active speaker and up to 5 smaller views for recently active or pinned participants.
 - **Gallery** – 9 views (or 25 if enabled) of recently active participants.

For more details, check out the README file.

Customizations

Calling `BlueJeansSDK.initialize()` before any other SDK methods allows a developer to override default behaviour.

720p support

- With the 1.4.0 release, both sending and receiving video can be configured to allow 720p capture for highest video quality. The initialize function takes a `BlueJeansSDKInitParams` struct with two Boolean flags for enabling 720p send and receive.

Video backgrounds customization

- With the 1.4.0 release, developers can configure the background for incoming video streams. The initialize function takes a `BlueJeansSDKInitParams` struct with a `CustomizationParams` struct which can supply custom colors / gradients for video tiles, audio tiles and the background of the remote video view.

Enable/Disable Media Stream

- If you have a use case that doesn't need either remote video, audio or content, it is possible to disable these streams and reduce bandwidth and CPU usage as a result. This can be done with the `setRemoteVideoMuted`, `setRemoteAudioMuted()` and `setContentMuted()` methods on the `MeetingService` object.



5x5 Support

- Enabling up to 25 remote video endpoints to be visible at one time in gallery layout. The default is set to 9. The initialize function takes a BlueJeansSDKInitParams struct with a Boolean flag for enabling 5x5 support.

Recording Announcements

- The SDK initialize function takes a BlueJeansSDKInitParams struct with a Boolean flag `playIVRs` to disable this behavior if desired.

Screen Share behaviour

- Configure screen share behaviour in the following ways:
 - Specify if a participant can begin screen sharing while another participant is already screen sharing (*called as "hijacking" the screen share*), or if an App level confirmation is needed first.
 - Customize the message shown to a user if their screen share is stopped for any reason (*reasons could be "hijacking by another participant" or "moderator removing the screen share privilege" etc*).
 - Refer to the ContentShareGuide for specific steps.

What about features supported in the Desktop/Mobile version of BlueJeans?

Use of the iOS Client SDK in conjunction with other BlueJeans clients like Desktop app, BlueJeans browser client or BlueJeans Mobile apps, may result in a limited experience for SDK client users as some of the advanced features that are supported on BlueJeans developed clients are not yet supported via the iOS Client SDK.

Following is a list of features that are not supported via the IOS Client SDK:

- ❖ **Breakout Sessions** (*ability for SDK client to move in and out of Breakout Session*)
- ❖ **Restricted Meetings** (*ability for SDK client to trigger unique passcode for participants to join*)
- ❖ **Telehealth Meetings** (*ability for SDK client to join Telehealth enabled meetings*)

Logging

The SDK records logs that are kept for approximately a week (less if logging is set to verbose). `LoggingService` exposes a method to upload these logs to BlueJeans. Once uploaded, logs will be available to the BJN team in the private log server under the username folder.

In order to trace the logs corresponding to an upload, the BlueJeans SDK team will need the username and time at which the upload was done.



Update: In cases where audio issues (e.g. echo or feedback) are being tracked, it is also possible to use `LoggingService.enableAudioCaptureDump()` to capture detailed logs that BlueJeans can use to diagnose the issue. Note this should only be turned on for a short duration (30-60s if possible), as the logs produced are large.

Sample Apps

The SDK includes two sample apps:

- **Hello BlueJeans** – a simple app designed to provide working code samples for SDK features.
- **Screen Share** – an app that shows how to create an iOS Broadcast Extension to allow screen sharing in a BlueJeans meeting.

Tracking & Analytics

BlueJeans collects data from app clients who integrate the SDK to join BlueJeans meetings such as Device information and usage data. The App Store requires this to be declared as part of the App submission process – Specifically, it is suggested that you identify that the Localized Device Model, Device Model and OS Version Identifiers are collected and shared with BlueJeans and Mixpanel.

Known caveats

None

Looking for additional help?

For any queries or issues related to iOS Client SDK, you can raise support case via our [Contact Support page](#).

Contributing

The BlueJeans iOS Client SDK is closed source and proprietary. As a result, we cannot accept pull requests. However, we enthusiastically welcome feedback on how to make our SDK better.

If you think you have found a bug, or have an improvement or feature request, please file a GitHub issue or [Contact Support](#) page and we will get back to you. Thanks in advance for your help!



License

Copyright © 2022 BlueJeans Network. All usage of the SDK is subject to the Developer Agreement that can be found [here](#).

Download the agreement and send an email to api-sdk@bluejeans.com with a signed version of this agreement, before any commercial or public facing usage of this SDK.

Legal Requirements

Use of this SDK is subject to our [Terms & Conditions](#) and [Privacy Policy](#).

Frequently Asked Questions

[Do my customers need to have BlueJeans account to use a BlueJeans SDK integrated app?](#)

No. The integrated app should allow customers to join meetings without needing a BlueJeans account.

[Can I schedule a meeting using the iOS Client SDK?](#)

No, scheduling a meeting can be achieved through our public RESTful API functions, more detail can be found [here](#).

[Can I chat to a person outside of a meeting?](#)

No. Chat can only be done within a meeting.

[Will my meeting attendees be required to download anything?](#)

No. It's not required for the user to download anything from BlueJeans. The app that you create using the iOS Client SDK is all that is needed.

[What is the maximum participant count in a meeting or video conference?](#)

The maximum number of participants supported on your meeting is based on the plan and differs for Trial plan, various E-COMM plans & Enterprise plans. More details [here](#).

[Does the BlueJeans SDK support chat bots?](#)

No. Chat bots are not supported as of now.

[Is this BlueJeans SDK free to use?](#)

BlueJeans iOS Client SDK can only be used with an active BlueJeans subscription. There are no extra charges for using the SDK.

[Am I able to customize the look of the remote video views?](#)

Yes. The 1.4.0 release supports custom colors and gradients for backgrounds.

