Class

ImmersiveMediaRemotePreviewSender

An observable object that helps an app send the required data to all connected receiver apps to help facilitate the complete preview of the immersive media playback.

macOS 26.0+

```
class ImmersiveMediaRemotePreviewSender
```

Overview

The apps have to implement the protocol `ImmersiveMediaPreviewMessagingProtocol` so that messages exchanged between sender and receivers are in the correct format for processing. The apps need to provide this as custom protocol when establishing the network connection between the sender and the receivers.

It's also important to configure the NWParameters to be secure by adding TLS security options as the example below.

```
func setupNWParameters() -> NWParameters {
    let tcpOptions = NWProtocolTCP.Options()
    let tls = /* setup TLS security options */

    let ret = NWParameters(tls: tls, tcp: tcpOptions)

    let options = NWProtocolFramer.Options(definition: ImmersiveMediaPreviewMessagir
    ret.defaultProtocolStack.applicationProtocols.insert(options, at: 0)

    return ret
}
```

```
let parameters = setupNWParameters()
let browser = NWBrowser(for: .bonjour(type: serviceType, domain: nil), using: parame
```

```
let parameters = setupNWParameters()
let listener = try NWListener(using: parameters)
```

Topics

Initializers

`init(networkParameters: NWParameters?) async throws`

Creates a preview sender using the specified network parameters, if any.

Instance Properties

`var connectedReceiverNames: [String]`

An array with the names of all receives currently receiving data from this instance. When a receiver goes offline, this array will be automatically updated.

`var isReadyToSendData: Bool`

A Boolean value that indicates whether this preview sender is ready to send data.

`var preferredFrameRate: Int`

The preferred frame rate to use when sending and previewing frames. This is optional; a value of −1 enables the system to decide the best framerate based on network quality.

`var preferredVideoHeight: Int`

The preferred video height to be used when sending and previewing frames. This is optional; a value of –1 enables the system to decide the best resolution.

`var preferredVideoWidth: Int`

The preferred video width to use when sending and previewing frames. This is optional; a value of –1 enables the system to decide the best resolution.

Instance Methods

`func connectReceiver(name: String, endpoint: NWEndpoint) async throws`

Adds an `ImmersiveMediaRemotePreviewReceiver` to the sender as an active participant of the network preview. Any updates on the sender will be propagated to all active receivers (frames, camera information, static metadata).

`func disconnectReceiver(name: String) async`

Disconnects a specific remote preview receiver associated with the name provided when `connectReceiver(name:endpoint:)` was called.

`func send(audioBuffer: CMSampleBuffer) async throws`

Sends an audio frame to all connected receivers.

`func send(taggedBuffers: [CMTaggedBuffer], presentationTimeStamp: CMTime, frameDuration: CMTime) async throws`

Sends a video frame to all the connected receivers using its tagged buffers representation.

`func send(venueDescriptor: VenueDescriptor) async throws`

Sends a venue descriptor to all connected receivers.

`func send(videoBuffer: CMSampleBuffer) async throws`

Sends the video frame to the receivers.

`func send(videoFrame: ImmersiveVideoFrame, presentationTimeStamp: CMTime, frameDuration: CMTime, metadata: [PresentationCommand]) async throws`

Sends a video frame to all the connected receivers using its sample buffer representation.

`func sendVenueDescriptor(at: URL) async throws`

Sends an AIME to all connected receivers.

`func start() async`

Starts the sender.

```
func stop() async
```

Stops the sender - all current connected receivers will be disconnected and streaming will stop.

Relationships

Conforms To

```
Copyable
Observable
Sendable
SendableMetatype
```