

ClearVR SDK for Unity - API v10.1.3 - 2023-08-22





Namespace com.tiledmedia.clearvr

Classes

ABRLevel

The ABRLevel object contains information about the currently active ABRLevel.

AudioDecoder

The audio decoder is the device that decodes audio samples before playback. Note that this class is not of any use right now, and is merely provided as a boiler-plate concept for the future.

IMPORTANT

One is encouraged NOT to use any of these APIs.

AudioPlaybackEngine

The audio playback engine is the device that routes decoded audio to an output device. Note that this class is not of any use right now, and is merely provided as a boiler-plate concept for the future.

IC IMPORTANT

One is encouraged NOT to use any of these APIs.

AudioTrackAndPlaybackParameters

This class is used to signal the audio track index that one wants to play. Typically used in conjunction with SwitchContentParameters. Notes.

• It is strongly recommended to keep the AudioDecoder and AudioPlaybackEngine arguments at null. This will allow the SDK to select the optional audio decoder and playback engine respectively for the current platform.

AudioTrackInfo

Helper class that holds the info of one audio track like codec, sample rate and number of channels. For detailed information about the ContentInfo, FeedInfo, VideoTrackInfo, AudioTrackInfo, SubtitleTrackInfo objects and their relationship, refer to theContentItem documentation.

CallCoreResponse

CheckViewportBounds

Component used to verify if the camera is within the expected bounds. If not the camera will move to the closest allowed pose. This component has to be reloaded each time a new content start to be

displayed. Calling initialize with the current ClearVRPlayer is enough to reload the component.

ClearVRAsyncRequest

IP IMPORTANT

This is an internal class that should not be used. Public access will be removed after 2020/12/31

ClearVRConstants

ClearVRDisplayObject

This class has been removed in v9.0 and can no longer be used.

ClearVRDisplayObjectControllerBase

ClearVRDisplayObjectControllerMesh

ClearVRDisplayObjectControllerOVROverlay

Holds the ClearVRDisplayObjectController gameobject which is rendered on an OVROverlay

ClearVRDisplayObjectControllerSprite

Holds the ClearVRDisplayObjectControllerBase component attached to the gameobject which render the Sprite

ClearVRDisplayObjectControllerUnmanagedMesh

The ClearVRDisplayObjectControllerUnmanagedMesh is to be used to render a video on a custom mesh, NOT managed by the ClearVRPlayer. This control properly update the shader parameters, in sync with the video frame currently displayed on the object, but does not interact with the mesh itself besides attaching the proper Material (and associated Shader) to it.

You should subscribe to the Clear VRD is play Object Events to be notified of essential Display Object life-cycle events.

A WARNING

This is considered an advanced Display Object. It is fundamentally incompatible when playing ClearVR content.

ClearVRDisplayObjectEvent

Since v9.0 The event as emitted by a Clear VRD is play Object.

ClearVRDisplayObjectEvents

The ClearVRDisplayObjects Events listener prototype.

ClearVRDisplayObjectEventsInternal

ClearVREvent

The primary interface for handling events from the ClearVRPlayer in your own application. As soon as you have created your CLearVRPlayer object, one shuuld subscribe an EventListener through clearVRPlayer.clearVREvents.AddListener(this.02);

ClearVREvents

The ClearVREvents listener prototype.

ClearVRLayoutManager

For details on the ClearVRLayoutManager and how to use it, please refer to the documentationhere. On runtime, you never access the ClearVRLayoutManager directly, you always interface with the ClearVRPlayer instead.

ClearVRMessage

Each ClearVREvent contains a ClearVRMessage. The ClearVRMessage contains valuable information about the event. Please refer to ClearVRMessageCodes for details on how to interpret the various codes. The code field can hold a value of eitherClearVRMessageCodes or ClearVRCoreErrorCodes Use the GetIsClearVRCoreErrorCode() API to figure our whether the code is a ClearVRCoreErrorCode or ClearVRMessageCode. Please refer to the GetClearVRMessageCode() API to convert the Integer into its ClearVRMessageCode and GetClearVRCoreErrorCode() API to convert the Integer into its ClearVRCoreErrorCode respectively.

ClearVRPlayer

The ClearVRPlayer class is the primary entry point, also refer to ClearVRPlayer for a detailed description.

ClearVRPose

This class has been removed in v9.0 and can no longer be used.

ClearVRScale

Object describing the scale of an object.

ClearVRSubtitle

Helper class that holds the info of one subtitle.

ClearVRViewportAndObjectPose

This class has been removed in v9.0 and can no longer be used.

Contentinfo

Helper class that holds the info of one content item. For detailed information about the ContentInfo, FeedInfo, VideoTrackInfo, AudioTrackInfo, SubtitleTrackInfo objects and their relationship, refer to the ContentItem documentation.

Contentitem

This object is used to load an initial clip and to switch to a new clip.

ContentItemList

Convenience array of ContentItems

ContentSupportedTesterParameters

Parameters when testing whether a (or multiple) clip can be played back or not.

ContentSupportedTesterReport

The TestIsContentSupported API returns a ContentSupportedTesterReport object upon completion. When parsing, one should first verify that the test was a success by calling GetIsSuccess() on the argClearVRMessage argument in the callback.

- Success == true the contentItemList will hold an array of the sameContentItem as you specified in the ContentSupportedTesterParameters argument of the original test call, as you passed those along by reference. Note that, even in case of a successful test, a ContentItem can still have a ContentSupportedStatus 'Unknown' because not all content item types are checken. Currently, only ClearVR clips (manifest.json) are tested.
- Success == false one can check the argClearVRMessage argument for details on why it has failed.

DeviceParameters

Helpers class holdig device-specific parameters.

DisplayObjectMapping

Parameters to specify the mapping between a DisplayObject and a feed

DRMInfo

This helper class specifies information on a clip's DRM protection. This class supports token and certificat-based decryption, but DRM protected content playback is only support on Android.

FallbackLayout

Advanced API, do not use.

FeedInfo

Helper class that holds the info of one feed. For detailed information about the ContentInfo, FeedInfo,

VideoTrackInfo, AudioTrackInfo, SubtitleTrackInfo objects and their relationship, refer to the ContentItem documentation.

FishEyeSettings

Describes the exact fish eye lens parameters. A number of pre-defined lenses are available, see FisheyePresets.

LayoutParameters

Parameters to specify the feed to display object mapping for the SetFeedLayout API This object has a mutable builder pattern.

Logger

Class that handles the ClearVR and App logging.

LoggingConfiguration

This class is used to configure the log level and logging output of each component of Tiledmedia SDK. Do not forget to call ClearVRPlayer.EnableLogging(loggingConfiguration) to apply the configuration.

A WARNING

Take special care to keep only the default constructed instance of this class in the ClearVRPlayer to avoid performance impacts

OVROverlayOptions

Holds OVROverlay playback specific settings.

PlatformOptionsAndroid

Android platform specific platform options. Currently, there are NO additional options/fields to set.

PlatformOptionsBase

Configuring your ClearVRPlayer object can be achieved using the platformOptions class.

A WARNING

Reading or changing any field *after* calling ClearVRPlayer.Initialize() is not allowed and can result in undefined behaviour. The only exception to this rule is when using the applicationRegainedFocusDelegate delegate to customize player behaviour during an application suspend/resume cycle.

PlatformOptionsIOS

iOS platform specific platform options. Currently, there are NO additional options/fields to set.

PlatformOptionsPC

PC (Linux/Windows) platform specific platform options. Currently, there are NO additional options/fields to set.

PlaybackParameters

PopulateMediaInfoParameters

The PopulateMediaInfoParameters class is used in conjunction with the PopulateMediaInfo() API.

PrepareContentParameters

Parameters used to prepare a ClearVRPlayer for content playout.

PrewarmCacheParameters

Specifies what content to pre-load. This class is deprecated and can no longer be used. Cache prewarming has been removed in v8 of the SDK

ProxyParameters

Helper class to define HTTP/HTTPS proxy parameters.

SeekParameters

Parameters to specify seek operation. Note that frame accurate seek is not guaranteed.

SharedPointersWithSDK

StatisticsBase

StatisticsBase.AudioStatisticsBase

StatisticsBase.VideoStatisticsBase

SubtitleTrackInfo

Helper class that holds information of one subtitle track. For detailed information about the ContentInfo, FeedInfo, VideoTrackInfo, AudioTrackInfo, SubtitleTrackInfo objects and their relationship, refer to the ContentItem documentation.

SwitchContentParameters

Parameters required for switching to a different ContentItem.

SyncSettings

This object is used for advanced configuration of the livestream sync feature.

SyncStateChanged

SyncStatus

TelemetryConfiguration

This object contains the configuration of one or more Telemetry targets. SeeTelemetryTarget for details. You would set this on platformOptions.telemetryConfiguration

Example:

```
// Define telemetry target configuration (in this example for a NewRelic end-point)

TelemetryTargetConfigNewRelic telemetryTargetConfig = new

TelemetryTargetConfigNewRelic("YOUR_ACCOUNT_ID", "YOUR_LICENSE", "YOUR_END_POINT");

// Configure your telemetry target (you can have multiple)

TelemetryTarget telemetryTarget = new

TelemetryTarget(TelemetryIPSignallingTypes.TelemetryIpSignallingMasked, new

List<TelemetryTargetConfigBase>() { telemetryTargetConfig });

// Set the telemetry configuration.

platformOptions.telemetryConfiguration = new TelemetryConfiguration(new List<TelemetryTarget>() { telemetryTarget });
```

Telemetry Target

Configure your Telemetry Target(s) of a specifictype.

TelemetryTargetConfigBase

Abstract base class for TelemetryTargetConfigurations. All Telemtry Targets should extend this base class.

TelemetryTargetConfigNewRelic

The Telemetry Target configuration for New Relic metrics aggregation. See NewRelic.com for details.

TelemetryUpdateCustomData

Used to send custom key/value pair data to the configured Telemetry Targets. Refer to TelemetryUpdateTargetCustomData for details.

TelemetryUpdateTargetCustomData

Data fields to set send custom data to the telemtry target of choice.

TimingParameters

The TimingParameters object is used when loading content, seeking and switching content. It defines the (new) start position (in milliseconds) and how this should be interpreted.

TimingReport

This object holds detailed information about the position in the currently playing clip, as well as the lower and upper seek bounds. This object is especially useful when interested in the distance to the

live edge in case of LIVE clip. Interpretation of the various timing fields should be based on the flag field.

I IMPORTANT

The timing information on this object is only useful if the GetIsSuccess() API returns true. In case of false, the timing fields should be ignored as their values and meaning are undefined.

TrackID

TrackID is an object containing the feed index and the track index of the track (video, audio or subtitle) The feed index tells you to which feed this track belongs (which can be found with clearVRPlayer.mediaInfo.getContentInfo()) The track index tells you which track this is in the list of tracks (useful when there are multiple video / audio / subtitle tracks) within the feed.

Note that this object in immutable after construction.

Utils

The utils class provides various convenience methods at your disposal. Public APIs on this class will maintained.

VideoDecoderCapabilities

Helper object that is of little use to the integrator. Allows one to parse the ClearVREventTypes.GenericMessage with

ClearVRMessageCodes.ClearVRCoreWrapperVideoDecoderCapabilities. This cotntains some information about the reported video decoder capabilities. Notably, it can be used to query the video decoder level

VideoTrackInfo

Helper class that holds information of one video track, like codec, dimensions and framerate. For detailed information about the ContentInfo, FeedInfo, VideoTrackInfo, AudioTrackInfo, SubtitleTrackInfo objects and their relationship, refer to theContentItem documentation.

VRKeyboard

VRTagalong

Structs

FallbackCubefaceInfoStruct

Advanced API.

Interfaces

ClearVRCoreWrapperAudioStatisticsInterface

(c) Tiledmedia B.V. 2017-2022

This class is used to expose various audio-performance related metrics.

ClearVRCoreWrapperStatisticsInterface

Various ClearVRCoreWrapper performance statistics are exposed via the ClearVRCoreWrapperStatistics object and its subclasses ClearVRCoreWrapperVideoStatisticsInterface and ClearVRCoreWrapperAudioStatisticsInterface Notes:

1. Android specific: each call to any of its members involves several JNI calls, which are considered to be "expensive". It is therefor advised to use any of these fields with moderation and to not query them every Update() cycle.

ClearVRCoreWrapperVideoStatisticsInterface

This class is used to expose various video-performance related metrics.

ClearVRDisplayObjectControllerInterface

DebugInterface

Please do *not* use these APIs in your own project. They are for debugging purposes only and will change without notice

MediaControllerInterface

The controller interface is used to control media playback. APIs like Pause(), Unpause() and Seek() are all part of this interface.

MediaInfoInterface

This interface specifies various convenience methods that allws quick access to the most important media stream properties.

MediaPlayerInterface

The MediaPlayer interface specifies interactions with the MediaPlayer object.

PerformanceInterface

The performance interface exposes various methods to query the performance of the ClearVRCoreWrapper and the underlying ClearVRCore. Note that the ClearVRCore exposes many more metrics, refer to the ClearVRCore documentation and the "perf.*" parameter keys for details.

PlatformOptionsInterface

This is an internal interface and should not be used by the end-user.

SyncInterface

The sync interface can be used to control and view synchronization for live events.

Enums

ABRStartModes

ApplicationFocusAndPauseHandlingTypes

The ClearVRPlayer object has various modes on how it can handle app pause/unpause and suspend/resume events. Under typical conditions, one is strongly recommended to use Recommended mode.

AudioCodecTypes

AudioDecoderTypes

Various AudioDecoders are defined, yet none of them are in use yet. One is encouraged to refrain from using this enum.

AudioFocusChangedHandlingTypes

How the loss and gain of audio focus is being handled by the player.

AudioPlaybackEngineTypes

Various Audio Playback Engines are defined, yet none of them are in use yet. One is encouraged to refrain from using this enum.

CameraAndContentPlacementModes

The ClearVRPlayer can take care of automatically placing camera and/or content based on the content type. Various modes are supported.

ClearVRCoreErrorCodes

ClearVRDisplayObjectEventTypes

Since v9.0 Throughout the lifecycle of aClearVRDisplayObject, it emits various events. One can subscribe to these events by using the AddEventListener method on clearVRPlayer.clearVRDisplayObjectEvents. This enum describes all the events that can be emitted.

ClearVRDRMLicenseServerTypes

Enum listing the various DRM license server types.

ClearVREventTypes

Throughout the life-cycle of a ClearVRPlayer object, several events are generated. Attach a listener using clearVRPlayer.clearVREvents.AddListener(CbClearVREvent); to be informed of these events in your own class. The signature of this event handler is:

void CbClearVREvent(ClearVRPlayer argClearVRPlayer, ClearVREvent argClearVREvent)

ClearVRFishEyeStereoTypes

ClearVRFishEyeTypes

Various Fish Eye Mesh Types are supported.

ClearVRMeshTypes

Devices the type of a mesh. This allows you to infer the actual shape of the mesh

ClearVRMessageCodes

Every ClearVREvent that you receive contains a ClearVRMessage. Each ClearVRMessage holds a code, that can either be:

- 1. a number from the ClearVRMessageCodes enum
- 2. a number from the ClearVRCoreErrorCodes enum This enum describes the ClearVRMessageCodes.

ClearVRMessageTypes

A ClearVRMessage is of certain type. See also ClearVRMessage class for details.

ColorSpaces

Supported color spaces. Currently, only Gamma is supported.

ColorSpaceStandards

ColorSpace is used to indicate the color space used to generate the YUV coordinates. It is used to convert the color back to RGBA. ColorSpace.Unspecified may default to BT.709

ContentFormat

An enum describing the format of the currently playing content.

ContentProtectionRobustnessLevels

Specifies the Content Protection Robustness Level required to play encrypted content.

ContentSupportedStatus

Enum describing whether a ContentItem can be played back or not. Note that this only holds for device configuration as specified when testing using the TestIsContentSupported API

DeviceTypes

A list of the various supported devices andd headsets. These are automatically detected. You can use Utils.GetDeviceType() to know which device type was detected.

DisplayObjectClassTypes

DisplayObjectDescriptorFlags

An internal enum that is not of any use to the integrator.

DRMTypes

EventTypes

FallbackCubefaceIdentifier

FallbackLayoutIdentifier

Advanced enum, do not use.

FishEyeCameraAndLensTypes

Legacy enum that was used for presets for known camera types.

A WARNING

The FishEyeCameraAndLensTypes enum is deprecated. One must use the FisheyePresets enum instead.

FishEyeLensTypes

Lens types for fish eye cameras

FisheyePresets

Fish-eye settings presets for known camera types

InitializeFlags

Flags specific to initializing the ClearVRPlayer.

InteractionModes

This enum is used to help distinguish between different interaction modes depending on the content type.

LogComponents

LogLevels

MeshTextureModes

An internal enum that is not of any use to the integrator.

PrewarmCacheFlags

Flags related to prewarming cache. See ClearVRPlayer.PrewarmCache() for details.

ProjectionTypes

ProxyTypes

(c) Tiledmedia B.V. 2017-2022

Enum that distinguishes between different supported proxy types.

RenderModes

This enum specifies how the video is rendered to the display.

SDKTypes

SeekFlags

Allows one to specify how the new content position in a Seek() or SwitchContent() call (specified in milliseconds) should be interpreted.

SwitchContentFlags

SwitchContent specific flags.

SyncModes

The mode of the synchronization algorithm

SyncState

The state of the synchronization algorithm

TelemetryIPSignallingTypes

TelemetryTargetTypes

TextureBlitModes

This enum specifies the texture blit mode, e.g. how the decoded video frame will be rendered to the screen.

A WARNING

W.r.t. TextureBlitModes.OVROverlayZeoCopy and when playing Widevine L1 protected content only:

Oculus provides two types of plugins: the traditional OVR plugin and the new OVR OpenXR plugin. Both are supported, but some versions of the traditional OVR plugin will spam logcat with the following message:

D/OVRPlugin: ovrpLayerFlag_ProtectedContent requested without a protected front-buffer. This is a bug in this plugin as your content is being played back in a protected context (you can confirm this by taking a screenshot, your video will show up as black or green).

TextureTypes

TextureType is used to indicate how to use the planeIds (what color format is used and how the color planes are arraged)

TimingTypes

Allows one to specify how the new content position in TimingParameters() should be interpreted.

TransitionTypes

When seeking or switching content, one can specify how this should be effectuated.

VideoCodecProfiles

VideoCodecTypes

VideoStereoMode

Devices the type of a mesh. This allows you to infer the actual shape of the mesh

VRAPITypes

VRAPITypes enum, currently, only OculusVR is supported.

A WARNING

This enum is of no use to the integrator and might repurposed, changed and or removed without notice. Do not use.

Delegates

ClearVRPlayer.ClearVRApplicationRegainedFocus

The Delegate that needs to be implemented for the loading of the previous state callback.

ClearVRPlayer.ClearVRApplicationUnpausedDelegate

A delegate of this signature will be triggered when the application is unpaused after it got paused by the OS. Refer to clear VRApplication Unpaused Delegate for details and how to subscribe to a delegate of this signature.

ClearVRPlayer.ClearVRLoadPreviousStateDelegate

The Delegate that needs to be implemented for the loading of the previous state callback. This delegate has been renamed in v9.1 to [ClearVRApplicationRegainedFocus] (xref:com.tiledmedia.clearvr.ClearVRApplicationRegainedFocus(com.tiledmedia.clearvr.ClearVRPlay er.ClearVRApplicationRegainedFocus), which takes ContentInfo as its third argument.

Class ABRLevel

The ABRLevel object contains information about the currently active ABRLevel.

Inheritance

System.Object

ABRLevel

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ABRLevel: object

Fields

url

The currently active URL.

Declaration

public readonly String url

Field Value

TYPE	DESCRIPTION
String	

Methods

ToString()

Declaration

public override String ToString()

Returns

TYPE	DESCRIPTION
String	

Enum ABRStartModes

Namespace	: com.tileamedia.clearv
Assembly:	Assembly-CSharp.dll

Syntax

NAME	DESCRIPTION
Default	
Highest	
Lowest	
Middle	

Enum

ApplicationFocusAndPauseHandlingTypes

The ClearVRPlayer object has various modes on how it can handle app pause/unpause and suspend/resume events. Under typical conditions, one is strongly recommended to use Recommended mode.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ApplicationFocusAndPauseHandlingTypes: int

NAME	DESCRIPTION
Disabled	Handling app focus/pause is left to you as an integrator. Use with care!
	Always put player to pause when app looses focus or is paused: drains battery as the player will remain active and resources will not be freed.
Legacy	▲ WARNING Legacy mode is not available on iOS. It will default to Recommended instead on this platform.
Recommended	Player is killed when app is paused (e.g. pushed to background); player is paused when app looses focus (e.g. in case of a pop-up or when the user puts down the headset). ClearVREventTypes.StateChangedStopped below demonstrates how one could resume playback when the app resumes after it was paused. Note that this behaviour might not suit your requirements, so please check it thoroughly.

Enum AudioCodecTypes

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum AudioCodecTypes : int

NAME	DESCRIPTION
AacLc	
Unspecified	

Class AudioDecoder

The audio decoder is the device that decodes audio samples before playback. Note that this class is not of any use right now, and is merely provided as a boiler-plate concept for the future.

ICH IMPORTANT

One is encouraged NOT to use any of these APIs.

Inheritance

System.Object

AudioDecoder

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class AudioDecoder: object

Constructors

AudioDecoder(AudioDecoderTypes)

Default constructor.

Declaration

public AudioDecoder(AudioDecoderTypes argType)

Parameters

TYPE	NAME	DESCRIPTION
AudioDecoderTypes	argType	The type to construct.

Properties

audioDecoderType

Get the audio decoder type of this object.

Declaration

public AudioDecoderTypes audioDecoderType { get; }

Property Value

TYPE	DESCRIPTION
AudioDecoderTypes	The type

Methods

GetDefaultAudioDecoderForPlatform()

Declaration

public static AudioDecoder GetDefaultAudioDecoderForPlatform()

Returns

TYPE	DESCRIPTION
AudioDecoder	

GetDefaultAudioDecoderForPlatform(RuntimePlatform)

A convenience method that returns a default AudioDecoder object, based on the provided RuntimePlatform.

Declaration

public static AudioDecoder GetDefaultAudioDecoderForPlatform(RuntimePlatform argPlatform)

Parameters

TYPE	NAME	DESCRIPTION
RuntimePlatform	argPlatform	The RuntimePlatform for which to construct the default AudioDecoder.

Returns

TYPE	DESCRIPTION
AudioDecoder	The default AudioDecoder for the specified platform, or an AudioDecoder with type Unknown if platform is not supported.

ToString()

Declaration

public override String ToString()

Returns

TYPE	DESCRIPTION
String	

Enum AudioDecoderTypes

Various AudioDecoders are defined, yet none of them are in use yet. One is encouraged to refrain from using this enum.

Namespace: com.tiledmedia.clearv	1
Assembly: Assembly-CSharp.dll	

Syntax

<pre>public enum AudioDecoderTypes : int</pre>	
--	--

NAME	DESCRIPTION
AndroidDefault	
AndroidMediaCodec	
IOSDefault	
Unknown	
WindowsDefault	

Enum AudioFocusChangedHandlingTypes

How the loss and gain of audio focus is being handled by the player.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum AudioFocusChangedHandlingTypes: int

NAME	DESCRIPTION
Recommended	The video playback will pause on loss of audio focus (ex: receiving a call) and will resume on gaining audio focus again.

Class AudioPlaybackEngine

The audio playback engine is the device that routes decoded audio to an output device. Note that this class is not of any use right now, and is merely provided as a boiler-plate concept for the future.

IMPORTANT

One is encouraged NOT to use any of these APIs.

Inheritance

System.Object

AudioPlaybackEngine

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class AudioPlaybackEngine : object

Constructors

AudioPlaybackEngine(AudioPlaybackEngineTypes)

Default constructor

Declaration

public AudioPlaybackEngine(AudioPlaybackEngineTypes argType)

Parameters

TYPE	NAME	DESCRIPTION
AudioPlaybackEngineTypes	argType	The type of the audio playback engine to create.

Properties

audioPlaybackEngineType

Returns the type of the audio playback engine

Declaration

public AudioPlaybackEngineTypes audioPlaybackEngineType { get; }

Property Value

TYPE	DESCRIPTION
AudioPlaybackEngineTypes	The AudioPlaybackEngineType value of the object.

supportsSpatialAudio

Queries the object whether spatial audio playback is support or note.

Declaration

public bool supportsSpatialAudio { get; }

Property Value

TYPE	DESCRIPTION
System.Boolean	True if spatial audio playback is supported, false otherwise.

Methods

GetDefaultAudioPlaybackEngineForPlatform()

Declaration

public static AudioPlaybackEngine GetDefaultAudioPlaybackEngineForPlatform()

Returns

TYPE	DESCRIPTION
AudioPlaybackEngine	

GetDefaultAudioPlaybackEngineForPlatform(RuntimePlatform)

A convenience method that returns a default AudioPlaybackEngine object, based on the provided RuntimePlatform.

Declaration

public static AudioPlaybackEngine GetDefaultAudioPlaybackEngineForPlatform(RuntimePlatform argPlatform)

Parameters

TYPE	NAME	DESCRIPTION
RuntimePlatform	argPlatform	The RuntimePlatform for which to construct the default AudioPlaybackEngine.

Returns

TYPE	DESCRIPTION
AudioPlaybackEngine	The default AudioPlaybackEngine for the specified platform, or an AudioPlaybackEngine with type Unknown if platform is not supported.

ToString()

Declaration

public override String ToString()

Returns

TYPE	DESCRIPTION
String	

Enum AudioPlaybackEngineTypes

Various Audio Playback Engines are defined, yet none of them are in use yet. One is encouraged to refrain from using this enum.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum AudioPlaybackEngineTypes: int

NAME	DESCRIPTION
AndroidAudioTrack	The AudioTrack-based Audio Playback Engine on Android.
AndroidDefault	The default Audio Playback Engine on Android.
AndroidOpenSL	The OpenSL-based Audio Playback Engine on Android.
IOSDefault	The default Audio Playback Engine on iOS.
Unknown	Unknown/unsupported
WindowsDefault	The default Audio Playback Engine on Windows.

Class AudioTrackAndPlaybackParameters

This class is used to signal the audio track index that one wants to play. Typically used in conjunction with SwitchContentParameters. Notes.

• It is strongly recommended to keep the AudioDecoder and AudioPlaybackEngine arguments at null. This will allow the SDK to select the optional audio decoder and playback engine respectively for the current platform.

Inheritance

System.Object

AudioTrackAndPlaybackParameters

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class AudioTrackAndPlaybackParameters : object

Constructors

AudioTrackAndPlaybackParameters(Int32, AudioDecoder, AudioPlaybackEngine)

Default constructor

Declaration

public AudioTrackAndPlaybackParameters(int argAudioTrackIndex = 0, AudioDecoder argAudioDecoder = null, AudioPlaybackEngine argAudioPlaybackEngine = null)

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	argAudioTrackIndex	The audio track index to play. Default value: 0.
AudioDecoder	argAudioDecoder	The AudioDecoder to use. If set to null, the default AudioDecoder for the current platform will be used.
AudioPlaybackEngine	argAudioPlaybackEngine	The AudioPlaybackEngine to use. If set to null, the default AudioPlaybackEngine for the current platform will be used.

Fields

audioDecoder

Declaration

public AudioDecoder audioDecoder

Field Value

TYPE	DESCRIPTION
AudioDecoder	

audioPlaybackEngine

Declaration

public AudioPlaybackEngine audioPlaybackEngine

Field Value

TYPE	DESCRIPTION
AudioPlaybackEngine	

audioTrackIndex

Declaration

public int audioTrackIndex

Field Value

TYPE	DESCRIPTION
System.Int32	

Properties

estimatedPlaybackLatencyInNanoseconds

The indicative minimum playback buffer duration, as reported by the host device, in milliseconds. This value might be inaccurate, and actual end-to-end latency might differ from this value. When unknown, this value will be 0. Read-only

Declaration

public long estimatedPlaybackLatencyInNanoseconds { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	

Methods

Clone()

Create a deep copy of the object.

Declaration

public AudioTrackAndPlaybackParameters Clone()

Returns

TYPE	DESCRIPTION
AudioTrackAndPlaybackParameters	

GetDefault()

Returns an AudioTrackAndPlaybackParameters with all it's fields set to the current platform's default values.

Declaration

public static AudioTrackAndPlaybackParameters GetDefault()

Returns

TYPE	DESCRIPTION
AudioTrackAndPlaybackParameters	Default AudioTrackAndPlaybackParameters for the current platform (=Application.platform).

GetDefault(RuntimePlatform)

Returns an AudioTrackAndPlaybackParameters with all it's fields set to the current platform's default values.

Declaration

public static AudioTrackAndPlaybackParameters GetDefault(RuntimePlatform argRuntimePlatform)

Parameters

TYPE	NAME	DESCRIPTION
RuntimePlatform	argRuntimePlatform	The runtime platform for which to retrieve the default parameters.

Returns

TYPE	DESCRIPTION
AudioTrackAndPlaybackParameters	

ToString()

Declaration

public override string ToString()

Returns

TYPE	DESCRIPTION
System.String	

Class AudioTrackInfo

Helper class that holds the info of one audio track like codec, sample rate and number of channels. For detailed information about the ContentInfo, FeedInfo, VideoTrackInfo, AudioTrackInfo, SubtitleTrackInfo objects and their relationship, refer to theContentItem documentation.

Inheritance

System.Object AudioTrackInfo

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class AudioTrackInfo: object

Properties

audioCodecType

The audio codec type of this audio track as AudioCodecTypes

Declaration

public AudioCodecTypes audioCodecType { get; }

Property Value

TYPE	DESCRIPTION
AudioCodecTypes	

bitrateInKbps

Get the bitrate of this track in kilobits per second (kbps), as advertised in the manifest



This will return 0 if no bitrate is advertised in the manifest file.

Declaration

public Int32 bitrateInKbps { get; }

Property Value

TYPE	DESCRIPTION
Int32	

boundToVideoTrackIndex

Some audio tracks cannot be individually accessed and played as they are bound to a specific video track. This is for example the case when the audio track and video track are multiplexed together into the same file. If this is the case, this API will return the index of that video track. If the audio track can be individually accessed, this API will return -1. This returns the index of the video track this audio track is bound to, or -1 if this audio track is not bound to any specific video track and can be individually accessed.

Declaration

public int boundToVideoTrackIndex { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

drmType

The type of DRM used in this audio track as DRMTypes.

Declaration

public DRMTypes drmType { get; }

Property Value

TYPE	DESCRIPTION
DRMTypes	

feedIndex

The Feed index associated with this audio track.

Declaration

public int feedIndex { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

isActive

Whether this audio track is active or not.

Declaration

public bool isActive { get; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

language

The language of the audio track as String.

Declaration

public String language { get; }

Property Value

TYPE	DESCRIPTION
String	

name

The name of the audio track as String.

Declaration

public String name { get; }

Property Value

TYPE	DESCRIPTION
String	

numberOfChannels

The number of channels of this audio track as int.

Declaration

public int numberOfChannels { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

sampleRate

The sample rate of this audio track as int.

Declaration

public int sampleRate { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

trackID

Returns the TrackID, a unique identifier for this track and the FeedInfo it belongs to. It can be used when constructing your FeedConfiguration.

Declaration

public TrackID trackID { get; }

Property Value

ТҮРЕ	DESCRIPTION
TrackID	The TrackID, it cannot be null.

trackIndex

The index of the audio track as int.

Declaration

public int trackIndex { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

url

Get the URL

Declaration

public String url { get; }

Property Value

TYPE	DESCRIPTION
String	

Methods

ToString()

Declaration

public override string ToString()

TYPE	DESCRIPTION
System.String	

Class CallCoreResponse

Inheritance

System.Object

CallCoreResponse

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class CallCoreResponse : object

Properties

errorCode

Declaration

public ClearVRMessageCodes errorCode { get; }

Property Value

TYPE	DESCRIPTION
ClearVRMessageCodes	

errorMessage

Declaration

public string errorMessage { get; }

Property Value

TYPE	DESCRIPTION
System.String	

optionalArguments

The optional arguments can be defined when calling the asynchronous PollSyncStatus API.

Declaration

public object[] optionalArguments { get; }

TYPE	DESCRIPTION
System.Object[]	

Methods

ToString()

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	

Enum CameraAndContentPlacementModes

The ClearVRPlayer can take care of automatically placing camera and/or content based on the content type. Various modes are supported.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum CameraAndContentPlacementModes: int

Fields

NAME	DESCRIPTION
Default	The ClearVRPlayer object will automatically move the camera and the display object. For planar or rectilinear content, the display object will be positioned such that if the camera is at position (0,0,0) with default rotation, full content is visible; for omnidirectional content the display object will be moved at the center of the scene (0, 0, 0) will be positionned and its rotation will be reset.
Disabled	The ClearVRPlayer object will NOT automatically move the camera and the display object. The application is responsible for the positioning of the camera and the display object when the FirstFrameRendered ClearVREvent is triggered.
MoveDisplayObjectAndIgnoreCamera	Default behaviors for headset. The camera pose is not touched.
MoveDisplayObjectResetCamera	Default behaviors for flat display. The camera pose is reset when needed.

Class CheckViewportBounds

Component used to verify if the camera is within the expected bounds. If not the camera will move to the closest allowed pose. This component has to be reloaded each time a new content start to be displayed. Calling initialize with the current ClearVRPlayer is enough to reload the component.

Inheritance

System.Object

CheckViewportBounds

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class CheckViewportBounds: MonoBehaviour

Methods

GetMaxZoom()

Returna the max level of zoom that is allowed for planar content.

Declaration

public float GetMaxZoom()

Returns

TYPE	DESCRIPTION
System.Single	The max z value the mesh can be placed at.

GetMinZoom()

Returns the min level of zoom allowed for planar content.

Declaration

public float GetMinZoom()

TYPE	DESCRIPTION
System.Single	The min z value the mesh can be placed at.

Initialize(ClearVRPlayer)

Each time the content has changed (e.g. after a SwitchContent), one MUST call Initialize() again. A suggested place for calling this method is ClearVREventTypes.FirstFrameRendered.

Declaration

public void Initialize(ClearVRPlayer argClearVRPlayer)

Parameters

TYPE	NAME	DESCRIPTION
ClearVRPlayer	argClearVRPlayer	The ClearVRPlayer object that triggered the creation of the ClearVRDisplayObjectControllerBase.

Class ClearVRAsyncRequest

I IMPORTANT

This is an internal class that should not be used. Public access will be removed after 2020/12/31

Inheritance

System.Object

ClearVRAsyncRequest

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRAsyncRequest: object

Methods

ToString()

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	

Class ClearVRConstants

Inheritance

System.Object

ClearVRConstants

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRConstants : object

Enum ClearVRCoreErrorCodes

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ClearVRCoreErrorCodes: int

Fields

NAME	DESCRIPTION
ABRLevelDoesntExist	
AudioFormatGenericError	
AudioFormatUnsupportedAudioFormatError	
AudioTrackDoesntExist	
BinaryOutputError	
CannotPerformABRSwitchDuringNonRunningState	
CannotSwitchToStereoContentRepresentationIsMono	
CannotSwitchToStereoForceMonoEnabled	
CannotSwitchToStereoInsufficientDecoder	
ClearVRCoreConfigurationError	
ClearVRCoreInstanceAlreadyExistsError	
ClearVRCoreInstanceAlreadyInitializedError	
ClearVRCoreStateChangeNotAllowed	
CMSBackendUnknown	
CMSErrorParsingContentList	

NAME	DESCRIPTION
CMSErrorParsingLegacyContentList	
CMSErrorParsingLocalFile	
CMSErrorParsingRawJSON	
CMSErrorReadingLocalFile	
CMSGenericError	
CMSRequestError	
ContentFormatGenericError	
ContentFormatInvalid	
ContentFormatUnsupportedProjectionError	
ContentFormatUnsupportedType	
ContentFormatUnsupportedVersionError	
ContentNotProtected	
ContentSettingsError	
ContentSwitchNotPossible	
DataFormatGenericError	
DataFormatUnsupportedVersionError	
DecoderCapabilitiesNetworkRetrievalError	
DecoderCapabilitiesNotFound	
DecoderOverflowModeActivated	
DecryptionError	

DRMSchemesNotSupportedOnDevice EnableSyncNotAllowed EventManiflestGenericError EventManiflestParsingError EventManiflestUnsupportedVersion GenericError GenericFatalError GenericFatalError GenericFatalError HardwareDecoderLimitationError HitsBlockingManiflestReloadSegmentTimeout HitsBlockingManiflestReloadUnexpectedMSN IdentityDetectionError IllegalFlags IndexFormatUnsupportedVersionError IndexFormatUnsupportedVersionError InteractionNotAvailableDuringLiveEvent InternalIoError InterruptedCommand InvalidLicenseServerType	NAME	DESCRIPTION
EventManifestGenericError EventManifestParsingError EventManifestUnsupportedVersion GenericError GenericFatalError GoHEVCGenericError HardwareDecoderLimitationError HLSBlockingManifestReloadSegmentTimeout HLSBlockingManifestReloadUnexpectedMSN IdentityDetectionError IllegalFlags IndexFormatGenericError InfoStoreGenericError InteractionNotAvailableDuringLiveEvent InterruptedCommand	DRMSchemesNotSupportedOnDevice	
EventManifestParsingError EventManifestUnsupportedVersion GenericError GenericFatalError GoHEVCGenericError HardwareDecoderLimitationError HLSBlockingManifestReloadSegmentTimeout HLSBlockingManifestReloadUnexpectedMSN IdentityDetectionError IllegalFlags IndexFormatGenericError IndexFormatUnsupportedVersionError InfoStoreGenericError InteractionNotAvallableDuringLiveEvent InternallOError InterruptedCommand	EnableSyncNotAllowed	
EventManifestUnsupportedVersion GenericError GenericFatalError GoHEVCGenericError HardwareDecoderLimitationError HLSBlockingManifestReloadSegmentTimeout HLSBlockingManifestReloadUnexpectedMSN IdentityDetectionError IllegalFlags IndexFormatGenericError IndexFormatUnsupportedVersionError InfoStoreGenericError InteractionNotAvailableDuringLiveEvent InternalIOError InternalIOError InterruptedCommand	EventManifestGenericError	
GenericError GenericFatalError GoHEVCGenericError HardwareDecoderLimitationError HLSBlockingManifestReloadSegmentTimeout HLSBlockingManifestReloadUnexpectedMSN IdentityDetectionError IllegalFlags IndexFormatGenericError IndexFormatUnsupportedVersionError InfoStoreGenericError InteractionNotAvailableDuringLiveEvent InternalIOError InterruptedCommand	EventManifestParsingError	
GenericFatalError GoHEVCGenericError HardwareDecoderLimitationError HLSBlockingManifestReloadSegmentTimeout HLSBlockingManifestReloadUnexpectedMSN IdentityDetectionError IllegalFlags IndexFormatGenericError IndexFormatUnsupportedVersionError InteractionNotAvailableDuringLiveEvent InternallOError InterruptedCommand	EventManifestUnsupportedVersion	
GoHEVCGenericError HardwareDecoderLimitationError HLSBlockingManifestReloadSegmentTimeout HLSBlockingManifestReloadUnexpectedMSN IdentityDetectionError IllegalFlags IndexFormatGenericError InfoStoreGenericError InteractionNotAvailableDuringLiveEvent InternalIOError InterruptedCommand	GenericError	
HardwareDecoderLimitationError HLSBlockingManifestReloadSegmentTimeout HLSBlockingManifestReloadUnexpectedMSN IdentityDetectionError IllegalFlags IndexFormatGenericError IndexFormatUnsupportedVersionError InfoStoreGenericError InteractionNotAvailableDuringLiveEvent InternallOError InterruptedCommand	GenericFatalError	
HLSBlockingManifestReloadSegmentTimeout HLSBlockingManifestReloadUnexpectedMSN IdentityDetectionError IllegalFlags IndexFormatGenericError IndexFormatUnsupportedVersionError InfoStoreGenericError InteractionNotAvailableDuringLiveEvent InternallOError InterruptedCommand	GoHEVCGenericError	
HLSBlockingManifestReloadUnexpectedMSN IdentityDetectionError IllegalFlags IndexFormatGenericError IndexFormatUnsupportedVersionError InfoStoreGenericError InteractionNotAvailableDuringLiveEvent InternallOError InterruptedCommand	HardwareDecoderLimitationError	
IdentityDetectionError IllegalFlags IndexFormatGenericError IndexFormatUnsupportedVersionError InfoStoreGenericError InteractionNotAvailableDuringLiveEvent InternallOError InterruptedCommand	HLSBlockingManifestReloadSegmentTimeout	
IllegalFlags IndexFormatGenericError IndexFormatUnsupportedVersionError InfoStoreGenericError InteractionNotAvailableDuringLiveEvent InternallOError InterruptedCommand	HLSBlockingManifestReloadUnexpectedMSN	
IndexFormatGenericError IndexFormatUnsupportedVersionError InfoStoreGenericError InteractionNotAvailableDuringLiveEvent InternallOError InterruptedCommand	IdentityDetectionError	
IndexFormatUnsupportedVersionError InfoStoreGenericError InteractionNotAvailableDuringLiveEvent InternalIOError InterruptedCommand	IllegalFlags	
InfoStoreGenericError InteractionNotAvailableDuringLiveEvent InternallOError InterruptedCommand	IndexFormatGenericError	
InteractionNotAvailableDuringLiveEvent InternallOError InterruptedCommand	IndexFormatUnsupportedVersionError	
InternalIOError InterruptedCommand	InfoStoreGenericError	
InterruptedCommand	InteractionNotAvailableDuringLiveEvent	
	InternalIOError	
InvalidLicenseServerType	InterruptedCommand	
	InvalidLicenseServerType	

NAME	DESCRIPTION
InvalidLogicalFeedIdx	
InvalidManifestURLError	
InvalidPEM	
InvalidPersistencePath	
InvalidPlaybackRate	
InvalidProjectionOverrideSettings	
InvalidProtoPayload	
InvalidStatsReport	
InvalidSyncSettings	
InvalidURLSchemeError	
LicenseFileSignatureInvalidError	
LicenseFormatVersionUnsupportedError	
LicenseGenericError	
LicenseNoneProvided	
LicenseQueueOverFlow	
LicenseSignatureValidationFailedError	
LoadingCoreState	
ManifestGenericError	
ManifestInvalidError	
ManifestNotSupportedError	

NAME	DESCRIPTION
MeasurementOutputError	
NetworkContextCancelled	
NetworkCorruptData	
NetworkDisconnectError	
NetworkError	
NetworkHTTP400	
NetworkHTTP401	
NetworkHTTP402	
NetworkHTTP403	
NetworkHTTP404	
NetworkHTTP405	
NetworkHTTP406	
NetworkHTTP407	
NetworkHTTP408	
NetworkHTTP409	
NetworkHTTP410	
NetworkHTTP411	
NetworkHTTP412	
NetworkHTTP4xx	
NetworkHTTP500	

NetworkHTTP502 NetworkHTTP503 NetworkHTTP504 NetworkHTTP505 NetworkHTTP506 NetworkHTTP507 NetworkHTTP509 NetworkHTTP509 NetworkHTTP510 NetworkHTTP511 NetworkHTTP5xx NetworkHTTP5xx NetworkHTTP5xx NetworkHTTP5xx NetworkHTTP5xx NetworkHTTP5xx NetworkHTTP5xx NotworkHTTP5xx NotworkHTTP5xx NotworkHTTP5xx NotworkHTTP5xx NotworkHTTP5xx NotworkHTTP5xx NotworkHTTP5xx NotworkHTTP5xx NotworkHTTP0mactionError NoClear/VRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound OfflineLicenseExpired	NAME	DESCRIPTION
NetworkHTTP503 NetworkHTTP505 NetworkHTTP506 NetworkHTTP507 NetworkHTTP508 NetworkHTTP509 NetworkHTTP510 NetworkHTTP511 NetworkHTTP5xx NetworkHTTP5xx NetworkHTTP5xx NetworkTCPConnectionError NoClearVRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkHTTP501	
NetworkHTTP504 NetworkHTTP505 NetworkHTTP506 NetworkHTTP507 NetworkHTTP508 NetworkHTTP509 NetworkHTTP510 NetworkHTTP511 NetworkHTTP5xx NetworkHTTP5xx NetworkHTTPUnknownError NoClearVRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkHTTP502	
NetworkHTTP506 NetworkHTTP507 NetworkHTTP508 NetworkHTTP509 NetworkHTTP510 NetworkHTTP511 NetworkHTTP5xx NetworkHTTP5xx NetworkHTTPUnknownError NoClearVRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkHTTP503	
NetworkHTTP506 NetworkHTTP508 NetworkHTTP509 NetworkHTTP510 NetworkHTTP511 NetworkHTTP5xx NetworkHTTPUnknownError NoClearVRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkHTTP504	
NetworkHTTP508 NetworkHTTP509 NetworkHTTP510 NetworkHTTP511 NetworkHTTP5xx NetworkHTTPUnknownError NetworkTCPConnectionError NoClearVRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkHTTP505	
NetworkHTTP509 NetworkHTTP510 NetworkHTTP511 NetworkHTTP5xx NetworkHTTPUnknownError NetworkTCPConnectionError NoClearVRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkHTTP506	
NetworkHTTP510 NetworkHTTP511 NetworkHTTP5xx NetworkHTTPUnknownError NetworkTCPConnectionError NoClearVRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkHTTP507	
NetworkHTTP511 NetworkHTTP5xx NetworkHTTPUnknownError NetworkTCPConnectionError NoClearVRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkHTTP508	
NetworkHTTP511 NetworkHTTP5xx NetworkHTTPUnknownError NetworkTCPConnectionError NoClearVRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkHTTP509	
NetworkHTTPUnknownError NetworkTCPConnectionError NoClearVRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkHTTP510	
NetworkHTTPUnknownError NetworkTCPConnectionError NoClearVRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkHTTP511	
NetworkTCPConnectionError NoClearVRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkHTTP5xx	
NoClearVRCoreActiveError NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkHTTPUnknownError	
NoDRMStructProvided NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NetworkTCPConnectionError	
NonHomogenousTileSize NoOpCommand NoSupportedBundlesFound	NoClearVRCoreActiveError	
NoOpCommand NoSupportedBundlesFound	NoDRMStructProvided	
NoSupportedBundlesFound	NonHomogenousTileSize	
	NoOpCommand	
OfflineLicenseExpired	NoSupportedBundlesFound	
	OfflineLicenseExpired	

NAME	DESCRIPTION
OfflineLicenseInvalid	
ParameterAlreadyInitializedError	
ParameterAlreadyRunningError	
ParameterDoesntSupportContentIDError	
ParameterDoesntSupportIndexingError	
ParameterGenericError	
ParameterInvalidContentIDError	
ParameterInvalidIndexError	
ParameterIsReadOnlyError	
ParameterKeyDoesntExistError	
ParameterNoAudioSelectedError	
ParameterNotInitializedError	
ParameterRequiresIndexingError	
ParameterSetContainsInvalidValue	
ProtobufOutputError	
SavingCoreState	
SavingPersistence	
SeekNotAvailable	
SubtitleTrackDoesntExist	
TestFrameGenericError	

NAME	DESCRIPTION
TileAssignmentGenericError	
TileAssignmentNoFreeSlotAvailableError	
TileCalculationGenericError	
TimingNotAvailableInCurrentState	
TimingUnsupportedValue	
TimingWallclockNotSupportedForNonLive	
UnableToFitRequestedFeedLayoutInDecoder	
UnableToObtainSessionLicense	
UnableToRetrieveLicense	
UnknownDecoderLevelError	
UnknownError	
UnsupportedTestFrameCodec	
UnsupportedTestFrameCodecProfile	
UnsupportedTestFrameDimensions	
UnsupportedTestFrameEncrypted	

Interface

ClearVRCoreWrapperAudioStatisticsInterface

This class is used to expose various audio-performance related metrics.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public interface ClearVRCoreWrapperAudioStatisticsInterface

Properties

framesDropped

Number of frames dropped since creation of the ClearVRPlayer object. In well behaving environment, this value should be close to zero (0). A dropped audio frame is defined as an audio frame that could not be played to the user, e.g. because there was a hick-up in the pipeline or performance was not sufficient.

Declaration

long framesDropped { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	The number of dropped frames.

framesRendered

Number of audio frames rendered since creation of the ClearVRPlayer object. An audio frame is defined as a set of PCM samples of a specific length. For AAC-encoded audio, this audio frame is approximately 21 msec. A rendered frame is defined as a frame that was actually played to the user.

Declaration

long framesRendered { get; }

TYPE	DESCRIPTION
------	-------------

TYPE	DESCRIPTION
System.Int64	The number of rendered frames.

playbackUnderrunCount

The number of buffer underruns during playback. A buffer underrun is defined as a moment in time where no audio data was available to fill the playback buffer. A buffer underrun can result in an audible glitch.

Declaration

int playbackUnderrunCount { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	The number of underruns.

Methods

Destroy()

Destroys the object, freeing any resources it might have claimed.

Declaration

void Destroy()

Interface

ClearVRCoreWrapperStatisticsInterface

Various ClearVRCoreWrapper performance statistics are exposed via the ClearVRCoreWrapperStatistics object and its subclasses ClearVRCoreWrapperVideoStatisticsInterface and ClearVRCoreWrapperAudioStatisticsInterface Notes:

1. Android specific: each call to any of its members involves several JNI calls, which are considered to be "expensive". It is therefor advised to use any of these fields with moderation and to not query them every Update() cycle.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public interface ClearVRCoreWrapperStatisticsInterface

Methods

Destroy()

the ClearVRCoreWrapperStatistics *must* be properly destroyed to release any claimed resources.

Declaration

void Destroy()

Interface

ClearVRCoreWrapperVideoStatisticsInterface

This class is used to expose various video-performance related metrics.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public interface ClearVRCoreWrapperVideoStatisticsInterface

Properties

averageDecoderInputQueueSize

An advanced metric measuring the average decoder input queue size. This value should be less than 0.3 under normal conditions. A value larger than 1 indicates that the device might not be able to keep up with the video, resulting in very poor user experience. This value is averaged over approximately 6 seconds.

Declaration

float averageDecoderInputQueueSize { get; }

Property Value

TYPE	DESCRIPTION
System.Single	The average decodere input queue size.

averageDecoderOutputQueueSize

An advanced metric measuring the average decoder output queue size (e.g. how many decoded frames are queued up and ready for release). This value should be less than 3 under normal conditions. A value larger than 5 indicates that the device might not be able to keep up with the video, resulting in very poor user experience. This value is averaged over approximately 6 seconds.

Declaration

float averageDecoderOutputQueueSize { get; }

TYPE	DESCRIPTION
System.Single	The average decodere output queue size.

endToEndFrameLatencyMean

The average end-to-end inter-frame latency is defined as the time between the moment an encoded (HEVC) frame was created and it actually being displayed to the end-user. This metric should be less than 90 msec and can be as low as 30 msec. Notes.

- 1. when in Paused state, this value is expected to increase.
- 2. this metric has a bit of overshoot. Actual end-to-end latency is slightly lower than reported.

Declaration

float endToEndFrameLatencyMean { get; }

Property Value

TYPE	DESCRIPTION
System.Single	The mean end-to-end inter-frame latency in milliseconds or 0 if unknown.

endToEndFrameLatencyStandardDeviation

The average end-to-end inter-frame latency is defined as the time between the moment an encoded (HEVC) frame was created and it actually being displayed to the end-user. This metric should be less than 90 msec and can be as low as 30 msec. Notes.

- 1. when in Paused state, this value is expected to increase.
- 2. this metric has a bit of overshoot. Actual end-to-end latency is slightly lower than reported.

Declaration

float endToEndFrameLatencyStandardDeviation { get; }

TYPE	DESCRIPTION
System.Single	The average end-to-end inter-frame latency standard deviation in milliseconds, or 0 if unknown.

frameReleaseQuality

An advanced metric giving an indication of the quality of the frame release rhythm. In ideal situations, this value should be (very close to) 100%. The lower this value, the more likely the end-user will notice stuttering/jittering. This value is averaged over approximately 6 seconds.

Declaration

float frameReleaseQuality { get; }

Property Value

TYPE	DESCRIPTION
System.Single	The frame release quality as a percentage.

framesDropped

Number of frames dropped since creation of the ClearVRPlayer object. In well behaving environment, this value should be close to zero (0). A dropped frame is defined as a frame that could not be shown to the user, e.g. because the application framerate was too low, or because there was a hick-up in the decoder pipeline.

Declaration

long framesDropped { get; }

Property Value

ТҮРЕ	DESCRIPTION
System.Int64	The number of dropped frames.

framesRendered

Number of frames rendered since creation of the ClearVRPlayer object. A rendered frame is defined as a frame that was actually shown to the user.

Declaration

long framesRendered { get; }

TYPE	DESCRIPTION
System.Int64	The number of rendered frames.

interFrameApplicationLatencyMean

The average application inter-frame latency is defined as the time between two consequtive application frame updates. In Unity, this equals to the time between two Update() cycles. This metric should be (very) close to the vsync.

Declaration

float interFrameApplicationLatencyMean { get; }

Property Value

TYPE	DESCRIPTION
System.Single	The mean application inter-frame latency in milliseconds or 0 if unknown.

interFrameApplicationLatencyStandardDeviation

The average application inter-frame latency is defined as the time between two consequtive application frame updates. In Unity, this equals to the time between two Update() cycles. This metric should be (very) close to the vsync. In well behaving application, standard deviation should be very low (e.g. less than 3 msec). Higher values might indicate a performance bottleneck.

Declaration

float interFrameApplicationLatencyStandardDeviation { get; }

Property Value

TYPE	DESCRIPTION
System.Single	The average application inter-frame latency standard deviation in milliseconds, or 0 if unknown.

interFrameDecoderLatencyMean

The average decoder inter-frame latency is defined as the time between two consecutively decoded frames. Notes.

1. when in Paused state, this value is expected to increase.

2. if you are interested in the inter-video-frame render latency (e.g. the average "render framerate"), please check interFrameRenderLatencyMean.

Declaration

float interFrameDecoderLatencyMean { get; }

Property Value

TYPE	DESCRIPTION
System.Single	The mean inter-frame decoder latency in milliseconds or 0 if unknown.

interFrameDecoderLatencyStandardDeviation

The average decoder inter-frame latency standard deviation is defined as the time between two consecutively decoded frames. Notes.

1. when in Paused state, this value is expected to increase.

Declaration

float interFrameDecoderLatencyStandardDeviation { get; }

Property Value

TYPE	DESCRIPTION
System.Single	The average inter-frame decoder latency standard deviation in milliseconds, or 0 if unknown.

interFrameRenderLatencyMean

The average render inter-frame latency is defined as the time between two consecutively rendered frames. Notes.

1. when in Paused state, this value is expected to increase.

Declaration

float interFrameRenderLatencyMean { get; }

TYPE	DESCRIPTION
System.Single	The mean inter-frame render latency in milliseconds or 0 if unknown.

interFrameRenderLatencyStandardDeviation

The average render inter-frame latency standard deviation is defined as the time between two consecutively rendered frames. Notes.

1. when in Paused state, this value is expected to increase.

Declaration

float interFrameRenderLatencyStandardDeviation { get; }

Property Value

TYPE	DESCRIPTION
System.Single	The average inter-frame render latency standard deviation in milliseconds, or 0 if unknown.

vsyncQuality

An advanced metric giving an indication of the quality of the vsync rhythm. In ideal situations, this value should be (very close to) 100%. This value is averaged over approximately 6 seconds.

Declaration

float vsyncQuality { get; }

Property Value

TYPE	DESCRIPTION
System.Single	The vsync quality as a percentage.

Methods

Destroy()

Destroys the object, freeing any resources it might have claimed.

Declaration

void Destroy()

GetEndToEndFrameLatencyAsPrettyString()

Returns the inter-frame end-to-end latency as a formatted string. Example: mean +/- stddev. Both values have two decimal values to it. Note that this metric has a slight overshoot, and in reality is lower than reported.

Declaration

String GetEndToEndFrameLatencyAsPrettyString()

Returns

TYPE	DESCRIPTION
String	A formatted string.

GetInterFrameApplicationLatencyAsPrettyString()

Returns the inter-frame application latency as a formatted string. Example: mean +/- stddev. Both values have two decimal values to it.

Declaration

String GetInterFrameApplicationLatencyAsPrettyString()

Returns

TYPE	DESCRIPTION
String	A formatted string.

GetInterFrameApplicationRateInFramesPerSecond()

Returns the inter frame application latency as a two-decimal pretty print framerate string. Example: 59.94.

Declaration

float GetInterFrameApplicationRateInFramesPerSecond()

ТҮРЕ	DESCRIPTION
System.Single	The pretty-printed framerate.

GetInterFrameDecoderLatencyAsPrettyString()

Returns the inter-frame decoder latency as a formatted string. Example: mean +/- stddev. Both values have two decimal values to it.

Declaration

String GetInterFrameDecoderLatencyAsPrettyString()

Returns

TYPE	DESCRIPTION
String	A formatted string.

GetInterFrameDecoderRateInFramesPerSecond()

Returns the inter frame decoder latency as a two-decimal pretty print framerate string. Example: 59.94.

Declaration

float GetInterFrameDecoderRateInFramesPerSecond()

Returns

TYPE	DESCRIPTION
System.Single	The pretty-printed framerate.

GetInterFrameRenderLatencyAsPrettyString()

Returns the inter-frame render latency as a formatted string. Example: mean +/- stddev. Both values have two decimal values to it.

Declaration

String GetInterFrameRenderLatencyAsPrettyString()

Returns

TYPE	DESCRIPTION
String	A formatted string.

GetInterFrameRenderRateInFramesPerSecond()

Returns the inter frame render latency as a two-decimal pretty print framerate string. Example: 59.94.

Declaration

float GetInterFrameRenderRateInFramesPerSecond()

TYPE	DESCRIPTION
System.Single	The pretty-printed framerate.

Class ClearVRDisplayObject

This class has been removed in v9.0 and can no longer be used.

Inheritance

System.Object

ClearVRDisplayObject

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRDisplayObject : object

Fields

pose

The Pose

Declaration

public ClearVRPose pose

Field Value

TYPE	DESCRIPTION
ClearVRPose	

scale

The scale

Declaration

public ClearVRScale scale

Field Value

TYPE	DESCRIPTION
ClearVRScale	

Class ClearVRDisplayObjectControllerBase

Inheritance

System.Object

ClearVRDisplayObjectControllerBase

ClearVRDisplayObjectControllerMesh

ClearVRDisplayObjectControllerSprite

ClearVRDisplayObjectControllerUnmanagedMesh

Implements

ClearVRDisplayObjectControllerInterface

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public abstract class ClearVRDisplayObjectControllerBase : MonoBehaviour, ClearVRDisplayObjectControllerInterface, ClearVRDisplayObjectControllerInterfaceInternal

Fields

_cameraPositionId

Declaration

protected static int _cameraPositionId

Field Value

ТҮРЕ	DESCRIPTION
System.Int32	

_contentFormat

Declaration

protected ContentFormat _contentFormat

Field Value

TYPE	DESCRIPTION
ContentFormat	

_isOESFastPathOnKeywordEnabled

protected bool _isOESFastPathOnKeywordEnabled

Field Value

TYPE	DESCRIPTION
System.Boolean	

_isPicoVREyeIndexKeywordEnabled

Declaration

protected bool _isPicoVREyeIndexKeywordEnabled

Field Value

TYPE	DESCRIPTION
System.Boolean	

_isReadyForDestruction

Declaration

protected bool _isReadyForDestruction

Field Value

TYPE	DESCRIPTION
System.Boolean	

_isRegisteredAndInitialized

Declaration

protected bool _isRegisteredAndInitialized

Field Value

TYPE	DESCRIPTION
System.Boolean	

$_is Stereo Custom UVs On Keyword Enabled$

protected bool _isStereoCustomUVsOnKeywordEnabled

Field Value

TYPE	DESCRIPTION
System.Boolean	

_isTransparencyOnKeywordEnabled

Declaration

protected bool _isTransparencyOnKeywordEnabled

Field Value

TYPE	DESCRIPTION
System.Boolean	

_isZWriteEnabled

Declaration

protected bool _isZWriteEnabled

Field Value

TYPE	DESCRIPTION
System.Boolean	

_lastTextureId

Declaration

protected IntPtr _lastTextureId

Field Value

TYPE	DESCRIPTION
IntPtr	

_material

protected Material _material

Field Value

TYPE	DESCRIPTION
Material	

_originalActiveState

Declaration

protected bool _originalActiveState

Field Value

TYPE	DESCRIPTION
System.Boolean	

_originalMaterial

Declaration

protected Material _originalMaterial

Field Value

TYPE	DESCRIPTION
Material	

_originalShader

Declaration

protected Shader _originalShader

Field Value

TYPE	DESCRIPTION
Shader	

$_platformOptions$

protected PlatformOptionsBase _platformOptions

Field Value

TYPE	DESCRIPTION
PlatformOptionsBase	

_renderMode

Declaration

protected RenderModes _renderMode

Field Value

TYPE	DESCRIPTION
RenderModes	

_shaderColor

Declaration

protected Color _shaderColor

Field Value

ТҮРЕ	DESCRIPTION
Color	

_shaderColorSpaceYUVtoRGBTransformMatrix

Declaration

protected Matrix4x4 _shaderColorSpaceYUVtoRGBTransformMatrix

Field Value

TYPE	DESCRIPTION
Matrix4x4	

_viewMatrixId

protected static int _viewMatrixId

Field Value

TYPE	DESCRIPTION
System.Int32	

clearVRShader

Declaration

protected static Shader clearVRShader

Field Value

TYPE	DESCRIPTION
Shader	

forceDisableMeshRendererAtAllTimes

Declaration

protected bool forceDisableMeshRendererAtAllTimes

Field Value

TYPE	DESCRIPTION
System.Boolean	

genericNonClearVROmniShader

Declaration

protected static Shader genericNonClearVROmniShader

Field Value

ТҮРЕ	DESCRIPTION
Shader	

hideInAwake

You can use a placeholder mesh to design your scene more efficiently. By setting this boolean to true, this placeholder mesh is immediately hidden upon awake. If set to false, the mesh placeholder

will not be hidden and simply replaced by video once that video has been loaded. The latter mode can be useful when the placeholder mesh contains e.g. a "loading..." texture.

Declaration

public bool hideInAwake

Field Value

TYPE	DESCRIPTION
System.Boolean	

Properties

activeFeedIndex

Get the feed index that is currently present on the associated gameObject. Note that a DisplayObjectController can be *mapped* to a feed index (through the LayoutManager), but this mapping might not be active yet (because the feed is still loading for example). In this case, this API returns -1.

Declaration

public int activeFeedIndex { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	The currently active feed index, or -1 if no Feed is active yet.

clearVRLayoutManager

Declaration

protected ClearVRLayoutManager clearVRLayoutManager { get; }

Property Value

ТҮРЕ	DESCRIPTION
ClearVRLayoutManager	

clearVRMeshType

public ClearVRMeshTypes clearVRMeshType { get; }

Property Value

TYPE	DESCRIPTION
ClearVRMeshTypes	

contentFormat

Declaration

public ContentFormat contentFormat { get; }

Property Value

TYPE	DESCRIPTION
ContentFormat	

displayObjectClassType

Returns the current DisplayObjectClassType of the DisplayObject. This value is always up-to-date.

6 NOTE

The DisplayObjectClassType can change during the lifecycle of a DisplayObject based on the Layout you request. The ClearVRDisplayObjectEventTypes.ClassTypeChanged event is emitted when it has changed.

Declaration

public DisplayObjectClassTypes displayObjectClassType { get; }

Property Value

ТҮРЕ	DESCRIPTION
DisplayObjectClassTypes	The DisplayObjectClassType of the DisplayObject, or DisplayObjectClassTypes.Unknown if not available (when the DisplayObject is not active).

frameHeight

Declaration

public int frameHeight { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

frameWidth

Declaration

public int frameWidth { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

isActive

Returns the active state of this mesh. This is NOT the same as gameObject.activeSelf.

Declaration

public bool isActive { get; }

Property Value

TYPE	DESCRIPTION
System.Boolean	True if the DisplayObject is activated, false otherwise.

meshTextureMode

Declaration

public abstract MeshTextureModes meshTextureMode { get; set; }

Property Value

TYPE	DESCRIPTION
MeshTextureModes	

renderMode

Get and set the RenderMode of this DisplayObject.

(c) Tiledmedia B.V. 2017-2022

Declaration

public RenderModes renderMode { get; set; }

Property Value

TYPE	DESCRIPTION
RenderModes	

Methods

BindNativeChromaTexture2ToShader(Texture)

Declaration

public virtual void BindNativeChromaTexture2ToShader(Texture argNativeChromaTexture)

Parameters

TYPE	NAME	DESCRIPTION
Texture	argNativeChromaTexture	

BindNativeChromaTextureToShader(Texture)

Declaration

public virtual void BindNativeChromaTextureToShader(Texture argNativeChromaTexture)

Parameters

TYPE	NAME	DESCRIPTION
Texture	argNativeChromaTexture	

BindNativeTextureToShader(Texture)

Declaration

public virtual void BindNativeTextureToShader(Texture argNativeTexture)

Parameters

TYPE	NAME	DESCRIPTION
Texture	argNativeTexture	

ControllerAwake()

Declaration

protected void ControllerAwake()

DestroyNativeTextures()

Declaration

public virtual void DestroyNativeTextures()

DoBaseLateUpdate()

Declaration

public void DoBaseLateUpdate()

EnableOrDisableMeshRenderer(Boolean)

Declaration

public abstract void EnableOrDisableMeshRenderer(bool argIsEnabled)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	argIsEnabled	

EnableOrDisableOESFastPath(Boolean, Boolean)

Declaration

public void EnableOrDisableOESFastPath(bool argValue, bool argForceUpdate = false)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	argValue	
System.Boolean	argForceUpdate	

EnableOrDisableStereoscopicRendering(Boolean, Boolean)

Declaration

public virtual void EnableOrDisableStereoscopicRendering(bool argValue, bool argForceUpdate = false)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	argValue	
System.Boolean	argForceUpdate	

GetContentFormat()

Declaration

public ContentFormat GetContentFormat()

Returns

TYPE	DESCRIPTION
ContentFormat	

GetDisplayObjectClassType()

Returns the current DisplayObjectClassType of the DisplayObject. This value is always up-to-date.

6 NOTE

The DisplayObjectClassType can change during the lifecycle of a DisplayObject based on the Layout you request. The ClearVRDisplayObjectEventTypes.ClassTypeChanged event is emitted when it has changed.

Declaration

public DisplayObjectClassTypes GetDisplayObjectClassType()

Returns

TYPE	DESCRIPTION
DisplayObjectClassTypes	The DisplayObjectClassType of the DisplayObject, or DisplayObjectClassTypes.Unknown if not available (when the DisplayObject is not active).

GetFallbackLayout()

Declaration

public System.Object GetFallbackLayout()

Returns

TYPE	DESCRIPTION
System.Object	

GetIsActive()

Returns the active state of this mesh. This is NOT the same as gameObject.activeSelf.

Declaration

public bool GetIsActive()

Returns

TYPE	DESCRIPTION
System.Boolean	True if the DisplayObject is activated, false otherwise.

GetTexture()

Declaration

public virtual Texture GetTexture()

Returns

TYPE	DESCRIPTION
Texture	

GetTransformationMatrixFromMeshDescriptionStruct()

Declaration

protected virtual Matrix4x4 GetTransformationMatrixFromMeshDescriptionStruct()

Returns

TYPE	DESCRIPTION
Matrix4x4	

HasLocalShaderParameterChange()

Declaration

protected virtual bool HasLocalShaderParameterChange()

Returns

TYPE	DESCRIPTION
System.Boolean	

Initialize(PlatformOptionsBase, SharedPointersWithSDK, System.Object)

Initialize the ClearVRDisplayObjectController for video playback.

Declaration

public virtual void Initialize(PlatformOptionsBase argPlatformOptions, SharedPointersWithSDK sharedPointersWithSDK, System.Object argReserved)

Parameters

TYPE	NAME	DESCRIPTION
PlatformOptionsBase	argPlatformOptions	
SharedPointersWithSDK	sharedPointersWithSDK	
System.Object	argReserved	

InitializeRendererAndMaterial()

Declaration

protected abstract void InitializeRendererAndMaterial()

IsMeshRendererEnabled()

Declaration

public abstract bool IsMeshRendererEnabled()

Returns

TYPE	DESCRIPTION
System.Boolean	

LoadShaders()

Declaration

public static void LoadShaders()

OnDestroy()

Declaration

protected virtual void OnDestroy()

ReconfigureShader(Shader)

Declaration

protected void ReconfigureShader(Shader argShader)

Parameters

TYPE	NAME	DESCRIPTION
Shader	argShader	

RecreateNativeTextures()

Declaration

public virtual void RecreateNativeTextures()

SetMainColor(Color)

Declaration

public bool SetMainColor(Color argNewColor)

Parameters

TYPE	NAME	DESCRIPTION
Color	argNewColor	

Returns

TYPE	DESCRIPTION
System.Boolean	

SetRenderMode(RenderModes)

Declaration

public void SetRenderMode(RenderModes argNewRenderMode)

Parameters

TYPE	NAME	DESCRIPTION
RenderModes	argNewRenderMode	

SetZWrite(Boolean, Boolean)

Declaration

public void SetZWrite(bool argValue, bool argForceUpdate = false)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	argValue	
System.Boolean	argForceUpdate	

ToString()

Declaration

public override String ToString()

Returns

ТҮРЕ	DESCRIPTION
String	

UnbindAndReleaseNativeTextures()

Declaration

public void UnbindAndReleaseNativeTextures()

UpdateApplicationMeshState()

Declaration

public virtual void UpdateApplicationMeshState()

UpdateNativeTextures(IntPtr, IntPtr, IntPtr)

Declaration

public virtual void UpdateNativeTextures(IntPtr argTextureId0, IntPtr argTextureId1, IntPtr argTextureId2)

Parameters

TYPE	NAME	DESCRIPTION
IntPtr	argTextureId0	
IntPtr	argTextureId1	
IntPtr	argTextureId2	

UpdateShader()

Declaration

public virtual void UpdateShader()

Implements

 ${\bf Clear VRD is play Object Controller Interface}$

Interface ClearVRDisplayObjectControllerInterface

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public interface ClearVRDisplayObjectControllerInterface

Methods

EnableOrDisableStereoscopicRendering(Boolean, Boolean)

Since v9.0 Returns the content format. Can be used to figure out whether the content is 360 or 180 degrees and if it is monoscopic or stereoscopic. Notes.

1. if, for whatever reason, the content format changes, the callback event ContentFormatChanged will be invoked on the clearVRPlayer.clearVRDisplayObjectEvents event queue.

Declaration

void EnableOrDisableStereoscopicRendering(bool argValue, bool argForceUpdate = false)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	argValue	
System.Boolean	argForceUpdate	

GetContentFormat()

Returns the content format. Can be used to figure out whether the content is 360 or 180 degrees and if it is monoscopic or stereoscopic. Notes.

1. if, for whatever reason, the content format changes, the callback event ContentFormatChanged will be invoked per DisplayObject.

Declaration

ContentFormat GetContentFormat()

TYPE	DESCRIPTION
ContentFormat	The current content format.

GetFallbackLayout()

Since v4.1.2 Returns the layout of the fallback tiles.

Notes.

- 1. You should only call this method after you have received the FirstFrameRendered event.
- 2. Remember that the layout of the fallback can change when switching between clips. If it changed, you will receive a new FirstFrameRendered event and you should requery the FallbackLayout.
- 3. Currently, only the layout of the fallback tiles of the eft eye are signalled.
- 4. The layout of the left eye's fallback tiles is fixed, it will not change during runtime

Declaration

System.Object GetFallbackLayout()

Returns

TYPE	DESCRIPTION
System.Object	The layout of the fallback tiles.

GetIsActive()

Since v9.0 Query whether the DisplayObject is active or not.

O NOTE

Note that active is NOT the same as gameobject.activeSelf. You can use this API in conjunction with the ClearVRDisplayObjectEventTypes.ActiveStateChanged as it gets triggered once the state has changed.

Declaration

bool GetIsActive()

TYPE	DESCRIPTION
System.Boolean	

GetTexture()

Since v4.1.2 Returns the Texture2D object of the currently active texture. Note that the tiles in this texture are shuffled.

Notes.

- 1. You should only call this method after you have received the FirstFrameRendered event.
- 2. This texture might change as a result of switchContent() and/or ABR events.
- 3. This texture will NOT change when switching from monoscopic to stereoscopic rendering or vice versa.

Declaration

Texture GetTexture()

Returns

TYPE	DESCRIPTION
Texture	The currently active video texture as a properly bounded Texture2D.

SetMainColor(Color)

Change the main color of the video texture. This can also be used to make the sphere transparent by setting the alpha component accordingly. For common performance reasons not specifically related to ClearVR streaming, you should carefully assess the potential negative performance impact of using transparency on mobile devices.

Declaration

bool SetMainColor(Color argNewColor)

Parameters

TYPE	NAME	DESCRIPTION
Color	argNewColor	The new color.

TYPE	DESCRIPTION
System.Boolean	True if successful, false otherwise.

SetRenderMode(RenderModes)

This API is used to switch between monoscopic and stereoscopic rendering (if the content and player allows for it). Note that, in case of ClearVR contrent, this only changes how the content is *rendered*, NOT how it is retrieved from the network. If you want to switch between monoscopic or stereoscopic content **retrieval**, you should use the SetStereoMode API instead. Note that the SetStereoMode API is considered an advanced API and should not be used under typical conditions. The SetRenderMode API is especially suited for temporarily disabling stereoscopic rendering when you show an in-player menu (to prevent depth-fighting from happening) as this API is instantaneous while the SetStereoMode() API will take a bit of time to respond, reducing user experience.

You will receive a ClearVREventTypes.RenderModeChanged event on the clearVRPlayer.clearVRDisplayObjectEvents event queue when the render mode has changed. If an incorrect request is made (e.g. when you try to enable stereoscopic content playback on a monoscopic video), the request will be bounced and an appropriate error message can be found in the event's ClearVRMessage.

Declaration

void SetRenderMode(RenderModes argNewRenderMode)

Parameters

TYPE	NAME	DESCRIPTION
RenderModes	argNewRenderMode	The new render mode.

Class ClearVRDisplayObjectControllerMesh

Inheritance

System.Object

ClearVRDisplayObjectControllerBase

ClearVRDisplayObjectControllerMesh

ClearVRDisplayObjectControllerOVROverlay

Implements

ClearVRDisplayObjectControllerInterface

Inherited Members

 ${\bf Clear VRD is play Object Controller Base. hide In Awake}$

ClearVRDisplayObjectControllerBase.activeFeedIndex

ClearVRDisplayObjectControllerBase. contentFormat

ClearVRDisplayObjectControllerBase._renderMode

ClearVRDisplayObjectControllerBase.renderMode

ClearVRDisplayObjectControllerBase.contentFormat

ClearVRDisplayObjectControllerBase.displayObjectClassType

ClearVRDisplayObjectControllerBase.GetDisplayObjectClassType()

ClearVRDisplayObjectControllerBase.isActive

ClearVRDisplayObjectControllerBase.GetIsActive()

ClearVRDisplayObjectControllerBase.clearVRLayoutManager

ClearVRDisplayObjectControllerBase. isReadyForDestruction

ClearVRDisplayObjectControllerBase. material

ClearVRDisplayObjectControllerBase. isStereoCustomUVsOnKeywordEnabled

ClearVRDisplayObjectControllerBase. isOESFastPathOnKeywordEnabled

ClearVRDisplayObjectControllerBase. isTransparencyOnKeywordEnabled

ClearVRDisplayObjectControllerBase. isPicoVREyeIndexKeywordEnabled

ClearVRDisplayObjectControllerBase. isZWriteEnabled

ClearVRDisplayObjectControllerBase. shaderColor

ClearVRDisplayObjectControllerBase. shaderColorSpaceYUVtoRGBTransformMatrix

ClearVRDisplayObjectControllerBase. lastTextureId

ClearVRDisplayObjectControllerBase. platformOptions

ClearVRDisplayObjectControllerBase. cameraPositionId

ClearVRDisplayObjectControllerBase. viewMatrixId

ClearVRDisplayObjectControllerBase.clearVRMeshType

ClearVRDisplayObjectControllerBase.forceDisableMeshRendererAtAllTimes

ClearVRDisplayObjectControllerBase._isRegisteredAndInitialized

ClearVRDisplayObjectControllerBase.clearVRShader

ClearVRDisplayObjectControllerBase.genericNonClearVROmniShader

ClearVRDisplayObjectControllerBase._originalMaterial

ClearVRDisplayObjectControllerBase. originalShader

ClearVRDisplayObjectControllerBase. originalActiveState

ClearVRDisplayObjectControllerBase.frameWidth

ClearVRDisplayObjectControllerBase.frameHeight

ClearVRDisplayObjectControllerBase.UpdateApplicationMeshState()

ClearVRDisplayObjectControllerBase.LoadShaders()

ClearVRDisplayObjectControllerBase.ControllerAwake()

ClearVRDisplayObjectControllerBase.Initialize(PlatformOptionsBase, SharedPointersWithSDK,

System.Object)

ClearVRDisplayObjectControllerBase.DoBaseLateUpdate()

ClearVRDisplayObjectControllerBase.UpdateShader()

ClearVRDisplayObjectControllerBase.ReconfigureShader(Shader)

ClearVRDisplayObjectControllerBase.EnableOrDisableStereoscopicRendering(Boolean, Boolean)

ClearVRDisplayObjectControllerBase.EnableOrDisableOESFastPath(Boolean, Boolean)

ClearVRDisplayObjectControllerBase.SetZWrite(Boolean, Boolean)

ClearVRDisplayObjectControllerBase.GetContentFormat()

ClearVRDisplayObjectControllerBase.SetRenderMode(RenderModes)

ClearVRDisplayObjectControllerBase.SetMainColor(Color)

ClearVRDisplayObjectControllerBase.HasLocalShaderParameterChange()

ClearVRDisplayObjectControllerBase.GetTransformationMatrixFromMeshDescriptionStruct()

ClearVRDisplayObjectControllerBase.RecreateNativeTextures()

ClearVRDisplayObjectControllerBase.UpdateNativeTextures(IntPtr, IntPtr, IntPtr)

ClearVRDisplayObjectControllerBase.DestroyNativeTextures()

ClearVRDisplayObjectControllerBase.BindNativeTextureToShader(Texture)

ClearVRDisplayObjectControllerBase.BindNativeChromaTextureToShader(Texture)

ClearVRDisplayObjectControllerBase.BindNativeChromaTexture2ToShader(Texture)

ClearVRDisplayObjectControllerBase.GetTexture()

ClearVRDisplayObjectControllerBase.UnbindAndReleaseNativeTextures()

ClearVRDisplayObjectControllerBase.GetFallbackLayout()

ClearVRDisplayObjectControllerBase.ToString()

ClearVRDisplayObjectControllerBase.OnDestroy()

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRDisplayObjectControllerMesh : ClearVRDisplayObjectControllerBase, ClearVRDisplayObjectControllerInterface, ClearVRDisplayObjectControllerInterfaceInternal

Properties

meshTextureMode

Declaration

public override MeshTextureModes meshTextureMode { get; set; }

Property Value

TYPE	DESCRIPTION
MeshTextureModes	

Overrides

ClearVRDisplayObjectControllerBase.meshTextureMode

Methods

EnableOrDisableMeshRenderer(Boolean)

Declaration

public override void EnableOrDisableMeshRenderer(bool argIsEnabled)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	arglsEnabled	

Overrides

ClearVRDisplayObjectControllerBase.EnableOrDisableMeshRenderer(Boolean)

InitializeRendererAndMaterial()

Declaration

protected override void InitializeRendererAndMaterial()

Overrides

ClearVRDisplayObjectControllerBase.InitializeRendererAndMaterial()

IsMeshRendererEnabled()

Declaration

public override bool IsMeshRendererEnabled()

TYPE	DESCRIPTION
System.Boolean	

Overrides

 ${\bf ClearVRD} is play Object Controller Base. Is Mesh Renderer Enabled ()$

Implements

 ${\bf Clear VRD is play Object Controller Interface}$

Class

ClearVRDisplayObjectControllerOVROverlay

Holds the ClearVRDisplayObjectController gameobject which is rendered on an OVROverlay

Inheritance

System.Object

ClearVRDisplayObjectControllerBase

ClearVRDisplayObjectControllerMesh

ClearVRDisplayObjectControllerOVROverlay

Implements

ClearVRDisplayObjectControllerInterface

Inherited Members

ClearVRDisplayObjectControllerMesh.InitializeRendererAndMaterial()

ClearVRDisplayObjectControllerMesh.EnableOrDisableMeshRenderer(Boolean)

ClearVRDisplayObjectControllerMesh.IsMeshRendererEnabled()

ClearVRDisplayObjectControllerBase.hideInAwake

ClearVRDisplayObjectControllerBase.activeFeedIndex

ClearVRDisplayObjectControllerBase._contentFormat

ClearVRDisplayObjectControllerBase._renderMode

ClearVRDisplayObjectControllerBase.renderMode

ClearVRDisplayObjectControllerBase.contentFormat

ClearVRDisplayObjectControllerBase.displayObjectClassType

ClearVRDisplayObjectControllerBase.GetDisplayObjectClassType()

ClearVRDisplayObjectControllerBase.isActive

ClearVRDisplayObjectControllerBase.GetIsActive()

ClearVRDisplayObjectControllerBase.clearVRLayoutManager

ClearVRDisplayObjectControllerBase. isReadyForDestruction

ClearVRDisplayObjectControllerBase. material

ClearVRDisplayObjectControllerBase. isStereoCustomUVsOnKeywordEnabled

ClearVRDisplayObjectControllerBase. isOESFastPathOnKeywordEnabled

ClearVRDisplayObjectControllerBase. isTransparencyOnKeywordEnabled

ClearVRDisplayObjectControllerBase. isPicoVREyeIndexKeywordEnabled

ClearVRDisplayObjectControllerBase._isZWriteEnabled

ClearVRDisplayObjectControllerBase._shaderColor

ClearVRDisplayObjectControllerBase. shaderColorSpaceYUVtoRGBTransformMatrix

ClearVRDisplayObjectControllerBase. lastTextureId

ClearVRDisplayObjectControllerBase._platformOptions

ClearVRDisplayObjectControllerBase._cameraPositionId

ClearVRDisplayObjectControllerBase._viewMatrixId

ClearVRDisplayObjectControllerBase.clearVRMeshType

ClearVRDisplayObjectControllerBase.forceDisableMeshRendererAtAllTimes

ClearVRDisplayObjectControllerBase. isRegisteredAndInitialized

ClearVRDisplayObjectControllerBase.clearVRShader

ClearVRDisplayObjectControllerBase.genericNonClearVROmniShader

ClearVRDisplayObjectControllerBase. originalMaterial

ClearVRDisplayObjectControllerBase. originalShader

ClearVRDisplayObjectControllerBase. originalActiveState

ClearVRDisplayObjectControllerBase.frameWidth

ClearVRDisplayObjectControllerBase.frameHeight

ClearVRDisplayObjectControllerBase.UpdateApplicationMeshState()

ClearVRDisplayObjectControllerBase.LoadShaders()

ClearVRDisplayObjectControllerBase.ControllerAwake()

ClearVRDisplayObjectControllerBase.Initialize(PlatformOptionsBase, SharedPointersWithSDK,

System.Object)

ClearVRDisplayObjectControllerBase.DoBaseLateUpdate()

ClearVRDisplayObjectControllerBase.UpdateShader()

ClearVRDisplayObjectControllerBase.ReconfigureShader(Shader)

ClearVRDisplayObjectControllerBase.EnableOrDisableStereoscopicRendering(Boolean, Boolean)

ClearVRDisplayObjectControllerBase.EnableOrDisableOESFastPath(Boolean, Boolean)

ClearVRDisplayObjectControllerBase.SetZWrite(Boolean, Boolean)

ClearVRDisplayObjectControllerBase.GetContentFormat()

ClearVRDisplayObjectControllerBase.SetRenderMode(RenderModes)

ClearVRDisplayObjectControllerBase.SetMainColor(Color)

ClearVRDisplayObjectControllerBase.HasLocalShaderParameterChange()

ClearVRDisplayObjectControllerBase.GetTransformationMatrixFromMeshDescriptionStruct()

ClearVRDisplayObjectControllerBase.RecreateNativeTextures()

ClearVRDisplayObjectControllerBase.UpdateNativeTextures(IntPtr, IntPtr, IntPtr)

ClearVRDisplayObjectControllerBase.DestroyNativeTextures()

ClearVRDisplayObjectControllerBase.BindNativeTextureToShader(Texture)

ClearVRDisplayObjectControllerBase.BindNativeChromaTextureToShader(Texture)

ClearVRDisplayObjectControllerBase.BindNativeChromaTexture2ToShader(Texture)

ClearVRDisplayObjectControllerBase.GetTexture()

ClearVRDisplayObjectControllerBase.UnbindAndReleaseNativeTextures()

ClearVRDisplayObjectControllerBase.GetFallbackLayout()

ClearVRDisplayObjectControllerBase.ToString()

ClearVRDisplayObjectControllerBase.OnDestroy()

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRDisplayObjectControllerOVROverlay : ClearVRDisplayObjectControllerMesh, ClearVRDisplayObjectControllerInterface, ClearVRDisplayObjectControllerInterfaceInternal

Properties

meshTextureMode

Declaration

public override MeshTextureModes meshTextureMode { get; set; }

Property Value

TYPE	DESCRIPTION
MeshTextureModes	

Overrides

 ${\bf ClearVRD} is play Object Controller Mesh. mesh Texture Mode$

Implements

ClearVRDisplayObjectControllerInterface

Class ClearVRDisplayObjectControllerSprite

Holds the ClearVRDisplayObjectControllerBase component attached to the gameobject which render the Sprite

Inheritance

System.Object

ClearVRDisplayObjectControllerBase

ClearVRDisplayObjectControllerSprite

Implements

ClearVRDisplayObjectControllerInterface

Inherited Members

ClearVRDisplayObjectControllerBase.hideInAwake

ClearVRDisplayObjectControllerBase.activeFeedIndex

ClearVRDisplayObjectControllerBase._contentFormat

ClearVRDisplayObjectControllerBase. renderMode

ClearVRDisplayObjectControllerBase.renderMode

ClearVRDisplayObjectControllerBase.contentFormat

ClearVRDisplayObjectControllerBase.displayObjectClassType

ClearVRDisplayObjectControllerBase.GetDisplayObjectClassType()

ClearVRDisplayObjectControllerBase.isActive

ClearVRDisplayObjectControllerBase.GetIsActive()

ClearVRDisplayObjectControllerBase.clearVRLayoutManager

ClearVRDisplayObjectControllerBase. isReadyForDestruction

ClearVRDisplayObjectControllerBase._material

ClearVRDisplayObjectControllerBase. isStereoCustomUVsOnKeywordEnabled

ClearVRDisplayObjectControllerBase. isOESFastPathOnKeywordEnabled

ClearVRDisplayObjectControllerBase. isTransparencyOnKeywordEnabled

ClearVRDisplayObjectControllerBase. isPicoVREyeIndexKeywordEnabled

ClearVRDisplayObjectControllerBase. isZWriteEnabled

ClearVRDisplayObjectControllerBase. shaderColor

ClearVRDisplayObjectControllerBase._shaderColorSpaceYUVtoRGBTransformMatrix

ClearVRDisplayObjectControllerBase. lastTextureId

ClearVRDisplayObjectControllerBase._platformOptions

ClearVRDisplayObjectControllerBase._cameraPositionId

ClearVRDisplayObjectControllerBase. viewMatrixId

ClearVRDisplayObjectControllerBase.clearVRMeshType

ClearVRDisplayObjectControllerBase.forceDisableMeshRendererAtAllTimes

ClearVRDisplayObjectControllerBase. isRegisteredAndInitialized

ClearVRDisplayObjectControllerBase.clearVRShader

ClearVRDisplayObjectControllerBase.genericNonClearVROmniShader

ClearVRDisplayObjectControllerBase. originalMaterial

ClearVRDisplayObjectControllerBase. originalShader

ClearVRDisplayObjectControllerBase._originalActiveState

ClearVRDisplayObjectControllerBase.frameWidth

ClearVRDisplayObjectControllerBase.frameHeight

ClearVRDisplayObjectControllerBase.UpdateApplicationMeshState()

ClearVRDisplayObjectControllerBase.LoadShaders()

ClearVRDisplayObjectControllerBase.ControllerAwake()

ClearVRDisplayObjectControllerBase.Initialize(PlatformOptionsBase, SharedPointersWithSDK,

System.Object)

ClearVRDisplayObjectControllerBase.DoBaseLateUpdate()

ClearVRDisplayObjectControllerBase.ReconfigureShader(Shader)

ClearVRDisplayObjectControllerBase.EnableOrDisableStereoscopicRendering(Boolean, Boolean)

ClearVRDisplayObjectControllerBase.EnableOrDisableOESFastPath(Boolean, Boolean)

ClearVRDisplayObjectControllerBase.SetZWrite(Boolean, Boolean)

ClearVRDisplayObjectControllerBase.GetContentFormat()

ClearVRDisplayObjectControllerBase.SetRenderMode(RenderModes)

ClearVRDisplayObjectControllerBase.SetMainColor(Color)

ClearVRDisplayObjectControllerBase.HasLocalShaderParameterChange()

ClearVRDisplayObjectControllerBase.GetTransformationMatrixFromMeshDescriptionStruct()

ClearVRDisplayObjectControllerBase.RecreateNativeTextures()

ClearVRDisplayObjectControllerBase.UpdateNativeTextures(IntPtr, IntPtr, IntPtr)

ClearVRDisplayObjectControllerBase.DestroyNativeTextures()

ClearVRDisplayObjectControllerBase.BindNativeTextureToShader(Texture)

ClearVRDisplayObjectControllerBase.BindNativeChromaTextureToShader(Texture)

ClearVRDisplayObjectControllerBase.BindNativeChromaTexture2ToShader(Texture)

ClearVRDisplayObjectControllerBase.GetTexture()

ClearVRDisplayObjectControllerBase.UnbindAndReleaseNativeTextures()

ClearVRDisplayObjectControllerBase.GetFallbackLayout()

ClearVRDisplayObjectControllerBase.ToString()

ClearVRDisplayObjectControllerBase.OnDestroy()

Namespace: com.tiledmedia.clearvr

Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRDisplayObjectControllerSprite : ClearVRDisplayObjectControllerBase, ClearVRDisplayObjectControllerInterface, ClearVRDisplayObjectControllerInterfaceInternal

Properties

meshTextureMode

Declaration

public override MeshTextureModes meshTextureMode { get; set; }

Property Value

TYPE	DESCRIPTION
MeshTextureModes	

Overrides

ClearVRDisplayObjectControllerBase.meshTextureMode

Methods

EnableOrDisableMeshRenderer(Boolean)

Declaration

public override void EnableOrDisableMeshRenderer(bool argIsEnabled)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	arglsEnabled	

Overrides

ClearVRDisplayObjectControllerBase.EnableOrDisableMeshRenderer(Boolean)

InitializeRendererAndMaterial()

Declaration

protected override void InitializeRendererAndMaterial()

Overrides

ClearVRDisplayObjectControllerBase.InitializeRendererAndMaterial()

IsMeshRendererEnabled()

Declaration

public override bool IsMeshRendererEnabled()

TYPE	DESCRIPTION
System.Boolean	

Overrides

ClearVRDisplayObjectControllerBase.IsMeshRendererEnabled()

UpdateMaterial(Material)

Declaration

public void UpdateMaterial(Material m)

Parameters

TYPE	NAME	DESCRIPTION
Material	m	

UpdateShader()

Declaration

public override void UpdateShader()

Overrides

ClearVRDisplayObjectControllerBase.UpdateShader()

UpdateShaderAndMaySetDirty(Shader, Boolean)

Declaration

public void UpdateShaderAndMaySetDirty(Shader shader, bool shallSetDirty)

Parameters

TYPE	NAME	DESCRIPTION
Shader	shader	
System.Boolean	shallSetDirty	

Implements

ClearVRDisplayObjectControllerInterface

Class

ClearVRDisplayObjectControllerUnmanagedM esh

The ClearVRDisplayObjectControllerUnmanagedMesh is to be used to render a video on a custom mesh, NOT managed by the ClearVRPlayer. This control properly update the shader parameters, in sync with the video frame currently displayed on the object, but does not interact with the mesh itself besides attaching the proper Material (and associated Shader) to it.

You should subscribe to the Clear VRD is play Object Events to be notified of essential Display Object life-cycle events.

A WARNING

This is considered an advanced Display Object. It is fundamentally incompatible when playing ClearVR content.

Inheritance

System.Object

ClearVRDisplayObjectControllerBase

ClearVRDisplayObjectControllerUnmanagedMesh

Implements

ClearVRDisplayObjectControllerInterface

Inherited Members

ClearVRDisplayObjectControllerBase.hideInAwake

ClearVRDisplayObjectControllerBase.activeFeedIndex

ClearVRDisplayObjectControllerBase. contentFormat

ClearVRDisplayObjectControllerBase. renderMode

ClearVRDisplayObjectControllerBase.renderMode

ClearVRDisplayObjectControllerBase.contentFormat

ClearVRDisplayObjectControllerBase.displayObjectClassType

ClearVRDisplayObjectControllerBase.GetDisplayObjectClassType()

ClearVRDisplayObjectControllerBase.isActive

ClearVRDisplayObjectControllerBase.GetIsActive()

ClearVRDisplayObjectControllerBase.clearVRLayoutManager

ClearVRDisplayObjectControllerBase._isReadyForDestruction

ClearVRDisplayObjectControllerBase. material

ClearVRDisplayObjectControllerBase. isStereoCustomUVsOnKeywordEnabled

ClearVRDisplayObjectControllerBase._isOESFastPathOnKeywordEnabled

ClearVRDisplayObjectControllerBase._isTransparencyOnKeywordEnabled

ClearVRDisplayObjectControllerBase. isPicoVREyeIndexKeywordEnabled

ClearVRDisplayObjectControllerBase. isZWriteEnabled

ClearVRDisplayObjectControllerBase. shaderColor

ClearVRDisplayObjectControllerBase. shaderColorSpaceYUVtoRGBTransformMatrix

ClearVRDisplayObjectControllerBase. lastTextureId

ClearVRDisplayObjectControllerBase. platformOptions

ClearVRDisplayObjectControllerBase. cameraPositionId

ClearVRDisplayObjectControllerBase. viewMatrixId

ClearVRDisplayObjectControllerBase.clearVRMeshType

ClearVRDisplayObjectControllerBase.forceDisableMeshRendererAtAllTimes

ClearVRDisplayObjectControllerBase. isRegisteredAndInitialized

ClearVRDisplayObjectControllerBase.clearVRShader

ClearVRDisplayObjectControllerBase.genericNonClearVROmniShader

ClearVRDisplayObjectControllerBase. originalMaterial

ClearVRDisplayObjectControllerBase. originalShader

ClearVRDisplayObjectControllerBase. originalActiveState

ClearVRDisplayObjectControllerBase.frameWidth

ClearVRDisplayObjectControllerBase.frameHeight

ClearVRDisplayObjectControllerBase.UpdateApplicationMeshState()

ClearVRDisplayObjectControllerBase.LoadShaders()

ClearVRDisplayObjectControllerBase.ControllerAwake()

ClearVRDisplayObjectControllerBase.Initialize(PlatformOptionsBase, SharedPointersWithSDK,

System.Object)

ClearVRDisplayObjectControllerBase.DoBaseLateUpdate()

ClearVRDisplayObjectControllerBase.ReconfigureShader(Shader)

ClearVRDisplayObjectControllerBase.EnableOrDisableStereoscopicRendering(Boolean, Boolean)

ClearVRDisplayObjectControllerBase.EnableOrDisableOESFastPath(Boolean, Boolean)

ClearVRDisplayObjectControllerBase.SetZWrite(Boolean, Boolean)

ClearVRDisplayObjectControllerBase.GetContentFormat()

ClearVRDisplayObjectControllerBase.SetRenderMode(RenderModes)

ClearVRDisplayObjectControllerBase.SetMainColor(Color)

ClearVRDisplayObjectControllerBase.RecreateNativeTextures()

ClearVRDisplayObjectControllerBase.UpdateNativeTextures(IntPtr, IntPtr, IntPtr)

ClearVRDisplayObjectControllerBase.DestroyNativeTextures()

ClearVRDisplayObjectControllerBase.BindNativeTextureToShader(Texture)

ClearVRDisplayObjectControllerBase.BindNativeChromaTextureToShader(Texture)

ClearVRDisplayObjectControllerBase.BindNativeChromaTexture2ToShader(Texture)

ClearVRDisplayObjectControllerBase.GetTexture()

ClearVRDisplayObjectControllerBase.UnbindAndReleaseNativeTextures()

ClearVRDisplayObjectControllerBase.GetFallbackLayout()

ClearVRDisplayObjectControllerBase.ToString()

ClearVRDisplayObjectControllerBase.OnDestroy()

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

 $\frac{\text{public class ClearVRD}}{\text{clearVRD}} \\ \text{DisplayObjectControllerUnmanagedMesh}: \\ \text{ClearVRD}\\ \text{isplayObjectControllerInterface}, \\ \text{ClearVRD}\\ \text{ispla$

Fields

_meshRenderer

Declaration

protected MeshRenderer _meshRenderer

Field Value

TYPE	DESCRIPTION
MeshRenderer	

HorizontalFlip

Declaration

public bool HorizontalFlip

Field Value

TYPE	DESCRIPTION
System.Boolean	

VerticalFlip

Declaration

public bool VerticalFlip

Field Value

TYPE	DESCRIPTION
System.Boolean	

Properties

meshTextureMode

(c) Tiledmedia B.V. 2017-2022

Declaration

public override MeshTextureModes meshTextureMode { get; set; }

Property Value

TYPE	DESCRIPTION
MeshTextureModes	

Overrides

ClearVRDisplayObjectControllerBase.meshTextureMode

Methods

EnableOrDisableMeshRenderer(Boolean)

Declaration

public override void EnableOrDisableMeshRenderer(bool argIsEnabled)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	argIsEnabled	

Overrides

ClearVRDisplayObjectControllerBase.EnableOrDisableMeshRenderer(Boolean)

GetTransformationMatrixFromMeshDescriptionStruct()

Declaration

protected override Matrix4x4 GetTransformationMatrixFromMeshDescriptionStruct()

Returns

ТҮРЕ	DESCRIPTION
Matrix4x4	

Overrides

ClearVRDisplayObjectControllerBase.GetTransformationMatrixFromMeshDescriptionStruct()

HasLocalShaderParameterChange()

Declaration

protected override bool HasLocalShaderParameterChange()

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

ClearVRDisplayObjectControllerBase.HasLocalShaderParameterChange()

InitializeRendererAndMaterial()

Declaration

protected override void InitializeRendererAndMaterial()

Overrides

ClearVRDisplayObjectControllerBase.InitializeRendererAndMaterial()

IsMeshRendererEnabled()

Declaration

public override bool IsMeshRendererEnabled()

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

ClearVRDisplayObjectControllerBase.IsMeshRendererEnabled()

UpdateShader()

Declaration

public override void UpdateShader()

Overrides

ClearVRDisplayObjectControllerBase.UpdateShader()

Implements

ClearVRDisplayObjectControllerInterface

Class ClearVRDisplayObjectEvent

Since v9.0 The event as emitted by aClearVRDisplayObject.

Inheritance

System.Object

ClearVRDisplayObjectEvent

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRDisplayObjectEvent : object

Properties

clearVRDisplayObjectEventType

The event type.

Declaration

public ClearVRDisplayObjectEventTypes clearVRDisplayObjectEventType { get; }

Property Value

TYPE	DESCRIPTION
ClearVRDisplayObjectEventTypes	

clearVRMessage

The message payload.

Declaration

public ClearVRMessage clearVRMessage { get; }

Property Value

TYPE	DESCRIPTION
ClearVRMessage	

message

A convenience alias getter for the message that this event holds.

Declaration

public ClearVRMessage message { get; }

Property Value

TYPE	DESCRIPTION
ClearVRMessage	The ClearVRMessage

type

A convenience alias getter for clearVRDisplayObjectEventType.

Declaration

public ClearVRDisplayObjectEventTypes type { get; }

Property Value

TYPE	DESCRIPTION
ClearVRDisplayObjectEventTypes	

Methods

ToString()

Returns the fields of this object as a properly formatted string.

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	A properly formatted string.

Class ClearVRDisplayObjectEvents

The ClearVRDisplayObjects Events listener prototype.

Inheritance

System.Object

ClearVRDisplayObjectEvents

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRDisplayObjectEvents : UnityEngine.Events.UnityEvent<ClearVRPlayer, ClearVRDisplayObjectControllerBase, ClearVRDisplayObjectEvent>

Class ClearVRDisplayObjectEventsInternal

Inheritance

System.Object

ClearVRDisplayObjectEventsInternal

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRDisplayObjectEventsInternal :

UnityEngine.Events.UnityEvent<ClearVRDisplayObjectControllerBase, ClearVRDisplayObjectEvent>

Enum ClearVRDisplayObjectEventTypes

Since v9.0 Throughout the lifecycle of aClearVRDisplayObject, it emits various events. One can subscribe to these events by using the AddEventListener method on clearVRPlayer.clearVRDisplayObjectEvents. This enum describes all the events that can be emitted.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ClearVRDisplayObjectEventTypes : int

Fields

NAME	DESCRIPTION
ActiveStateChanged	Emitted when the associated DisplayObject has changed its Active state. Note that this is NOT the same as the gameObject.activeSelf property.
ClassTypeChanged	Emitted when the ClearVRDisplayObjectClassType on the associated DisplayObject has changed.
ContentFormatChanged	The ContentFormat of the associated DisplayObject has changed. Please refer to ContentFormat for details.
FirstFrameRendered	This event is emitted every time the mesh has changed. For example, after the content is loaded and started playing, but also after an ABR event or after a SwitchContent or SetLayout API call has completed.
None	This is an internal event and should not be used. It will never be triggered.
RenderModeChanged	The RenderMode of the associated DisplayObject has changed. Please refer to RenderModes for details.
Subtitle	Emitted when the associated DisplayObject receive subtitle information.

Enum ClearVRDRMLicenseServerTypes

Enum listing the various DRM license server types.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ClearVRDRMLicenseServerTypes : int

Fields

NAME	DESCRIPTION
DRMLicenseServerViaccessOrcaTokenBased	Viaccess-orca token based license server type.
DRMLicenseServerWidevineGeneric	Generic Widevine license server type.
Unspecified	The license server type is unspecified. This is typically selected when doing header-based encryption.

Class ClearVREvent

The primary interface for handling events from the ClearVRPlayer in your own application. As soon as you have created your CLearVRPlayer object, one shuuld subscribe an EventListener through clearVRPlayer.clearVREvents.AddListener(this.02);

Inheritance

System.Object ClearVREvent

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVREvent : object

Properties

clearVRAsyncRequestResponse

Declaration

public System.Object clearVRAsyncRequestResponse { get; }

Property Value

TYPE	DESCRIPTION
System.Object	

message

The message that this event holds.

Declaration

public ClearVRMessage message { get; set; }

Property Value

TYPE	DESCRIPTION
ClearVRMessage	The ClearVRMessage

optionalArguments

public object[] optionalArguments { get; }

Property Value

TYPE	DESCRIPTION
System.Object[]	

type

Declaration

public ClearVREventTypes type { get; set; }

Property Value

TYPE	DESCRIPTION
ClearVREventTypes	

Methods

GetIsStateChangedEvent()

Convenience method that allows you to quickly distinguish between a StateChanged event or any other type of event.

Declaration

public bool GetIsStateChangedEvent()

Returns

TYPE	DESCRIPTION
System.Boolean	True if this is a StateChanged event, false otherwise.

GetStateName()

Produces the state name of the ClearVREvent IF it is a state changed event.

Warning: The returned String will be null if the ClearVREventType is NOT a state.

Declaration

public String GetStateName()

Returns

TYPE	DESCRIPTION
String	The name of the state if the event is a state as a String. Otherwise null.

HasWarningOrFatalErrorMessage()

Convenience method to distinguish between Warning/FatalError events and Info events.

Declaration

public bool HasWarningOrFatalErrorMessage()

Returns

TYPE	DESCRIPTION
System.Boolean	True if the event is a Warning or FatalError, false otherwise.

Print()

Convenience method that prints this object as a pretty string to the Unity console as a debug message.

Declaration

public void Print()

PrintShort()

Convenience method that prints this object as a pretty string to the Unity console as a debug message.

Declaration

public void PrintShort()

ToString()

Returns the fields of this object as a properly formatted string.

Declaration

public override String ToString()

Returns

ТҮРЕ	DESCRIPTION
String	A properly formatted string.

Class ClearVREvents

The ClearVREvents listener prototype.

Inheritance

System.Object

ClearVREvents

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVREvents: UnityEngine.Events.UnityEvent<ClearVRPlayer, ClearVREvent>

Enum ClearVREventTypes

Throughout the life-cycle of a ClearVRPlayer object, several events are generated. Attach a listener using clearVRPlayer.clearVREvents.AddListener(CbClearVREvent); to be informed of these events in your own class. The signature of this event handler is:

void CbClearVREvent(ClearVRPlayer argClearVRPlayer, ClearVREvent argClearVREvent)

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ClearVREventTypes: int

NAME	DESCRIPTION
ABRLevelActivated	The adaptive bitrate mechanism has trigged a switch between representation (quality) layers available in this clip.
ABRSwitch	The adaptive bitrate mechanism has trigged a switch between representation (quality) layers available in this clip. Since v8.0, this has been renamed to ABRLevelActivated
ActiveTracksChanged	A video, audio or subtitle track has switched to a different one.
AudioFocusChanged	The app has lost or gained audio focus. Ex: a call is coming in.
AudioTrackSwitched	The Audio track has changed. You can use clearVREvent.message.ParseAudioTrackChanged() API to figure out which audio track index is currently selected.
CallCoreCompleted	Debug call core has completed.
(c) Tiledmedia B.V. 2017-2022	114

NAME	DESCRIPTION
ContentFormatChanged	Since v4.0 The format of the content has changed. This event is triggered after loading the first clip and when the content format changed after a SwitchContent() API call or an ABRLevelChanged event (for example if there are stereoscopic and monoscopic renditions available in an HLS ladder). You can use the clearVRPlayer.mediaInfo.GetContentFormat() to know what the new ContentFormat is. MARNING Since v9.x, this event is not emitted anymore, except when the LayoutManager is running in legacy mode for backwards compatibility. This event is now available per DisplayObject. Refer to ClearVRDisplayObjectEvent for more information.
ContentSwitched	A SwitchContent() request has been completed.
FirstFrameRendered	The first frame of the video has been rendered, or the first frame of the <i>new</i> video after SwitchContent() has been rendered. Note that by the time you receive this callback, the first frame is actually already rendered. Any action taken will only be effectuated when rendering the <i>second</i> video frame.
	▲ WARNING Since v9.x, this event is not emitted anymore, except when the LayoutManager is running in legacy mode for backwards compatibility. This event is now available per DisplayObject. Refer to ClearVRDisplayObjectEvent for more information.
GenericMessage	A generic message can contain information information, a warning or a fatal error. You can check the embedded ClearVRMessage for details.

NAME	DESCRIPTION
MediaInfoParsed	The media info of the selected clip has been parsed and you can now query its parameters.
None	Used internally, do not use
ParsingMediaInfo	Transient state, the ClearVRPlayer is parsing the media info of the selected clip.
PrewarmCacheCompleted	Cache prewarm has completed.
RenderModeChanged	This event is triggered when the way the video is being displayed to the user has changed. This is specifically applicable to stereoscopic content playback.
	▲ WARNING Since v9.x, this event is not emitted anymore, except when the LayoutManager is running in legacy mode for backwards compatibility. This event is now available per DisplayObject. Refer to ClearVRDisplayObjectEvent for more information.
ResumingPlaybackAfterApplicationPaused	Renamed in v9.1. This event has been renamed to ResumingPlaybackAfterApplicationRegainedFocus in v9.1.

NAME	DESCRIPTION
ResumingPlaybackAfterApplicationRegainedFocus	Since v9.1 (was known as ResumingPlaybackAfterApplicationPaused in older versions). Triggered when the player is resuming after the application regained focus again after it was has previously lost focus (e.g. because it was pushed to the background). SuspendingPlaybackAfterApplicationLostFocus is triggered when the application has lost focus.
	This event is only emitted when platformOptions.applicationFocusAndPauseHandling is set to its default value Recommended.
SetLayoutCompleted	Set feed layout call has completed.
StateChangedBuffering	The player's internal buffers have depleted and buffering is required to resume playback.
StateChangedContentPreparedForPlayout	The content is buffered and prepared for playout.
StateChangedCorePrepared	By now, the ClearVRPlayer is ready to load content.
StateChangedFinished	The current clip has finished playback (e.g. the end of the clip was reached). In this state, no new frames are generated until you seek to a new position in the content.
StateChangedInitialized	Transient state while initialization the ClearVRPlayer. Ready to load content.

NAME	DESCRIPTION
StateChangedInitializing	Basic initialization of the ClearVRPlayer, do not interfer.
StateChangedPaused	Video playback has paused.
StateChangedPausing	Transient state while playback is pausing. It can take a couple of frames before playback has paused.
StateChangedPlaying	In this state, the player stack is rendering audio and video.
StateChangedPreparingContentForPlayout	The ClearVRPlayer is loading the selected clip.
StateChangedPreparingCore	Transient state while loading content. The next state will be StateChangedCorePrepared
StateChangedSeeking	Transient state while seeking.
StateChangedStopped	The ClearVRPlayer object is completely cleaned-up, has released all its resources and is ready for destruction. You will receive no further events after this state change.
StateChangedStopping	The ClearVRPlayer is in the process of destruction. It is essential for this to complete, so wait for StateChangedStopped until you Destroy() the ClearVRPlayer object!

NAME	DESCRIPTION
StateChangedSwitchingContent	Transient state while switching content. When done, you will receive the ContentSwitched event. Playback has only commenced when you have received the StateChangedPlaying event.
StateChangedUninitialized	Initial state of the player
StereoModeSwitched	This event is triggered when there is a change in which eye(s) of the video are retrieved. For example, the video for the right eye is no longer fetched due to poor network conditions or, vice-versa, the right eye video is being fetched because of improved network conditions.
SuspendingPlaybackAfterApplicationLostFocus	Since v9.1 Triggered when the player is suspending after the application has lost focus (e.g. was pushed to the background). ResumingPlaybackAfterApplicationRegainedFocus is triggered after the application has regained focus again.
	This event is only emitted when platformOptions.applicationFocusAndPauseHandling is set to its default value Recommended.
SuspendingPlaybackBeforeApplicationPaused	Added in v9.0 Renamed in v9.1 This event has been renamed to SuspendingPlaybackAfterApplicationLostFocus in v9.1.
UnableToInitializePlayer	Triggered when the ClearVRPlayer object was unable to initialize. Typically the result of a faulty URL or missing license file data.

Enum ClearVRFishEyeStereoTypes

Namespace	:: com.tileamedia.clearvr
Assembly:	Assembly-CSharp.dll

Syntax

pul	blic	enum	Clear	VRF	-ishEy	yeStere	oTypes	3:	int
-----	------	------	-------	-----	--------	---------	--------	----	-----

NAME	DESCRIPTION
StereoTypeMono	
StereoTypeStereoSideBySide	
StereoTypeStereoTopBottom	

Enum ClearVRFishEyeTypes

Various Fish Eye Mesh Types are supported.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ClearVRFishEyeTypes : int

NAME	DESCRIPTION
EquiDistant	Equidistant
EquiSolid	Equisolid
NotSet	Not set: indicate that the Fish Eye parameters were not set.
Polynomial	Polynomial model

Class ClearVRLayoutManager

For details on the ClearVRLayoutManager and how to use it, please refer to the documentationhere. On runtime, you never access the ClearVRLayoutManager directly, you always interface with the ClearVRPlayer instead.

Inheritance

System.Object

ClearVRLayoutManager

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRLayoutManager: MonoBehaviour

Fields

LEGACY_LAYOUT_NAME

Declaration

public static readonly string LEGACY_LAYOUT_NAME

Field Value

TYPE	DESCRIPTION
System.String	

Enum ClearVRMeshTypes

Devices the type of a mesh. This allows you to infer the actual shape of the mesh

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ClearVRMeshTypes: int

NAME	DESCRIPTION
Cubemap	360 degree cubemap.
Cubemap180	180 degree cubemap (half a cube).
ERP	360 degree ERP sphere.
ERP180	180 degree ERP semi-sphere.
FishEye	Custom, fish eye compatible mesh.
Planar	a flat, large quad.
Rectilinear	Simple rectangular quad.
Unknown	Unknown mesh type, not allowed.

Class ClearVRMessage

Each ClearVREvent contains a ClearVRMessage. The ClearVRMessage contains valuable information about the event. Please refer to ClearVRMessageCodes for details on how to interpret the various codes. The code field can hold a value of eitherClearVRMessageCodes or ClearVRCoreErrorCodes Use the GetIsClearVRCoreErrorCode() API to figure our whether the code is a ClearVRCoreErrorCode or ClearVRMessageCode. Please refer to the GetClearVRMessageCode() API to convert the Integer into its ClearVRMessageCode and GetClearVRCoreErrorCode() API to convert the Integer into its ClearVRCoreErrorCode respectively.

Inheritance

System.Object

ClearVRMessage

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRMessage : object

Properties

code

The message code as an integer. The integer can refer to both aClearVRCoreErrorCodes as well as a ClearVRMessageCodes.

Declaration

public int code { get; set; }

Property Value

TYPE	DESCRIPTION
System.Int32	The code as an integer.

message

The message as a String, possibly containing additional information about the event this ClearVRMessage was attached to. Note that this field is typically empty in case of a ClearVRCoreWrapperGenericOK message code

Declaration

public String message { get; set; }

Property Value

TYPE	DESCRIPTION
String	The message as a string.

type

Get the type of this message. The type is either FatalError, Warning or Info.

Declaration

public ClearVRMessageTypes type { get; set; }

Property Value

TYPE	DESCRIPTION
ClearVRMessageTypes	The type of the message.

Methods

GetClearVRCoreErrorCode()

Converts code into ClearVRCoreErrorCode. This API will returnUnknownError in case the code field value cannot be converted. If this were to be the case, the code field was actually a ClearVRMessageCodes instead.

Declaration

public ClearVRCoreErrorCodes GetClearVRCoreErrorCode()

Returns

TYPE	DESCRIPTION
ClearVRCoreErrorCodes	The ClearVRCoreErrorCode equivalent of the code field value.

GetClearVRMessageCode()

Converts code into ClearVRMessageCodes. This API will returnUnknown in case the code field value cannot be converted. If this were to be the case, the code field was actually a ClearVRCoreErrorCodes instead.

public ClearVRMessageCodes GetClearVRMessageCode()

Returns

TYPE	DESCRIPTION
ClearVRMessageCodes	The ClearVRMessageCodes equivalent of the code field value.

GetCode()

Helper method that returns the code field as the String equivalent from the matching enum. This can be called on any code value.

Declaration

public String GetCode()

Returns

TYPE	DESCRIPTION
String	String representation of the enum that matches the code.

GetFullMessage()

Declaration

public String GetFullMessage()

Returns

TYPE	DESCRIPTION
String	

GetIsClearVRCoreErrorCode()

Check whether the code is a ClearVRCoreErrorCode or not.

Declaration

public bool GetIsClearVRCoreErrorCode()

Returns

TYPE	DESCRIPTION	
System.Boolean	True if the code holds a ClearVRCoreErrorCode, false if it holds a ClearVRMessageCode.	

GetIsFatalError()

Convenience method to determine whether this message contains a FatalError

Declaration

public bool GetIsFatalError()

Returns

ТҮРЕ	DESCRIPTION
System.Boolean	True if this is a FatalError, false otherwise.

GetIsInfo()

Convenience method to determine whether this message contains an informational message

Declaration

public bool GetIsInfo()

Returns

TYPE	DESCRIPTION
System.Boolean	True if this is an informational messgae, false otherwise

GetIsSuccess()

Declaration

public bool GetIsSuccess()

Returns

TYPE	DESCRIPTION
System.Boolean	

GetIsWarning()

Convenience method to determine whether this message contains a Warning

Declaration

public bool GetIsWarning()

Returns

TYPE	DESCRIPTION
System.Boolean	True if this is a Warning, false otherwise

ParseAudioTrackChanged(out AudioTrackAndPlaybackParameters)

If code == ClearVRCoreWrapperAudioTrackChanged this will parse the serialized payload in the message field. After successful parsing, the AudioTrackAndPlaybackParameters will be set to a non-null value. In case of failure, it will be set to null.

Declaration

public bool ParseAudioTrackChanged(out AudioTrackAndPlaybackParameters
argAudioTrackAndPlaybackParameters)

Parameters

TYPE	NAME	DESCRIPTION
AudioTrackAndPlaybackParameters	argAudioTrackAndPlaybackParameters	The parameters of the currently active audio track.

Returns

TYPE	DESCRIPTION
System.Boolean	True if message was successfully parsed, false otherwise.

ParseClearVRCoreWrapperABRLevelActivated(out ABRLevel)

Parses the payload of the ClearVRCoreWrapperABRLevelActivated ClearVRMessage.

public bool ParseClearVRCoreWrapperABRLevelActivated(out ABRLevel argABRLevel)

Parameters

TYPE	NAME	DESCRIPTION
ABRLevel	argABRLevel	The ABR level

Returns

TYPE	DESCRIPTION
System.Boolean	True if the message could be successfully parsed, false otherwise. The value of the out parameter is undefined if false is returned. If true is returned, it will never be null.

ParseClearVRCoreWrapperActiveTracksChanged(out ContentInfo)

Parses the payload of the ClearVRCoreWrapperActiveTracksChanged ClearVRMessage.

Declaration

public bool ParseClearVRCoreWrapperActiveTracksChanged(out ContentInfo contentInfo)

Parameters

TYPE	NAME	DESCRIPTION
ContentInfo	contentInfo	The ContentInfo object that will hold information about the current ContentItem.

Returns

TYPE	DESCRIPTION
System.Boolean	True if the message could be successfully parsed, false otherwise. The value of the out parameter is undefined if false is returned. If true is returned, it will never be null.

ParseClearVRCoreWrapperVideoDecoderCapabilities(out VideoDecoderCapabilities)

Convenience method that parses reported VideoDecoderCapabilities.

public bool ParseClearVRCoreWrapperVideoDecoderCapabilities(out VideoDecoderCapabilities argVideoDecoderCapabilities)

Parameters

TYPE	NAME	DESCRIPTION
VideoDecoderCapabilities	argVideoDecoderCapabilities	out argument that will contain the VideoDecoderCapabilities or null if parsing was unsuccessful.

Returns

TYPE	DESCRIPTION
System.Boolean	True if message was successfully parsed, false otherwise.

ParseClearVRSubtitle(out ClearVRSubtitle)

Parses the payload of the ClearVRCoreWrapperSubtitle ClearVRMessage.

Declaration

public bool ParseClearVRSubtitle(out ClearVRSubtitle clearVRSubtitle)

Parameters

TYPE	NAME	DESCRIPTION
ClearVRSubtitle	clearVRSubtitle	The ClearVRSubtitle object that will hold information about the current subtitle.

Returns

TYPE	DESCRIPTION
System.Boolean	True if the message could be successfully parsed, false otherwise. The value of the out parameter is undefined if false is returned. If true is returned, it will never be null.

ParseSyncStateChanged(out SyncStateChanged)

If code == ClearVRCoreWrapperSyncStateChanged this will parse the serialized payload in the

message field. After successful parsing, the SyncStateChanged will be set to a non-null value. In case of failure, it will be set to null.

Declaration

public bool ParseSyncStateChanged(out SyncStateChanged argSyncStateChanged)

Parameters

TYPE	NAME	DESCRIPTION
SyncStateChanged	argSyncStateChanged	

Returns

TYPE	DESCRIPTION
System.Boolean	True if message was successfully parsed, false otherwise.

PrintFullMessage()

Convenience method that prints the message to the Unity console as a debug message.

Declaration

public void PrintFullMessage()

ToString()

Declaration

public override String ToString()

Returns

ТҮРЕ	DESCRIPTION
String	

Enum ClearVRMessageCodes

Every ClearVREvent that you receive contains a ClearVRMessage. Each ClearVRMessage holds a code, that can either be:

- 1. a number from the ClearVRMessageCodes enum
- 2. a number from the ClearVRCoreErrorCodes enum This enum describes the ClearVRMessageCodes.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ClearVRMessageCodes: int

NAME	DESCRIPTION
APINotSupportedOnThisPlatform	The API is not implemented on this curr
APIObsolete	The API is deprecated and cannot be u check the docs for details.
ClearVRCoreWrapperABRLevelActivated	Info. An ABR level has been activated. calling clearVRMessage.ParseClearVRCoreW
ClearVRCoreWrapperActiveTracksChanged	Info. This event indicates that the active changed. This event will be triggered af for both the first clip and after each Switriggered when an ABR event happened calling bool clearVRMessage.ParseClearVRCoreWContentInfo).
ClearVRCoreWrapperAlreadyInitialized	Fatal. The core library is already initializ application logic problem (e.g. trying to twice)

NAME	DESCRIPTION
ClearVRCoreWrapperAudioDecoderCannotDecodeSample	Fatal. Audio decoder was unable to dec
ClearVRCoreWrapperAudioDecoderDecodingFailure	Fatal. The aduioo decoder threw a gene
ClearVRCoreWrapperAudioDecoderFormatNotSupported	Fatal. No audio decoder was found that
ClearVRCoreWrapperAudioDecoderNotInitialized	Fatal. Audio decoder fatiled to initialize.
ClearVRCoreWrapperAudioFocusGained	
ClearVRCoreWrapperAudioFocusLost	
ClearVRCoreWrapperAudioTrackChanged	Info. Audio track changed successfully. ClearVRMessage.ParseAudioTrackChathe ClearVRMessage's message field.
ClearVRCoreWrapperCallApp	Internal. You will never receive this mes
ClearVRCoreWrapperCannotDecodeFrame	Fatal. The video decoder was unable to
ClearVRCoreWrapperCannotGenerateDeviceAppId	Fatal. Unable to generate anonymous d
ClearVRCoreWrapperCannotSwitchAudioTrack	Warning. Unable to switch audio track.
ClearVRCoreWrapperCannotSwitchContent	Warning. Unable to switch content.
ClearVRCoreWrapperClearVRCoreStateBuffering	Internal. You will never receive this mes

NAME	DESCRIPTION
ClearVRCoreWrapperClearVRCoreStateFinished	Internal. You will never receive this mes
ClearVRCoreWrapperClearVRCoreStateInitialized	Internal. You will never receive this mes
ClearVRCoreWrapperClearVRCoreStateInitializing	Internal. You will never receive this mes
ClearVRCoreWrapperClearVRCoreStatePaused	Internal. You will never receive this mes
ClearVRCoreWrapperClearVRCoreStatePausing	Internal. You will never receive this mes
ClearVRCoreWrapperClearVRCoreStateRunning	Internal. You will never receive this mes
ClearVRCoreWrapperClearVRCoreStateSeeking	Internal. You will never receive this mes
ClearVRCoreWrapperClearVRCoreStateStopped	Internal. You will never receive this mes
ClearVRCoreWrapperClearVRCoreStateSwitchingContent	Internal. You will never receive this mes
ClearVRCoreWrapperClearVRCoreStateUninitialized	Internal. You will never receive this mes
ClearVRCoreWrapperContentDoesNotSupportSeek	Warning. Attempting to seek in a clip the
ClearVRCoreWrapperContentLoadingTimeout	Fatal. A timeout was triggered while loa connection issues?

NAME	DESCRIPTION
ClearVRCoreWrapperDeviceNotSupported	This device is not supported. Typically, blacklisted. Details on why the device is attached ClearVRMessage (String) mes
ClearVRCoreWrapperFatalException	Fatal. A fatal exception occurred in the
ClearVRCoreWrapperGenericInfo	Info. Generic informational message.
ClearVRCoreWrapperGenericOK	Info. Generic OK message. You will rec to a request that was serviced successf message field will typically be empty.
ClearVRCoreWrapperInitializationTimeout	Fatal. Playout was unable to succesfully See message for details.
ClearVRCoreWrapperInvalidOrUnsupportedProxySettings	Fatal. The provided proxy settings are e support.
ClearVRCoreWrapperInvalidState	Warning. Attempting to call an API that state.
ClearVRCoreWrapperNoHardwareVideoDecoderAvailable	Fatal. No hardware video decoder found decoder support is mandatory for HEVC
ClearVRCoreWrapperNonFatalClearVRCoreException	Warning. A non-fatal exception (warning library.
ClearVRCoreWrapperNotProperlyInitialized	Fatal. Unable to correctly initialize librar

NAME	DESCRIPTION
ClearVRCoreWrapperPrepareContentForPlayoutCancelled	The player was asked (by the applicatio to stop while it was preparing playback
ClearVRCoreWrapperRequestCancelled	Warning. An asynchronous request has
ClearVRCoreWrapperSpatialAudioNotSupported	Warning. Spatial audio track selected, be platform.
ClearVRCoreWrapperStereoscopicModeChanged	Info. Stereoscopic mode changed. This Remember that ClearVR content allows on stereoscopic content, significantly re trying to playback stereoscopic content stereoscopic rendering. This is NOT the
ClearVRCoreWrapperSubtitle	
ClearVRCoreWrapperSyncStateChanged	Info. The sync state changed.
ClearVRCoreWrapperTestContentSupportedInternalReport	Internal. You will never receive this mes
ClearVRCoreWrapperUnableToParseMediaInfo	Fatal. Unable to parse media info, cann details.
ClearVRCoreWrapperUnspecifiedFatalError	Fatal. An unspecified error occurred in t library. Unable to continue.
ClearVRCoreWrapperUnspecifiedWarning	Warning. An unexpected, unspecified w check the message payload for details.
ClearVRCoreWrapperVideoDecoderCannotConfigureDecoder	Fatal. Unable to configure the video dec

NAME	DESCRIPTION
ClearVRCoreWrapperVideoDecoderCapabilities	Info. VIdeo decoder capabilities reported ClearVRmessage.ParseClearVRCoreW for details on how to parse the ClearVR
ClearVRCoreWrapperVideoDecoderConfigureDecoderThrewWarning	There is the possibility that the video de configure-state. We will retry this a cour configurations) before we give up. Note message, you should treat it as a warning the configure will be successful at the ne
ClearVRCoreWrapperVideoDecoderDecodingFailure	Fatal. The video dcoder threw a fatal en
ClearVRCoreWrapperVideoDecoderDoesNotSupportProfileOrLevel	Fatal. The video decoder does not suppose mimetype/profile/level combination. For decoding up to 1080p while the content
ClearVRCoreWrapperVideoDecoderFrameDropped	Warning. The video decoder dropped a video data.
ClearVRCoreWrapperVideoDecoderFrameWithoutMetadataDropped	There is the remote possibility that a de associated with it. In that case, the vide 1. It is extremely unlikely that this even us if you see this event passing b
ClearVRCoreWrapperVideoDecoderInputOverflow	Warning. The video decoder input is over frames are produced faster than the vid can result in a lagging view-port. See al ClearVRMessageCodes.ClearVRCoreV below.
ClearVRCoreWrapperVideoDecoderNotInitialized	Fatal. Unable to initialize video decoder decoder limitation reached?

NAME	DESCRIPTION
ClearVRCoreWrapperVideoDecoderOutputOverflow	Warning. The video decoder output is o application cannot consume video framindicates severe performance issues in application framerate). See also ClearVRMessageCodes.ClearVRCoreV below.
learVRCoreWrapperVideoDecoderSlowPriming	Priming the video decoder took unexpending the video decoder took unexpending the primary of the
ClearVRCoreWrapperVideoDecoderTooSlow	A message with this code indicates the up with the video framerate. For example, the color of th
ClearVRCoreWrapperVideoDRMGenericError	Fatal. A generic DRM-related error occu
ClearVRCoreWrapperVideoDRMSchemeUnsupported	Fatal. The selected video DRM scheme
ClearVRCoreWrapperVideoDRMSessionDeniedByServer	Fatal. The DRM license server denied t
ClearVRCoreWrapperVideoDRMSessionEstablished	Info. Video DRM sessions was establish

NAME	DESCRIPTION
ClearVRCoreWrapperVideoDRMUnableToProvision	Fatal. Unable to provision DRM session Required protention level not supported
ClearVRCoreWrapperVideoMimetypeNotSupportBySDK	Fatal. You tried to play back a video mir this SDK.
ClearVRCorWrapperOpenGLVersionInfo	Info. Legacy message code that should OpenGL version formatted as 2.x or 3.x
GenericFatalError	A generic fatal error. This indicates an unhappened for which no dedicated Clear defined. As this is fatla, playback will be
GenericOK	Generic OK. This is always an informati more information in the ClearVRMessaç however, be empty).
GenericWarning	A generic warning. The message body warning.
SetRenderModeFailed	Unable to set requested render mode. I stereoscopic rendering on a monoscopi
Unknown	An unknown message code. You will ne

Enum ClearVRMessageTypes

A ClearVRMessage is of certain type. See also ClearVRMessage class for details.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ClearVRMessageTypes: int

NAME	DESCRIPTION	
FatalError	A fatal error is reported. PLayback will halt after receiving a message of this kind.	
Info	Informational message, the payload can be inferred by checking the message's code.	
Warning	A warning is reported. This might indicate trouble. Please refer to the message's code for details on how to interpret the message.	

Class ClearVRPlayer

The ClearVRPlayer class is the primary entry point, also refer to ClearVRPlayer for a detailed description.

Inheritance

System.Object ClearVRPlayer

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRPlayer : MonoBehaviour

Fields

applicationRegainedFocusDelegate

Normally, you would leave the ClearVRPlayer in full control over what happens during an application suspend/resume cycle (e.g. when your app is backgrounded and brought back to the foreground or a headset was put down and picked up again). This is configured by setting applicationFocusAndPauseHandling to Recommended. The ClearVRLoadPreviousStateDelegate allows greater control for your application over this automatic behaviour. When set, this delegate will be triggered *just* prior to playback resume after your application has resumed. The ClearVRPlayer argument will be your ClearVRPlayer object, and the PlatformOptions argument will be equal to its platformOptions (as set by you during initial construction). In this moment, you can make any alteration to the PlatformOptions (like changing the ContentItem that should start playback). When implemented, you will be in charge to call ClearVRPlayer.Initialize() again once you are done modifying the PlatformOptions.

This API was known as clearVRLoadPreviousStateDelegate in versions prior to v9.1

Since v9.1: Added the argLastPlayedContentInfo argument, which contains the ContentInfo of the last played ContentItem. Note that this value *can be null*!

Example usage:

In this example we override the default playback resume logic for LIVE content after an application lost focus/regained focus (was suspended and resumed) cycle. By default, playback would resume from the last known position. With the demonstrated logic, playback will resume from the live edge instead.

```
// When setting up your platformOptions
clearVRPlayer.applicationRegainedFocusDelegate = new
ClearVRPlayer.ClearVRApplicationRegainedFocusDelegate(ApplicationRegainedFocus);
public void ApplicationRegainedFocus(ClearVRPlayer argClearVRPlayer, PlatformOptionsBase
argPlatformOptions, ContentInfo argLastPlayedContentInfo) {
  // This delegate is triggered on the main unity thread. Do not perform blocking code here as it will freeze your
app.
  // You cannot interact with the ClearVRPlayer in any way while it is in this state. Be sure to block any (looping)
code from accessing the ClearVRPlayer object.
  if(argLastPlayedContentInfo != null && argLastPlayedContentInfo.eventType == EventTypes.Live) {
     argPlatformOptions.prepareContentParameters.timingParameters = null; // null == start from live edge for
live content, from the beginning for VOD.
  argClearVRPlayer.Initialize(argPlatformOptions,
     onSuccess: (cbClearVREvent, cbClearVRPlayer) => {
       // Handle success like you would handle failure when you initialized the player
    },
    onFailure: (cbClearVREvent, cbClearVRPlayer) => {
       // Handle failure like you would handle failure when you initialized the player
    }
  }));
}
```

NOTE

This delegate will only be triggered when ApplicationFocusAndPauseHandlingType is set to Recommended.

O NOTE

If the application is quickly suspended before playback has even started (e.g. during player initialization), your original Initialize() callback will NOT be triggered. Instead, your new Initialize() callback will be triggered upon completion. The same holds for any subsequent suspend/resume cycle before content load completed.

O NOTE

The callback will only be triggered once until you call Initialize() again. If the application is suspened before Initialize() is called, you will not be notified again.

A WARNING

This delegate is triggered on the main thread, so any blocking operations (like querying your CMS) should be deferred to another thread. Remember that ClearVRPlayer.Initialize() *must* be called from the main thread as well. Behaviour is undefined when called from any another thread.

A WARNING

All ClearVRPlayer interfaces (like controller and mediaPlayer) are null when this delegate is triggered.

Declaration

public ClearVRPlayer.ClearVRApplicationRegainedFocus applicationRegainedFocusDelegate

Field Value

TYPE	DESCRIPTION
ClearVRPlayer.ClearVRApplicationRegainedFocus	

applicationUnpausedDelegate

When platformOptions.ApplicationFocusAndPauseHandlingType is set to Recommended, the player will unpause after the application was unpaused by the OS (after the application first got paused by the OS, e.g. because the notification drawer was pulled down on a mobile device). By default, both VOD and LIVE playback will continue from the last known position. Additionally, for LIVE content only, playback will resume from the live edge if the last played video segment fell out of the live window of the live stream by the time the application unpaused. To override this default behaviour, the application can subscribe to this callback.

When you subscribe to this delegate, it is the application's responsibility to take action. Otherwise, playback will be paused indefinitely. Please refer to the Unpause(TimingParameters) API for details.

Example code:

In this example we demonstrate how one makes sure that playback will jump to Ithe live edge after the *application* as paused and unpaused by the OS.

```
// When setting up your platformOptions
clearVRPlayer.applicationUnpausedDelegate = new
ClearVRPlayer.ClearVRApplicationUnpausedDelegate(ApplicationUnpaused);
public void ApplicationUnpaused(ClearVRPlayer argClearVRPlayer, PlatformOptionsBase argPlatformOptions,
ContentInfo argLastPlayedContentInfo, bool argWasPlaybackPausedByUser) {
 if(!argWasPlaybackPausedByUser) {
  TimingParameters tp = null; // We want default unpause behaviour, except for live content.
  if(argLastPlayedContentInfo != null && argLastPlayedContentType == ContentTypes.Live) {
   tp = new TimingParameters(0, TimingTypes.LiveEdge);
  if(argClearVRPlayer.controller != null) {
   argClearVRPlayer.controller.Unpause(tp,
    onSuccess: (cbClearVREvent, cbClearVRPlayer) =>
      UnityEngine.Debug.Log("[ClearVR] Player UNPAUSED CUSTOM."),
    onFailure: (cbClearVREvent, cbClearVRPlayer) =>
      UnityEngine.Debug.LogWarning(String.Format("[ClearVR] Something went wrong while unpausing the
ClearVRPlayer after application was unpaused. Error Code: {0}; Message: {1} .",
cbClearVREvent.message.code, cbClearVREvent.message.message))
   );
  \} // else: the player is shutting down, perhaps application pause crossed with the application losing focus.
Nothing we can and should do here.
} // else: user explicitly paused playback before the application was paused, let's not unpause for him.
}
```

6 NOTE

This delegate will only be triggered when ApplicationFocusAndPauseHandlingType is set to Recommended.

▲ WARNING

This delegate is triggered on the main thread, so any blocking operations should be deferred to another thread.

Declaration

public ClearVRPlayer.ClearVRApplicationUnpausedDelegate applicationUnpausedDelegate

Field Value

TYPE	DESCRIPTION
ClearVRPlayer.ClearVRApplicationUnpausedDelegate	

clearVRLoadPreviousStateDelegate

This API has been renamed in v9.1 to applicationRegainedFocusDelegate.

public ClearVRPlayer.ClearVRLoadPreviousStateDelegate clearVRLoadPreviousStateDelegate

Field Value

TYPE	DESCRIPTION
ClearVRPlayer.ClearVRLoadPreviousStateDelegate	

loggingConfig

Configuration of the verbosity level and output configuration of the logging of the Tiledmedia SDK

6 NOTE

The value of this field is fixed from the moment you call clearVRPlayer.Initialize() OR ClearVRPlayer.TestIsContentSupported(), whichever is called first. Changing the value of this field afterwards has no effect.

A WARNING

Take special care in making sure that this value is always the default in release builds as it has a serious negative performance impact.

Declaration

public static LoggingConfiguration loggingConfig

Field Value

TYPE	DESCRIPTION
LoggingConfiguration	

Properties

clearVRDisplayObjectEvents

Attach a listener to this event channel to be notified of any changes to any Display Object.

Declaration

public ClearVRDisplayObjectEvents clearVRDisplayObjectEvents { get; }

Property Value

ТҮРЕ	DESCRIPTION
ClearVRDisplayObjectEvents	

clearVREvents

Used to subscribe to ClearVREvent events.

Declaration

public ClearVREvents clearVREvents { get; }

Property Value

ТҮРЕ	DESCRIPTION
ClearVREvents	

controller

This interface can be used to control the media player.

Declaration

public MediaControllerInterface controller { get; }

Property Value

TYPE	DESCRIPTION
MediaControllerInterface	The controller interface

coreLogFile

Instead of writing to stdout, the core ClearVRCore log will be written to the specified file. Default value: "" (e.g. do not log to disk) Refer to ClearVRPlayer.coreLogLevel to configure the core log verbosity.

O NOTE

The value of this field is fixed from the moment you call clearVRPlayer.Initialize() OR ClearVRPlayer.TestIsContentSupported(), whichever is called first. Changing the value of this field afterwards has no effect.

Declaration

public static String coreLogFile { get; set; }

Property Value

TYPE	DESCRIPTION
String	

coreLogLevel

ClearVRCore verbosity level. Please keep at the default value 0 at all times. Valid value: 0, 1, 2. Setting it to any other value will have a negative performance impact. Refer to ClearVRPlayer.coreLogFile to (optionally) write the core log file to disk.

O NOTE

The value of this field is fixed from the moment you call clearVRPlayer.Initialize() OR ClearVRPlayer.TestIsContentSupported(), whichever is called first. Changing the value of this field afterwards has no effect.

A WARNING

Take special care in making sure that this value is always 0 in release builds as it has a serious negative performance impact.

Declaration

public static int coreLogLevel { get; set; }

Property Value

TYPE	DESCRIPTION
System.Int32	

debug

This interface can be used to access debug-related APIs.

A WARNING

Do not use any API on this interface. They are subject to change without notice not backwards compatibility.

Declaration

public DebugInterface debug { get; }

Property Value

TYPE	DESCRIPTION
DebugInterface	The debug interface

mediaflowLogFile

Instead of writing to stdout, the core ClearVRCore log will be written to the specified file. Default value: "" (e.g. do not log to disk) Refer to ClearVRPlayer.coreLogLevel to configure the core log verbosity.

6 NOTE

The value of this field is fixed from the moment you call clearVRPlayer.Initialize() OR ClearVRPlayer.TestIsContentSupported(), whichever is called first. Changing the value of this field afterwards has no effect.

Declaration

public static String mediaflowLogFile { get; set; }

Property Value

TYPE	DESCRIPTION
String	

mediaflowLogLevel

ClearVRMediaFlow verbosity level. Please keep at the default value 0 at all times. Valid value: 0, 1, 2, 3, 4, 5. Setting it to any other value will have a negative performance impact. Refer to ClearVRPlayer.mediaflowLogFile to (optionally) write the mediaflow log file to disk.

6 NOTE

The value of this field is fixed from the moment you call clearVRPlayer.Initialize() OR ClearVRPlayer.TestIsContentSupported(), whichever is called first. Changing the value of this field afterwards has no effect.

A WARNING

Take special care in making sure that this value is always 0 in release builds as it has a serious negative performance impact.

Declaration

public static int mediaflowLogLevel { get; set; }

Property Value

TYPE	DESCRIPTION
System.Int32	

mediaInfo

This interface can be used to access media related information (like content duration and format)

Declaration

public MediaInfoInterface mediaInfo { get; }

Property Value

TYPE	DESCRIPTION
MediaInfoInterface	The mediainfo interface

mediaPlayer

This interface can be used to access mediaplayer interface.

Declaration

public MediaPlayerInterface mediaPlayer { get; }

Property Value

TYPE	DESCRIPTION
MediaPlayerInterface	The mediaplayer interface

performance

This interface can be used to access performance-related metrics.

Declaration

public PerformanceInterface performance { get; }

Property Value

TYPE	DESCRIPTION
PerformanceInterface	The performance interface

platformOptions

Getter for the currently configured platform options.

A WARNING

One must never change any value on the platformOptions after the ClearVRPlayer object has been initialized. Doing so will result in undefined behaviour.

Declaration

public PlatformOptionsBase platformOptions { get; }

Property Value

TYPE	DESCRIPTION
PlatformOptionsBase	

renderMode

Change the RenderMode of the main ClearVRDisplayObject.

Declaration

public RenderModes renderMode { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
RenderModes	

sync

This interface can be used to control the livestream sync feature.

Declaration

public SyncInterface sync { get; }

Property Value

TYPE	DESCRIPTION
SyncInterface	The sync interface

Methods

AddOrUpdateLayoutParameters(LayoutParameters)

Adds a new, or updates an existing, LayoutParameters set on the LayoutManager. Remember that the name of the LayoutParameters functions as a unique identifier.

Declaration

public bool AddOrUpdateLayoutParameters(LayoutParameters argLayoutParameters)

Parameters

TYPE	NAME	DESCRIPTION
LayoutParameters	argLayoutParameters	The LayoutParameters to add or update on the LayoutManager.

Returns

TYPE	DESCRIPTION
System.Boolean	True in case the LayoutParameters are correct and added or updated on the LayoutManager or if the argument is null. False will be returned in any other case.

CallCoreStatic(String, Action<String, Object[]>, Action<ClearVRMessage, Object[]>, Object[])

Sends a message to the core.

Declaration

public static void CallCoreStatic(String base64Message, Action<String, object[]> onSuccess, Action<ClearVRMessage, object[]> onFailure, params object[] optionalArguments)

TYPE	NAME	DESCRIPTION	
String	base64Message	The base64Message with instructions.	
Action <string, system.object[]=""></string,>	onSuccess	This is triggered when the CallCore call was succesful. Contains a base64 message. Cannot be null.	
Action <clearvrmessage, system.object[]=""></clearvrmessage,>	onFailure	This is triggered when the CallCore call was unsuccesful and inclides a ClearVRMessage object with the reason of this failiure.	
System.Object[]	optionalArguments	Any optional arguments you'd like to add.	

DestroyGameComponentAndCleanUp()

Declaration

public void DestroyGameComponentAndCleanUp()

EnableLogging(LoggingConfiguration)

Enable extra logging on the SDK To disable the extra logging you can pass null as an argument.

Declaration

public static void EnableLogging(LoggingConfiguration loggingConfiguration)

Parameters

TYPE	NAME	DESCRIPTION
LoggingConfiguration	loggingConfiguration	the configuration to change the path, log level etc. see LoggingConfiguration for more information.

EnableLoggingConfiguration(LoggingConfiguration)

Declaration

public static void EnableLoggingConfiguration(LoggingConfiguration argLoggingConfiguration)

TYPE	NAME	DESCRIPTION
LoggingConfiguration	argLoggingConfiguration	

GetClearVRCoreVersion()

Returns the SDK version or "Unknown" if unknown.

Declaration

public static String GetClearVRCoreVersion()

Returns

TYPE	DESCRIPTION
String	The SDK version as a String

GetDefaultPlatformOptions()

Returns the default PlatformOptionsBase for the current platform.

Declaration

public PlatformOptionsBase GetDefaultPlatformOptions()

Returns

TYPE	DESCRIPTION
PlatformOptionsBase	

GetIsHardwareHEVCDecoderAvailable()

Check whether an hardware HEVC video decoder is present or not. Having a hardware HEVC decoder is a prerequisit for being able to play Mosaic and ClearVR content.

Declaration

public static bool GetIsHardwareHEVCDecoderAvailable()

ТҮРЕ	DESCRIPTION		
------	-------------	--	--

TYPE	DESCRIPTION
System.Boolean	True if a hardware HEVC decoder is available, false otherwise.

GetIsPlatformSupported()

Query whether ClearVR supports the current active Application.platform

Declaration

public static bool GetIsPlatformSupported()

Returns

ТҮРЕ	DESCRIPTION
System.Boolean	true if the platform is supported, false otherwise

GetIsPlatformSupported(RuntimePlatform)

Query whether ClearVR supports a specific RuntimePlatform.

Declaration

public static bool GetIsPlatformSupported(RuntimePlatform argPlatform)

Parameters

TYPE	NAME	DESCRIPTION
RuntimePlatform	argPlatform	The platform to check for.

Returns

TYPE	DESCRIPTION
System.Boolean	

GetLayoutParametersByName(String)

Fetch a COPY of the LayoutParameters with the specified name. This API can return null in case no such Layout is found.

A WARNING

Note that this API returns a COPY of the LayoutParameters as specified on the LayoutManager. Any changes you make to it will not be reflected on the LayoutManager *until* you call

clearVRPlayer.AddOrUpdateLayoutParameters() OR you pass your LayoutParameters into

PrepareContentParameters or SwitchContentParameters.

Declaration

public LayoutParameters GetLayoutParametersByName(String argName)

Parameters

TYPE	NAME	DESCRIPTION
String	argName	The name of the Layout to fetch

Returns

TYPE	DESCRIPTION
LayoutParameters	A COPY of the LayoutParameters as specified on the LayoutManager, or null if no such named Layout is configured on the ClearVRLayoutManager

GetProxyParameters(ProxyParameters)

Declaration

public static ProxyParameters GetProxyParameters (ProxyParameters proxyParameters)

Parameters

TYPE	NAME	DESCRIPTION
ProxyParameters	proxyParameters	

Returns

TYPE	DESCRIPTION
ProxyParameters	

Initialize(PlatformOptionsBase)

Create and initialize the ClearVR player. Depending on the platformOptions, content will be loaded automatically and playback could start as soon as soon as content loading completed.



This API must be called from the main thread.

Declaration

public ClearVRAsyncRequest Initialize(PlatformOptionsBase argPlatformOptions)

Parameters

TYPE	NAME	DESCRIPTION
PlatformOptionsBase	argPlatformOptions	the platform specific (player) options as set by the parent.

Returns

TYPE	DESCRIPTION
ClearVRAsyncRequest	

Initialize(PlatformOptionsBase, Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Create and initialize the ClearVR player. Depending on the platformOptions, content will be loaded automatically and playback could start as soon as content loading completed.



This API must be called from the main thread.

Declaration

public void Initialize(PlatformOptionsBase argPlatformOptions, Action<ClearVREvent, ClearVRPlayer>
onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] argOptionalArguments)

TYPE	NAME	DESCRIPTION
------	------	-------------

ТҮРЕ	NAME	DESCRIPTION
PlatformOptionsBase	argPlatformOptions	the platform specific (player) options as set by the parent.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onSuccess	An optional callback that should be triggered after the request was successfully completed. You are highly encouragedto implement the callback, but it can be null.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	An optional callback that should be triggered in case the request has failed. You are highly encouragedto implement the callback, but it can be null.
System.Object[]	argOptionalArguments	Any optional argument one might want to pass into the callback.

Initialize(PlatformOptionsBase, Action<ClearVREvent, ClearVRPlayer>, Object[])

Declaration

public void Initialize(PlatformOptionsBase argPlatformOptions, Action<ClearVREvent, ClearVRPlayer>
argCbClearVRAsyncRequestResponseReceived, params object[] argOptionalArguments)

Parameters

ТҮРЕ	NAME	DESCRIPTION
PlatformOptionsBase	argPlatformOptions	
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	argCbClearVRAsyncRequestResponseReceived	
System.Object[]	argOptionalArguments	

OnApplicationQuit()

Declaration

public void OnApplicationQuit()

RemoveLayoutParameters(LayoutParameters)

Remove

Declaration

public bool RemoveLayoutParameters(LayoutParameters argLayoutParameters)

Parameters

TYPE	NAME	DESCRIPTION
LayoutParameters	argLayoutParameters	The LayoutParameters to remove

Returns

TYPE	DESCRIPTION
System.Boolean	True in case of success, false otherwise. If the argument is null, true will be returned.

TestIsContentSupported(ContentSupportedTesterParameters, Action<ClearVRMessage, ContentSupportedTesterReport>, Object[])

Test whether the list of ContentItems can be played on the current device or not. In the provided Action callback, query argClearVRMessage.GetIsSuccess() to determine whether the content check was successful or not. If the check was not successful, ContentSupportedTesterReport will be null. If successful, the ContentItemList will have its contentSupportedStatus field set to the appropriate value.

O NOTE

Since v7.4 the callback is guaranteed to be triggered on the main Unity thread IF this API is called on the main Unity thread. The callback will be triggered on a random thread if this API is not called on the main Unity thread.

Declaration

public static void TestIsContentSupported(ContentSupportedTesterParameters
argContentSupportedTesterParameters, Action<ClearVRMessage, ContentSupportedTesterReport>
argCbReportReceived, params object[] argOptionalArguments)

TYPE NAME DESCRIPTION	TYPE	NAME	DESCRIPTION
-----------------------	------	------	-------------

TYPE	NAME	DESCRIPTION
ContentSupportedTesterParameters	argContentSupportedTesterParameters	The parameters describing the test.
Action <clearvrmessage, contentsupportedtesterreport=""></clearvrmessage,>	argCbReportReceived	The callback to trigger upon success. Cannot be null.
System.Object[]	argOptionalArguments	Any optional arguments youw ant to pass along and receive in the callback.

TestIsContentSupported(ContentSupportedTesterParameters, Action<ContentSupportedTesterReport, Object[]>, Action<ClearVRMessage, Object[]>, Object[])

Test whether the list of ContentItems can be played on the current device or not. In the provided Action callback, query argClearVRMessage.GetIsSuccess() to determine whether the content check was successful or not. If the check was not successfull, ContentSupportedTesterReport will be null. If successful, the ContentItemList will have its contentSupportedStatus field set to the appropriate value.

1 NOTE

Since v7.4 the callback is guaranteed to be triggered on the main Unity thread IF this API is called on the main Unity thread. The callback will be triggered on a random thread if this API is not called on the main Unity thread.

Declaration

public static void TestIsContentSupported(ContentSupportedTesterParameters
contentSupportedTesterParameters, Action<ContentSupportedTesterReport, object[]> onSuccess,
Action<ClearVRMessage, object[]> onFailure, params object[] optionalArguments)

TYPE	NAME	DESCRIPTION
ContentSupportedTesterParameters	contentSupportedTesterParameters	The parameters describing the test.

ТҮРЕ	NAME	DESCRIPTION
Action <contentsupportedtesterreport, system.object[]=""></contentsupportedtesterreport,>	onSuccess	Callback to be triggered when the test is successful. Cannot be null
Action <clearvrmessage, system.object[]=""></clearvrmessage,>	onFailure	Callback to be triggered when the test is not successful.
System.Object[]	optionalArguments	Any optional arguments youw ant to pass along and receive in the callback.

UploadLogs(Byte[], Action<String, Object[]>, Action<ClearVRMessage, Object[]>, Object[])

This API is used to easily upload Tiledmedia player-generated logfiles to the Tiledmedia backend. This API should ONLY be used while debugging an issue that could be related to the Tiledmedia Player SDK, and should typically be disabled/not called in release builds of your product.

A WARNING

You are advised to only use this API while debugging your application. When debugging such an issue, you would typically follow these steps: 1) Enable Tiledmedia Player logging 2) When the problem has happened (just after, or after you have restarted the app in case of hard crash for example), you upload the logs.

See the following code snippet:

```
LoggingConfiguration loggingConfiguration = LoggingConfiguration.GetDefaultLoggingConfiguration();
ClearVRPlayer.EnableLogging(loggingConfiguration);
...
ClearVRPlayer.UploadLogs(yourLicenseFile,
    onSuccess: (uniqueUploadID, optionalArguments) => {
        UnityEngine.Debug.Log("Tiledmedia log upload success. Unique upload id: " + uniqueUploadID);
    },
    onFailure: (clearVRMessage, optionalArguments) => {
        UnityEngine.Debug.Log("Tiledmedia log upload failed. Error details: " + clearVRMessage);
    }
);
ClearVRPlayer.EnableLogging(loggingConfiguration);
```

Declaration

public static void UploadLogs(byte[] licenseFileBytes, Action<String, object[]> onSuccess,
Action<ClearVRMessage, object[]> onFailure, params object[] optionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
System.Byte[]	licenseFileBytes	Your Tiledmedia Player license file
Action <string, system.object[]=""></string,>	onSuccess	The success callback. The String argument contains a unique upload ID.
Action <clearvrmessage, system.object[]=""></clearvrmessage,>	onFailure	The failFure callback, triggered if something unexpectedly went wrong.
System.Object[]	optionalArguments	Any optional argument you would like to pass into the onSuccess and onFailure callbacks.

UploadLogs(Byte[], String, Action<String, Object[]>, Action<ClearVRMessage, Object[]>, Object[])

This API is used to easily upload Tiledmedia player-generated logfiles to the Tiledmedia backend. This API should ONLY be used while debugging an issue that could be related to the Tiledmedia Player SDK, and should typically be disabled/not called in release builds of your product.

A WARNING

You are advised to only use this API while debugging your application. When debugging such an issue, you would typically follow these steps: 1) Enable Tiledmedia Player logging 2) When the problem has happened (just after, or after you have restarted the app in case of hard crash for example), you upload the logs.

See the following code snippet:

```
LoggingConfiguration loggingConfiguration = LoggingConfiguration.GetDefaultLoggingConfiguration();
ClearVRPlayer.EnableLogging(loggingConfiguration);
...
ClearVRPlayer.UploadLogs(yourLicenseFile,
    onSuccess: (uniqueUploadID, optionalArguments) => {
        UnityEngine.Debug.Log("Tiledmedia log upload success. Unique upload id: " + uniqueUploadID);
    },
    onFailure: (clearVRMessage, optionalArguments) => {
        UnityEngine.Debug.Log("Tiledmedia log upload failed. Error details: " + clearVRMessage);
    }
);
ClearVRPlayer.EnableLogging(loggingConfiguration);
```

Declaration

public static void UploadLogs(byte[] licenseFileBytes, String logFolder, Action<String, object[]> onSuccess,
Action<ClearVRMessage, object[]> onFailure, params object[] optionalArguments)

ТҮРЕ	NAME	DESCRIPTION
System.Byte[]	licenseFileBytes	Your Tiledmedia Player license file
String	logFolder	The folder that contains the log fiels to upload. When you are not customizing the logging folder output, this value can be set to LoggingConfiguration.GetDefaultLoggingFolder()
Action <string, system.object[]=""></string,>	onSuccess	The success callback. The String argument contains a unique upload ID.
Action <clearvrmessage, system.object[]=""></clearvrmessage,>	onFailure	The failure callback, triggered if something unexpectedly went wrong.
System.Object[]	optionalArguments	Any optional argument you would like to pass into the onSuccess and onFailure callbacks.

Delegate ClearVRPlayer.ClearVRApplicationRegainedFo cus

The Delegate that needs to be implemented for the loading of the previous state callback.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public delegate void ClearVRApplicationRegainedFocus(ClearVRPlayer argClearVRPlayer, PlatformOptionsBase argPlatformOptions, ContentInfo argLastPlayedContentInfo);

TYPE	NAME	DESCRIPTION
ClearVRPlayer	argClearVRPlayer	The ClearVRPlayer that triggered the callback.
PlatformOptionsBase	argPlatformOptions	The PlatformOptions of the ClearVRPlayer that are saved in the core.
ContentInfo	argLastPlayedContentInfo	Since v9.1: Contains the ContentInfo of the last played ContentItem. Note that this value can be null!

Delegate ClearVRPlayer.ClearVRApplicationUnpausedD elegate

A delegate of this signature will be triggered when the application is unpaused after it got paused by the OS. Refer to clearVRApplicationUnpausedDelegate for details and how to subscribe to a delegate of this signature.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public delegate void ClearVRApplicationUnpausedDelegate(ClearVRPlayer argClearVRPlayer, PlatformOptionsBase argPlatformOptions, ContentInfo argLastPlayedContentInfo, bool wasPlaybackPausedBeforeApplicationPaused);

TYPE	NAME	DESCRIPTION
ClearVRPlayer	argClearVRPlayer	The ClearVRPlayer that is about to unpause.
PlatformOptionsBase	argPlatformOptions	The platform Options describing the configuration of the ClearVRPlayer. Making changes to this argument is not supported and will result in undefined behaviour.
ContentInfo	argLastPlayedContentInfo	Contains the ContentInfo of the last played ContentItem.
System.Boolean	wasPlaybackPausedBeforeApplicationPaused	True if playback was already paused (e.g. by the user) before the application was paused, false if playback was <i>not</i> paused before the application was paused.

Delegate ClearVRPlayer.ClearVRLoadPreviousStateDele gate

The Delegate that needs to be implemented for the loading of the previous state callback. This delegate has been renamed in v9.1 to [ClearVRApplicationRegainedFocus] (xref:com.tiledmedia.clearvr.ClearVRApplicationRegainedFocus(com.tiledmedia.clearvr.ClearVRPlay er.ClearVRApplicationRegainedFocus), which takes ContentInfo as its third argument.

Namespace: com.tiledmedia.clearvr

Assembly: Assembly-CSharp.dll

Syntax

public delegate void ClearVRLoadPreviousStateDelegate(ClearVRPlayer argClearVRPlayer, PlatformOptionsBase argPlatformOptions);

TYPE	NAME	DESCRIPTION
ClearVRPlayer	argClearVRPlayer	The ClearVRPlayer that triggered the callback.
PlatformOptionsBase	argPlatformOptions	The PlatformOptions of the ClearVRPlayer that are saved in the core.

Class ClearVRPose

This class has been removed in v9.0 and can no longer be used.

Inheritance

System.Object

ClearVRPose

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRPose: object

Fields

posX

Position X coordinate.

Declaration

public double posX

Field Value

TYPE	DESCRIPTION
System.Double	

posY

Position Y coordinate.

Declaration

public double posY

Field Value

TYPE	DESCRIPTION
System.Double	

posZ

Position Z coordinate.

Declaration

public double posZ

Field Value

TYPE	DESCRIPTION
System.Double	

W

Orientation w quaternion component.

Declaration

public double w

Field Value

TYPE	DESCRIPTION
System.Double	

X

Orientation x quaternion component.

Declaration

public double x

Field Value

TYPE	DESCRIPTION
System.Double	

У

Orientation y quaternion component.

Declaration

public double y

Field Value

TYPE	DESCRIPTION
System.Double	

Z

Orientation z quaternion component.

Declaration

public double z

Field Value

TYPE	DESCRIPTION
System.Double	

Class ClearVRScale

Object describing the scale of an object.

Inheritance

System.Object

ClearVRScale

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRScale: object

Constructors

ClearVRScale()

Default constructor

Declaration

public ClearVRScale()

ClearVRScale(Double, Double, Double)

Constructor taking three arguments

Declaration

public ClearVRScale(double argX, double argY, double argZ)

Parameters

TYPE	NAME	DESCRIPTION
System.Double	argX	The x scale
System.Double	argY	The y scale
System.Double	argZ	The z scale

Fields

X

Scale in the X local coordinate system of the object.

Declaration

public double x

Field Value

TYPE	DESCRIPTION
System.Double	

У

Scale in the Y local coordinate system of the object.

Declaration

public double y

Field Value

TYPE	DESCRIPTION
System.Double	

Z

Scale in the Z local coordinate system of the object.

Declaration

public double z

Field Value

TYPE	DESCRIPTION
System.Double	

Methods

Copy()

Creates a deep-copy of this object

Declaration

public ClearVRScale Copy()

(c) Tiledmedia B.V. 2017-2022

Returns

TYPE	DESCRIPTION
ClearVRScale	A deep-copy

Equals(Object)

Declaration

public override bool Equals(object argOther)

Parameters

TYPE	NAME	DESCRIPTION
System.Object	argOther	

Returns

TYPE	DESCRIPTION
System.Boolean	

GetHashCode()

Declaration

public override int GetHashCode()

Returns

ТҮРЕ	DESCRIPTION
System.Int32	

ScaleToVector3()

Converts this object into a Unity Vector3 object

Declaration

public Vector3 ScaleToVector3()

ТҮРЕ	DESCRIPTION
Vector3	The Vector3 equivalent

ToString()

Convenience method that returns an informational string listing all fields and their values.

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	A string

Class ClearVRSubtitle

Helper class that holds the info of one subtitle.

Inheritance

System.Object

ClearVRSubtitle

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRSubtitle: object

Methods

GetFeedIndex()

Declaration

public Int32 GetFeedIndex()

Returns

TYPE	DESCRIPTION
Int32	

GetText()

Declaration

public String GetText()

TYPE	DESCRIPTION
String	

Class ClearVRViewportAndObjectPose

This class has been removed in v9.0 and can no longer be used.

Inheritance

System.Object

ClearVRViewportAndObjectPose

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ClearVRViewportAndObjectPose : object

Fields

displayObject

Declaration

public ClearVRDisplayObject displayObject

Field Value

TYPE	DESCRIPTION
ClearVRDisplayObject	

viewportPose

Declaration

public ClearVRPose viewportPose

Field Value

TYPE	DESCRIPTION
ClearVRPose	

Enum ColorSpaces

Supported color spaces. Currently, only Gamma is supported.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ColorSpaces : int

Fields

NAME	DESCRIPTION
Gamma	Regular Gamma.
Linear	Linear (not supported).
Uninitialized	Not yet specified and/or auto-detected.

Enum ColorSpaceStandards

ColorSpace is used to indicate the color space used to generate the YUV coordinates. It is used to convert the color back to RGBA. ColorSpace.Unspecified may default to BT.709

Namespace: com.tiledmedia.clearv	1
Assembly: Assembly-CSharp.dll	

Syntax

public enum ColorSpaceStandards : int	
---------------------------------------	--

Fields

NAME	DESCRIPTION
BT2020_CL	
BT2020_NCL	
BT601	
BT709	
Unspecified	

Enum ContentFormat

An enum describing the format of the currently playing content.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ContentFormat : int

Fields

NAME	DESCRIPTION
Monoscopic180	Monosopic, 180 degree ClearVR content.
Monoscopic360	Monosopic, 360 degree ClearVR content.
MonoscopicERP180	Monosopic, 180 degree tranditional equirectangular content.
MonoscopicERP360	Monosopic, 360 degree tranditional equirectangular content.
MonoscopicFishEye	Monosopic, fishEye content.
MonoscopicFishEyeEquidistant180	Monosopic, 180 degree equidistant fishEye content.
MonoscopicFishEyeEquiSolid180	Monosopic, 180 degree equisolid fishEye content.
MonoscopicOmnidirectional	MOnoscopic ClearVR content.
MonoscopicRectilinear	Monosopic, traditional broadcast content.

NAME	DESCRIPTION
Planar	Monosopic, ultra-wide and ultra-high resolution "flat" content.
Stereoscopic180	Stereoscopic, 180 degree ClearVR content.
Stereoscopic360	Stereoscopic, 360 degree ClearVR content.
StereoscopicERP180SBS	Stereoscopic side-by-side, 180 degree tranditional equirectangular content.
StereoscopicERP360TB	Stereoscopic top-bottom, 360 degree tranditional equirectangular content.
StereoscopicFishEyeEquidistant180SBS	Stereoscopic side-by-side, 180 degree equidistant fishEye content.
StereoscopicFishEyeEquiSolid180SBS	Stereoscopic side-by-side, 180 degree equisolid fishEye content.
StereoscopicFishEyeSBS	Stereoscopic side-by-side, fishEye content.
StereoscopicOmnidirectional	Stereoscopic ClearVR content.
StereoscopicRectilinearSBS	Stereoscopic sibe-by-side, traditional broadcast content.
StereoscopicRectilinearTB	Stereoscopic top-bottom, traditional broadcast content.
Unknown	The format is unknown and not supported.

Class Contentinfo

Helper class that holds the info of one content item. For detailed information about the ContentInfo, FeedInfo, VideoTrackInfo, AudioTrackInfo, SubtitleTrackInfo objects and their relationship, refer to the ContentItem documentation.

Inheritance

System.Object

ContentInfo

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ContentInfo: object

Fields

feeds

Array containing all feeds that this content item contains.

Declaration

public FeedInfo∏ feeds

Field Value

TYPE	DESCRIPTION
FeedInfo[]	

Properties

eventType

Declaration

public EventTypes eventType { get; }

Property Value

ТҮРЕ	DESCRIPTION
EventTypes	

Methods

GetActiveAudioTrackID()

Get the TrackID that identifies the currently active audio track and the FeedInfo it can be found on. This API can used in conjunction with the SetLayout API,

O NOTE

Also refer to the GetActiveAudioTrack() API.

Declaration

public TrackID GetActiveAudioTrackID()

Returns

TYPE	DESCRIPTION
TrackID	The TrackID that uniquely identifies the active audio track, or null if no audio track is active.

GetActiveSubtitleTrack()

Get the SubtitleTrackInfo of the currently active subtitle track..

Declaration

public SubtitleTrackInfo GetActiveSubtitleTrack()

Returns

TYPE	DESCRIPTION
SubtitleTrackInfo	The SubtitleTrackInfo of the active subtitle track, or null if no subtitle track is active.

GetActiveSubtitleTrackID()

Get the TrackID that identifies the currently active subtitle track and the FeedInfo it can be found on. This API can used in conjunction with the SetLayout API,

6 NOTE

Also refer to the GetActiveSubtitleTrack() API.

Declaration

public TrackID GetActiveSubtitleTrackID()

Returns

TYPE	DESCRIPTION
TrackID	The TrackID that uniquely identifies the active subtitle track, or null if no subtitle track is active.

GetFeedInfoByDisplayObjectController(ClearVRDisplayObjectControllerBase)

Each ClearVR DisplayObject shows one Feed. Information about this Feed is stored in the FeedInfo object. This API returns the FeedInfo that matches the feed index currently associated to the provided ClearVRDisplayObjectControllerBase.

Declaration

public FeedInfo GetFeedInfoByDisplayObjectController(ClearVRDisplayObjectControllerBase argClearVRDisplayObjectController)

Parameters

ТҮРЕ	NAME	DESCRIPTION
ClearVRDisplayObjectControllerBase	argClearVRDisplayObjectController	The DisplayObjectController of interest. Should not be null.

Returns

TYPE	DESCRIPTION
FeedInfo	The FeedInfo, or null if not available or when the argClearVRDisplayObjectController is null

GetFeedsWithActiveVideoTrack()

This convenience API returns a list of all FeedInfo that have at least one active VideoTrackInfo. Note that the active AudioTrackInfo can be located on a FeedInfo that is different than any of the active VideoTrackInfo.

Declaration

public FeedInfo[] GetFeedsWithActiveVideoTrack()

Returns

TYPE	DESCRIPTION
FeedInfo[]	The array can have size 0 (in theory) but can never be null.

GetFeedWithActiveAudioTrack()

This convenience API returns the FeedInfo of the Feed that has the active audio track. Notes:

- There can be 0 or at most 1 active audio track at any point in time.
- The Feed with the active audio track can be different to the Feed that has the active video track(s).

Declaration

public FeedInfo GetFeedWithActiveAudioTrack()

Returns

TYPE	DESCRIPTION
FeedInfo	The AudioTrackInfo of the currently active audio track, or null if no audio track is active.

GetFeedWithActiveSubtitleTrack()

This convenience API returns the FeedInfo of the Feed that has the active subtitle track. Notes:

- There can be 0 or at most 1 active subtitle track at any point in time.
- The Feed with the active subtitle track can be different to the Feed that has the active video and/or audio track(s).

Declaration

public FeedInfo GetFeedWithActiveSubtitleTrack()

Returns

TYPE	DESCRIPTION
FeedInfo	The AudioTrackInfo of the currently active subtitle track, or null if no subtitle track is active.

GetNumberOfSelectableAudioTracks()

Returns the number of selectable audio tracks in the current player state. See

GetSelectableAudioTrackIDs for details.

Declaration

public int GetNumberOfSelectableAudioTracks()

Returns

TYPE	DESCRIPTION
System.Int32	Returns an integer value. 0 means: no audio tracks can be selected.

GetNumberOfSelectableSubtitlesTracks()

Returns the number of selectable subtitles tracks in the current player state. See GetSelectableSubtitlesTrackIDs for details.

Declaration

public int GetNumberOfSelectableSubtitlesTracks()

Returns

TYPE	DESCRIPTION
System.Int32	Returns an integer value. 0 means: no Subtitles tracks can be selected.

GetSelectableAudioTrackIDs()

Returns a list of TrackIDs of which at least one audio track could be selected in the current state. Please refer to GetSelectableAudioTrackIDs for the definition of a "selectable" audio track.



In most cases the GetSelectableAudioTracks() API is more convenient.

///

Declaration

public TrackID[] GetSelectableAudioTrackIDs()

Returns

TYPE	DESCRIPTION
TrackID[]	A list with a length of zero, one or more items. This can never be null.

GetSelectableAudioTracks()

Returns a list of AudioTrackInfo of which at least one audio track can be selected in the current state in any of the Feeds. Please refer to GetSelectableAudioTracks for the definition of a "selectable" audio track.

Declaration

public AudioTrackInfo[] GetSelectableAudioTracks()

Returns

TYPE	DESCRIPTION
AudioTrackInfo[]	A list with a length of zero, one or more items. This can never be null.

GetSelectableSubtitlesTrackIDs()

Returns a list of TrackIDs of which at least one subtitles track could be selected in the current state. Please refer to GetSelectableSubtitlesTrackIDs for details.

Declaration

public TrackID[] GetSelectableSubtitlesTrackIDs()

Returns

TYPE	DESCRIPTION
TrackID[]	A list with a length of zero, one or more items. This can never be null.

GetSelectableSubtitlesTracks()

Returns a list of SubtitleTrackInfo of which at least one subtitles track could be selected in the current state. Please refer to GetSelectableSubtitlesTracks for details.

Declaration

public SubtitleTrackInfo[] GetSelectableSubtitlesTracks()

Returns

TYPE	DESCRIPTION
SubtitleTrackInfo[]	A list with a length of zero, one or more items. This can never be null

ToString()

Declaration

public override String ToString()

Returns

TYPE	DESCRIPTION
String	

Class ContentItem

This object is used to load an initial clip and to switch to a new clip.

Inheritance

System.Object

ContentItem

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ContentItem: object

Constructors

ContentItem(String)

The default constructor for a ContentItem. One can set additional fields, likedrmInfo and fishEyeSettings by directly setting the appropriate fields.

Declaration

public ContentItem(String argManifestUri)

Parameters

TYPE	NAME	DESCRIPTION
String	argManifestUri	

ContentItem(String, ClearVRViewportAndObjectPose, ContentFormat, DRMInfo, FishEyeSettings)

Since v8 ContentItem constructor

The manifest url pointing to the clip that needs to be loaded.Start pose of the viewport and start pose and scale of the display object. One can use the ClearVRViewportAndObjectPose(double) constructor to set the default object. If set to null, the default will be used, which is position (0, 0, 0), orientation (w=1, x=0, y=0, z=0). If you set this to non-null AND if you change more than just the display object's scale, automatic mesh placement will be disabled. Typically, this field is only set to non-null to set a rotation offset for omnidirectional content. Override content format. Keep at its default value ContentFormat.Unknown unless needed otherwise. See overrideCOntentFormat for details. Specify DRM info, required to decrypt this content item.Configure camera and lens specific fish eye settings for ContentFormat.MonoscopicFishEye or

ContentFormat.StereoscopicFishEyeSBS, ignored otherwise. Default value: null

Declaration

public ContentItem(String argManifestUrl, ClearVRViewportAndObjectPose argStartViewportAndObjectPose = null, ContentFormat argOverrideContentFormat = default(ContentFormat), DRMInfo argDRMInfo = null, FishEyeSettings argFishEyeSettings = null)

Parameters

TYPE	NAME	DESCRIPTION
String	argManifestUrl	
ClearVRViewportAndObjectPose	argStartViewportAndObjectPose	
ContentFormat	argOverrideContentFormat	
DRMInfo	argDRMInfo	
FishEyeSettings	argFishEyeSettings	

ContentItem(String, ContentFormat, DRMInfo, FishEyeSettings)

ContentItem constructor

The manifest url pointing to the clip that needs to be loaded. Override content format. Keep at its default value ContentFormat. Unknown unless needed otherwise. See override COntentFormat for details. Specify DRM info, required to decrypt this content item. Configure camera and lens specific fish eye settings for ContentFormat. MonoscopicFishEye or

ContentFormat.StereoscopicFishEyeSBS, ignored otherwise. Default value: null



Since v9.x, one can no longer specify a custom initial mesh positon. Instead, use the LayoutManager.

Declaration

public ContentItem(String argManifestUrl, ContentFormat argOverrideContentFormat = default(ContentFormat),
DRMInfo argDRMInfo = null, FishEyeSettings argFishEyeSettings = null)

TYPE	NAME	DESCRIPTION
String	argManifestUrl	

ТҮРЕ	NAME	DESCRIPTION
ContentFormat	argOverrideContentFormat	
DRMInfo	argDRMInfo	
FishEyeSettings	argFishEyeSettings	

Fields

drmInfo

Since v7

Default value: null

Allows one to specify DRM-specific information for this clip. Please note the limitations of playing back DRM protected content.

Notes:

- Widevine L1 protected content can only be played on an OVROverlay. Requires platformOptions.contentProtectionRobustnessLevel = HWSecureAll
- 2. Widevine L3 protected or SampleAES encrypted content requires platformOptions.contentProtectionRobustnessLevel = SWSecureDecode
- 3. Setting this field to null equals to specifying that this particular clip is NOT DRM protected.
- 4. One can freely switch between protected and non-protected content.
- 5. Playback of Widevine protected content is only available on Android.

Declaration

public DRMInfo drmInfo

Field Value

ТҮРЕ	DESCRIPTION
DRMInfo	

manifestUrl

The manifest URL that should be loaded

Declaration

public string manifestUrl

Field Value

TYPE	DESCRIPTION
System.String	

overrideContentFormat

When playing non-ClearVR and non-Mosaic content (e.g. HLS or pMP4), you have to explicitly set the ContentFormat (aka projection type).

When set to the default value ContentFormat.Unknown, the SDK will:

- 1. in case of ClearVR content: infer the ConfentFormat automatically. This will always be correct as the ContentFormat is embedded in the stream.
- 2. in case of non-ClearVR content: assume the ContentFormat to be Rectiinear (aka traditional rectangular video).

If you want to play content of another projection type (e.g. monocopic ERP360), please configure this field accordingly, ref. ContentFormat enum.

When playing ClearVR or Mosaic content, you can always leave this field at its default value: ContentFormat.Unknown.

Declaration

public ContentFormat overrideContentFormat

Field Value

TYPE	DESCRIPTION
ContentFormat	

startViewportAndObjectPose

Since v7 Deprecated since v9.0

startViewportAndObjectPose specify the start pose of the viewport and start pose of the display object for the first frame of the new content.

Setting this value to null will assume that you want the ClearVRPlayer to decide where the object should be placed. This would be preferred behaviour. If you set this to non-null AND if you change more than just the display object's scale, automatic mesh placement will be disabled.

[!NOTE Normally, you would only set a custom ClearVRViewportAndObjectPose if you want to adjust the initial orientation of an omnidirectional video (e.g. you want to apply a 90 degree offset). In that case, set the values of choice on clearVRViewportAndObjectPose.displayobject.pose

Declaration

public ClearVRViewportAndObjectPose startViewportAndObjectPose

Field Value

TYPE	DESCRIPTION
ClearVRViewportAndObjectPose	

Properties

contentSupportedStatus

Whether playback of this clip is known to work or not on the current device. By default, this is unknown. One should use the TestIsContentSupported API to test for content compatibility.

Declaration

public ContentSupportedStatus contentSupportedStatus { get; }

Property Value

ТҮРЕ	DESCRIPTION
ContentSupportedStatus	

fishEyeSettings

Since v7.4 Optional in case of explicitly overriding the content format to configurable fish eye, i.e. setting ContentFormat.MonoscopicFishEye or ContentFormat.StereoscopicFishEyeSBS The class allows to configure camera and lens specific fish eye settings. Default value: null

Declaration

public FishEyeSettings fishEyeSettings { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
FishEyeSettings	

Methods

Clone()

Create a deepcopy of the object.

Declaration

public ContentItem Clone()

Returns

TYPE	DESCRIPTION
ContentItem	A deepcopy on the object.

Copy()

Returns a deep copy of this object.

Declaration

public ContentItem Copy()

Returns

ТҮРЕ	DESCRIPTION
ContentItem	A deep copy of this object.

ToString()

Returns a properly formatted description of this object as a string

Declaration

public override String ToString()

Returns

TYPE	DESCRIPTION
String	A verbose print of all object's fields

Verify()

Declaration

public bool Verify()

Returns

TYPE	DESCRIPTION
System.Boolean	

Class ContentItemList

Convenience array of ContentItems

Inheritance

System.Object

ContentItemList

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ContentItemList : object

Fields

content_items

Declaration

public ContentItem[] content_items

Field Value

TYPE	DESCRIPTION
ContentItem[]	

contentItems

Declaration

public ContentItem[] contentItems

Field Value

TYPE	DESCRIPTION
ContentItem[]	

Enum ContentProtectionRobustnessLevels

Specifies the Content Protection Robustness Level required to play encrypted content.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ContentProtectionRobustnessLevels : int

NAME	DESCRIPTION
HWSecureAll	On Android, thsi would equal to Widevine L1.
HWSecureCrypto	On Android, this is equal to Widevine L2 which is NOT supported.
HWSecureDecode	On Android, do not use. See HWSecureAll instead.
SWSecureCrypto	On Android: do not use. See SWSecureDecode if targeting Widevine L3
SWSecureDecode	On Android, this would equal to Widevine L3
Unprotected	No content protection required.

Enum ContentSupportedStatus

Enum describing whether a ContentItem can be played back or not. Note that this only holds for device configuration as specified when testing using the TestIsContentSupported API

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ContentSupportedStatus : int

NAME	DESCRIPTION	
NotSupported	Indicates that the ContentItem can NOT be played on the DeviceType and screen dimensions used while testing.	
Supported	Indicates that the ContentItem can be played on the DeviceType and screen dimensions used while testing.	
	Indicates that the ContentItem can be played on the DeviceType and screen dimensions used while testing BUT as monoscopic only.	
SupportedAsMonoscopicOnly	This value will only be returned if allowDecoderContraintsInducedStereoToMono is set to true on PlatformOptions or ContentSupportedTesterParameters, whichever comes first.	
Unknown	By default, it is unknown whether a ContentItem is supported or not. Note that even after a succesful test, the status might still be Unknown as only ClearVR clips can be tested.	

Class ContentSupportedTesterParameters

Parameters when testing whether a (or multiple) clip can be played back or not.

Inheritance

System.Object

ContentSupportedTesterParameters

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ContentSupportedTesterParameters : object

Constructors

 $\label{lem:contentSupportedTesterParameters} ContentSupportedTesterParameters (ref ContentSupportedTesterParameters), DeviceParameters, String, Boolean, TelemetryConfiguration)$

Default constructor. If one wants to specify HTTP/HTTPS proxy parameters, one can use the SetProxyParameters as you cannot directly assign to thehttpProxyParameters and httpsProxyParameters fields.

Declaration

public ContentSupportedTesterParameters(ref ContentItem[] argContentItemList, DeviceParameters
argDeviceParameters = null, String argOverrideUserAgent = null, Boolean
argAllowDecoderContraintsInducedStereoToMono = null, TelemetryConfiguration argTelemetryConfiguration = null)

TYPE	NAME	DESCRIPTION
ContentItem[]	argContentItemList	A list of ContentItems that need to be checked. Cannot be null, nor empty. Passed by reference, as their contentSupportedStatus fields will be overwritten.
DeviceParameters	argDeviceParameters	The device parameters. Keep at the default value null for auto-detection.

ТҮРЕ	NAME	DESCRIPTION
String	argOverrideUserAgent	OVerride the user agent, send with every HTTP request. One is strongly advised to always leave this at its default value, an empty string.
Boolean	argAllowDecoderContraintsInducedStereoToMono	If enabled, a stereoscopic clip will be played as monoscopic if the available video decoder has insufficient decoding capacity to decode it stereoscopically. Note that this can only be configured once, either when calling the TestIsContentSupported() API or Initializing a ClearVRPlayer, whichever comes first. Default value: false
TelemetryConfiguration	argTelemetryConfiguration	Configure telemetry services reporting. See TelemetryConfiguration for details. Default value: null

ContentSupportedTesterParameters(ref ContentItem[], DeviceTypes, Int16, Int16, String)

Default constructor. If one wants to specify HTTP/HTTPS proxy parameters, one can use the SetProxyParameters as you cannot directly assign to thehttpProxyParameters and httpsProxyParameters fields.

Declaration

public ContentSupportedTesterParameters(ref ContentItem[] argContentItemList, DeviceTypes argDeviceType =
default(DeviceTypes), short argScreenWidth = null, short argScreenHeight = null, String argOverrideUserAgent =
null)

TYPE	NAME	DESCRIPTION
ContentItem[]	argContentItemList	A list of ContentItems that need to be checked. Cannot be null, nor empty. Passed by reference, as their contentSupportedStatus fields will be overwritten.
DeviceTypes	argDeviceType	The device type. Keep at the default value (DeviceTypes.Unknown) for auto-detection.
System.Int16	argScreenWidth	The width of the screen in pixels. Keep at the default value 0 for auto-detection.
System.Int16	argScreenHeight	The height of the screen in pixels. Keep at the default value 0 for auto-detection.
String	argOverrideUserAgent	OVerride the user agent, send with every HTTP request. One is strongly advised to always leave this at its default value, an empty string.

Fields

allowDecoderContraintsInducedStereoToMono

In some cases, a stereoscopic clip cannot be rendered stereoscopic due to insufficient decoder capabilities. For example, playing a 12K stereoscopic clip requires a hardware decoder that can decode at least 6K. If you would've tried to play this clip on a 4K decoder, playback would simply fail. If you enable this option (e.g. set it to true), the player will attempt to render the clip in monoscopic in case insufficient decoder capacity is detected for stereoscopic playback, permitting that the decoder has sufficient capacity to do so. Default value: false

▲ WARNING

Note that this can only be configured once, either when calling the ClearVRPlayer.TestIsContentSupported() API or clearVRPlayer.Initialize(), whichever comes first. From that point onward, it cannot be changed until you completely stop the ClearVRPlayer.

Declaration

Field Value

TYPE	DESCRIPTION
System.Boolean	

httpProxyParameters

Since v5.1 Specify the HTTP proxy settings. Note that the lower-level SDK will attempt to detect proxy host and port automatically if host and port are at their default values ("<auto>" and -1 respectively). Due to platform security constraints, we cannot detect username and password automatically. We must rely on the application to provide those. See SetProxyParameters for details.

Declaration

public readonly ProxyParameters httpProxyParameters

Field Value

TYPE	DESCRIPTION
ProxyParameters	

httpsProxyParameters

Since v5.1 Specify the HTTPS proxy settings. Note that the lower-level SDK will attempt to detect proxy host and port automatically if host and port are at their default values ("<auto>" and -1 respectively). Due to platform security constraints, we cannot detect username and password automatically. We must rely on the application to provide those. See SetProxyParameters for details.

Declaration

public readonly ProxyParameters httpsProxyParameters

Field Value

TYPE	DESCRIPTION
ProxyParameters	

overrideUserAgent

This field allows you to override the user agent field in each video-streaming related HTTP request. As viewport-adaptive streaming results in many HTTP requests, one is STRONGLY discouraged to set this field as it will increase network overhead.. If needed, be sure to keep this String as short as possible. Default value: "" (an empty string)

Declaration

public String overrideUserAgent

Field Value

ТҮРЕ	DESCRIPTION
String	

telemetryConfiguration

Configure telemetry services reporting. See TelemetryConfiguration for details. Default value: null.

Declaration

public TelemetryConfiguration telemetryConfiguration

Field Value

ТҮРЕ	DESCRIPTION
TelemetryConfiguration	

Class ContentSupportedTesterReport

The TestIsContentSupported API returns a ContentSupportedTesterReport object upon completion. When parsing, one should first verify that the test was a success by calling GetIsSuccess() on the argClearVRMessage argument in the callback.

- Success == true the contentItemList will hold an array of the sameContentItem as you specified in the ContentSupportedTesterParameters argument of the original test call, as you passed those along by reference. Note that, even in case of a successful test, a ContentItem can still have a ContentSupportedStatus 'Unknown' because not all content item types are checken. Currently, only ClearVR clips (manifest.json) are tested.
- Success == false one can check the argClearVRMessage argument for details on why it has failed.

Inheritance

System.Object

ContentSupportedTesterReport

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ContentSupportedTesterReport : object

Properties

contentItemList

The list of ContentItems. These have all been checked and are a reference of the original once passed along.

Declaration

public ContentItem[] contentItemList { get; }

Property Value

TYPE	DESCRIPTION
ContentItem[]	

optionalArguments

Holds the optional arguments initially passed along in the call toTestIsContentSupported

Declaration

public object[] optionalArguments { get; }

Property Value

ТҮРЕ	DESCRIPTION
System.Object[]	The objects (nothing, one or multiple).

Methods

ToString()

Returns the contents of this report as a string.

Declaration

public override String ToString()

Returns

TYPE	DESCRIPTION
String	

Interface DebugInterface

Please do *not* use these APIs in your own project. They are for debugging purposes only and will change without notice

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public interface DebugInterface

Methods

CallCore(Byte[], Action<ClearVREvent, ClearVRPlayer>, Object[])

Send a message with instructions in the form of a protobuf message to the core.

Declaration

void CallCore(byte[] argRawMessage, Action<ClearVREvent, ClearVRPlayer>
argCbClearVRAsyncRequestResponseReceived, params object[] argOptionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
System.Byte[]	argRawMessage	
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	argCbClearVRAsyncRequestResponseReceived	The response event handler for this call
System.Object[]	argOptionalArguments	Any optional argument that you would like to pass inside the callback.

CallCoreSync(String)

Send a message to the core. Generally the message should be serialized protobuf defined in Core.proto.

Declaration

string CallCoreSync(string base64Message)

TYPE	NAME	DESCRIPTION
System.String	base64Message	The raw message as a string.

Returns

TYPE	DESCRIPTION
System.String	

ForceClearVRCoreCrash(Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Send a message to the core to make it forcefully crash.

Declaration

void ForceClearVRCoreCrash(Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] optionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onSuccess	An optional callback that should be triggered after the request was successfully completed. You are highly encouraged to implement the callback, but it can be null.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	An optional callback that should be triggered in case the request has failed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	optionalArguments	Any optional argument that you would like to pass inside the callback.

ForceClearVRCoreCrash(Action<ClearVREvent, ClearVRPlayer>, Object[])

Send a message to the core to make it forcefully crash.

Declaration

void ForceClearVRCoreCrash(Action<ClearVREvent, ClearVRPlayer>
argCbClearVRAsyncRequestResponseReceived, params object[] argOptionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	argCbClearVRAsyncRequestResponseReceived	The response event handler for this call
System.Object[]	argOptionalArguments	Any optional argument that you would like to pass inside the callback.

SwitchABRLevel(Boolean, Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Send a message to the core to switch the ABR level.

Declaration

void SwitchABRLevel(bool shouldIncrease, Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] optionalArguments)

TYPE	NAME	DESCRIPTION
System.Boolean	shouldIncrease	Should the ABR level go up or down? true if up, false if down
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onSuccess	An optional callback that should be triggered after the request was successfully completed. You are highly encouraged to implement the callback, but it can be null.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	An optional callback that should be triggered in case the request has failed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	optionalArguments	Any optional argument that you would like to pass inside the callback.

SwitchABRLevel(Boolean, Action<ClearVREvent, ClearVRPlayer>, Object[])

Send a message to the core to switch the ABR level.

Declaration

void SwitchABRLevel(bool argUp, Action<ClearVREvent, ClearVRPlayer>
argCbClearVRAsyncRequestResponseReceived, params object[] argOptionalArguments)

TYPE	NAME	DESCRIPTION
System.Boolean	argUp	Should the ABR level go up or down? true if up, false if down
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	argCbClearVRAsyncRequestResponseReceived	An optional callback that should be triggered after the request was completed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	argOptionalArguments	Any optional argument that you would like to pass inside the callback.

Class DeviceParameters

Helpers class holdig device-specific parameters.

Inheritance

System.Object

DeviceParameters

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class DeviceParameters : object

Constructors

DeviceParameters(Int16, Int16, DeviceTypes)

Default constructor

Declaration

public DeviceParameters(short argScreenWidth = null, short argScreenHeight = null, DeviceTypes
argDeviceType = default(DeviceTypes))

Parameters

TYPE	NAME	DESCRIPTION
System.Int16	argScreenWidth	The screen width in pixels. The default value 0 = auto-detect. For headsets, this is the full dual-eye equivalent (so if one eye has a resolution of 2160 pixels, this value should be 4320.
System.Int16	argScreenHeight	The screen height in pixels. The default value 0 = auto-detect.
DeviceTypes	argDeviceType	The Device Type. The default value DeviceTypes.Unknown will trigger auto-detection.

Properties

deviceType

Returns the configured device type.

Declaration

(c) Tiledmedia B.V. 2017-2022

public DeviceTypes deviceType { get; }

Property Value

TYPE	DESCRIPTION
DeviceTypes	DeviceType

screenHeight

Returns the configured screen height.

Declaration

public short screenHeight { get; }

Property Value

TYPE	DESCRIPTION
System.Int16	Screen height in pixels.

screenWidth

Returns the configured screen width.

Declaration

public short screenWidth { get; }

Property Value

TYPE	DESCRIPTION
System.Int16	Screen width in pixels.

Methods

ToString()

Declaration

public override string ToString()

Returns

TYPE	DESCRIPTION
System.String	

Enum DeviceTypes

A list of the various supported devices andd headsets. These are automatically detected. You can use Utils.GetDeviceType() to know which device type was detected.

Namespace	: com.tiledmedia.clearv
Assembly:	Assembly-CSharp.dll

Syntax

public enum Dev	/ice I ypes	:	ınt
-----------------	-------------	---	-----

NAME	DESCRIPTION
AndroidFlat	
AndroidGenericCardboard	
AndroidGenericDaydream	
AndroidGenericHMD	
AndroidGSXRGeneric	
AndroidMobfishCardboard	
AndroidOculusGearVR	
AndroidOculusGeneric	
AndroidOculusGo	
AndroidOculusQuest	
AndroidOculusQuest2	
AndroidPicoVRGeneric	
AndroidSkyworthVRGeneric	
AndroidWaveVRGeneric	

NAME	DESCRIPTION
AppleTV	
IOSFlat	
IOSGenericCardboard	
IOSGenericHMD	
IOSMobfishCardboard	
PCFlat	
PCGenericHMD	
PCHTCGeneric	
PCHTCVive	
PCHTCViveCosmos	
PCHTCVivePro	
PCOculusGeneric	
PCOculusLinkQuest	
PCOculusRiftCV1	
PCOculusRiftDK1	
PCOculusRiftDK2	
PCOculusRiftS	
PCValveGeneric	
PCValveIndex	
Tester	

NAME	DESCRIPTION
Unknown	

Enum DisplayObjectClassTypes

namespace	e: com.tileamedia.clearvi
Assembly:	Assembly-CSharp.dll

Syntax

NAME	DESCRIPTION
FullScreen	
LargePanel	
SmallPanel	
Thumbnail	
Unknown	

Enum DisplayObjectDescriptorFlags

An internal enum that is not of any use to the integrator.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum DisplayObjectDescriptorFlags : int

NAME	DESCRIPTION
ActiveStateChanged	
ClassTypeChanged	
Created	
FeedIndexChanged	
LateTextureLatch	
LateVertexUpload	
MeshChanged	
MeshUpdated	
ShaderParameterChanged	
TextureChanged	
TextureLatched	
TextureUpdated	
Unknown	Unknown is a valid value, returned when ClearVRDisplayObjectControllerBase::UpdateMesh() cannot complete because the mesh is (no longer) initialized.

Class DisplayObjectMapping

Parameters to specify the mapping between a DisplayObject and a feed

Inheritance

System.Object

DisplayObjectMapping

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class DisplayObjectMapping : object

Constructors

DisplayObjectMapping(ClearVRDisplayObjectControllerBase, Int32, DisplayObjectClassTypes, ContentFormat)

Constructor that fully set all the value of the display object to feed mapping

Declaration

public DisplayObjectMapping(ClearVRDisplayObjectControllerBase argClearVRDisplayObjectController, int argFeedIndex, DisplayObjectClassTypes argDisplayObjectClass, ContentFormat argContentFormat)

TYPE	NAME	DESCRIPTION
ClearVRDisplayObjectControllerBase	argClearVRDisplayObjectController	The ClearVRDisplayObjectControlle
System.Int32	argFeedIndex	The ID of the feed, in the selected ContentItem, we want map on the display object
DisplayObjectClassTypes	argDisplayObjectClass	The DisplayObject class of the selected display object.
ContentFormat	argContentFormat	The content format of this displation object.

Properties

clearVRDisplayObjectController

The ClearVRDisplayObjectController which this Display Object is mapped.

Declaration

public ClearVRDisplayObjectControllerBase clearVRDisplayObjectController { get; }

Property Value

TYPE	DESCRIPTION
ClearVRDisplayObjectControllerBase	

contentFormat

The ContentFormat of the Display Object.

Declaration

public ContentFormat { get; set; }

Property Value

TYPE	DESCRIPTION
ContentFormat	

displayObjectClassType

The class of the Display Object, refer to Display Object Class Types for details.

Declaration

 $\textcolor{red}{\textbf{public}} \ \textbf{DisplayObjectClassTypes} \ \textbf{displayObjectClassType} \ \{ \ \textbf{get}; \ \textbf{set}; \ \}$

Property Value

TYPE	DESCRIPTION
DisplayObjectClassTypes	

feedIndex

The feed index of the feed to which this Display Object is mapped.

Declaration

public int feedIndex { get; set; }

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

ToString()

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	

Class DRMInfo

This helper class specifies information on a clip's DRM protection. This class supports token and certificat-based decryption, but DRM protected content playback is only support on Android.

Inheritance

System.Object

DRMInfo

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class DRMInfo: object

Constructors

DRMInfo(ClearVRDRMLicenseServerTypes)

Default constructor. Specify fields like url, token and/or certificate data manually.

Declaration

public DRMInfo(ClearVRDRMLicenseServerTypes argClearVRDRMLicenseServerType)

Parameters

TYPE	NAME	DESCRIPTION
ClearVRDRMLicenseServerTypes	argClearVRDRMLicenseServerType	

DRMInfo(ClearVRDRMLicenseServerTypes, KeyValuePair<String, String>[], KeyValuePair<String, String>[], KeyValuePair<String, String>[])

Construct a DRMInfo object for content that is protected with a license- and tokenization key/value pair. This constructor can be used for content that is not protected by a certificate/key and/or a token/password. This is a convenience constructor.

Declaration

public DRMInfo(ClearVRDRMLicenseServerTypes argClearVRDRMLicenseServerType, KeyValuePair<String,
String>[] argLicenseAuthenticationHeaders, KeyValuePair<String, String>[] argTokenisationQueryStrings, KeyValuePair<String, String>[] argTokenisationHeaders,
KeyValuePair<String, String>[] argTokenisationQueryStrings)

Parameters

TYPE	NAME	DESCRIPTION
ClearVRDRMLicenseServerTypes	argClearVRDRMLicenseServerType	The DRMLicenseServerType. Cannot be null. WHen using this constructor, this value is typically set to Unspecified.
KeyValuePair <string, string="">[]</string,>	argLicenseAuthenticationHeaders	The license authentication headers. The argument can be null, but keys and values can not be null.
KeyValuePair <string, string="">[]</string,>	argLicenseAuthenticationQueryStrings	The license authentication query strings value. The argument can be null, but keys and values can not be null.
KeyValuePair <string, string="">[]</string,>	argTokenisationHeaders	The tokenisation headers. The argument can be null, but keys and values can not be null.
KeyValuePair <string, string="">[]</string,>	argTokenisationQueryStrings	The tokenisation query strings. The argument can be null, but keys and values can not be null.

DRMInfo(ClearVRDRMLicenseServerTypes, String, String, Byte[], Byte[], Byte[], String)

Construct a DRMInfo object for content that is protected by a certificate/key and/or a token/password. This is a convenience constructor

Declaration

public DRMInfo(ClearVRDRMLicenseServerTypes argClearVRDRMLicenseServerType, String argUrl, String argToken, byte[] argCertificate, byte[] argKey, byte[] argCAChain, String argPassword)

Parameters

TYPE	NAME	DESCRIPTION
ClearVRDRMLicenseServerTypes	argClearVRDRMLicenseServerType	The DRMLicenseServerType. Cannot be null.
String	argUrl	The DRM license server URL. Can be null.
String	argToken	The token to use during validation. Can be null.
System.Byte[]	argCertificate	The certificate to use. Can be null.
System.Byte[]	argKey	The key to use. Can be null.
System.Byte[]	argCAChain	The CA Chain certificate to use. Can be null.
String	argPassword	The password required to decrypt the certificate(s). Can be null.

DRMInfo(ClearVRDRMLicenseServerTypes, String, String, Byte[], Byte[], Byte[], String, KeyValuePair<String, String>[], KeyValuePair<String, String>[], KeyValuePair<String, String>[])

Construct a DRMInfo object by setting all its fields explicitly. Note that there are convenience constructors that might suit your need even better.

Declaration

public DRMInfo(ClearVRDRMLicenseServerTypes argClearVRDRMLicenseServerType, String argUrl, String argToken, byte[] argCertificate, byte[] argKey, byte[] argCAChain, String argPassword, KeyValuePair<String, String>[] argLicenseAuthenticationHeaders, KeyValuePair<String, String>[] argTokenisationHeaders, KeyValuePair<String, String>[] argTokenisationHeaders, KeyValuePair<String, String>[] argTokenisationHeaders, KeyValuePair<String, String>[] argTokenisationQueryStrings)

Parameters

TYPE	NAME	DESCRIPTION
ClearVRDRMLicenseServerTypes	argClearVRDRMLicenseServerType	The DRMLicenseServerType. Cannot be null.
String	argUrl	The DRM license server URL. Can be null.
String	argToken	The token to use during validation. Can be null.
System.Byte[]	argCertificate	The certificate to use. Can be null.
System.Byte[]	argKey	The key to use. Can be null.
System.Byte[]	argCAChain	The CA Chain certificate to use. Can be null.
String	argPassword	The password required to decrypt the certificate(s). The argument can be null, but keys and values can not be null.
KeyValuePair <string, string="">[]</string,>	argLicenseAuthenticationHeaders	The license authentication headers. The argument can be null, but keys and values can not be null.

TYPE	NAME	DESCRIPTION
KeyValuePair <string, string="">[]</string,>	argLicenseAuthenticationQueryStrings	The license authentication query strings value. The argument can be null, but keys and values can not be null.
KeyValuePair <string, string="">[]</string,>	argTokenisationHeaders	The tokenisation headers. The argument can be null, but keys and values can not be null.
KeyValuePair <string, string="">[]</string,>	argTokenisationQueryStrings	The tokenisation query strings. The argument can be null, but keys and values can not be null.

Properties

caChain

Specifies the CA certificate if a non-standard CA authority is used.

Declaration

public byte[] caChain { get; set; }

Property Value

TYPE	DESCRIPTION
System.Byte[]	

certificate

Specifies the certificate.

Declaration

public byte[] certificate { get; set; }

TYPE	DESCRIPTION
System.Byte[]	

clearVRDRMLicenseServerType

The DRM License Server type. Note that Unspecified is a valid value.

Declaration

public ClearVRDRMLicenseServerTypes clearVRDRMLicenseServerType { get; }

Property Value

TYPE	DESCRIPTION
ClearVRDRMLicenseServerTypes	

ivOverrideBase64

Override the decryption IVR, in BASE64 format.

A WARNING

This is an advanced field which should not be used in your application unless you know exactly what you're doing.

Declaration

public String ivOverrideBase64 { get; set; }

Property Value

TYPE	DESCRIPTION
String	

key

Specifies the key

Declaration

public byte[] key { get; set; }

TYPE	DESCRIPTION
System.Byte[]	

keyOverrideBase64

Override the decryption key, in BASE64 format.

A WARNING

This is an advanced field which should not be used in your application unless you know exactly what you're doing.

Declaration

public String keyOverrideBase64 { get; set; }

Property Value

TYPE	DESCRIPTION
String	

licenseAuthenticationHeaders

An array of license authentication header key/value pairs.

Declaration

public KeyValuePair<String, String>[] licenseAuthenticationHeaders { get; set; }

Property Value

TYPE	DESCRIPTION
KeyValuePair <string, String>[]</string, 	The license authentication header key/value pairs, as set during object construction. Can be empty, but cannot be null.

licenseAuthenticationQueryStrings

An array of license authentication query string key/value pairs.

Declaration

public KeyValuePair<String, String>[] licenseAuthenticationQueryStrings { get; set; }

TYPE	DESCRIPTION
KeyValuePair <string, String>[]</string, 	The license authentication query string key/value pairs, as set during object construction. Can be empty, but cannot be null.

password

THe password, use to decrypt the crypto keys if applicable.

Declaration

```
public String password { get; set; }
```

Property Value

ТҮРЕ	DESCRIPTION
String	

token

The token used for DRM decryption.

Declaration

```
public String token { get; set; }
```

Property Value

TYPE	DESCRIPTION
String	

tokenisationHeaders

An array of tokenisation header key/value pairs.

Declaration

```
public KeyValuePair<String, String>[] tokenisationHeaders { get; set; }
```

TYPE	DESCRIPTION
KeyValuePair <string, String>[]</string, 	The tokenisation header key/value pairs, as set during object construction. Can be empty, but cannot be null.

tokenisationQueryStrings

An array of tokenisation query strings key/value pairs.

Declaration

public KeyValuePair<String, String>[] tokenisationQueryStrings { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
KeyValuePair <string, String>[]</string, 	The license authentication query string key/value pairs, as set during object construction. Can be empty, but cannot be null.

tokenizationHeaders

An array of tokenization header key/value pairs.

Declaration

public KeyValuePair<String, String>[] tokenizationHeaders { get; set; }

Property Value

TYPE	DESCRIPTION
KeyValuePair <string, String>[]</string, 	The tokenization header key/value pairs, as set during object construction. Can be empty, but cannot be null.

tokenizationQueryStrings

An array of tokenization query strings key/value pairs.

Declaration

public KeyValuePair<String, String>[] tokenizationQueryStrings { get; set; }

ТҮРЕ	DESCRIPTION
KeyValuePair <string, String>[]</string, 	The license authentication query string key/value pairs, as set during object construction. Can be empty, but cannot be null.

url

The DRM server URL

Declaration

public String url { get; set; }

Property Value

TYPE	DESCRIPTION
String	

Methods

Copy()

Construct a deep copy of the object.

Declaration

public DRMInfo Copy()

Returns

TYPE	DESCRIPTION
DRMInfo	A deep copy of the current object.

ToString()

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	

Enum DRMTypes

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum DRMTypes : int	
----------------------------	--

NAME	DESCRIPTION
Fairplay	
HlsAes128	
HIsSampleAes	
None	
Playready	
Tbd	
Unspecified	
Widevine	

Enum EventTypes

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum EventTypes : int		
------------------------------	--	--

NAME	DESCRIPTION
FinishedLive	
Live	
Unknown	
Vod	

Enum FallbackCubefaceIdentifier

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum	FallbackCubefaceIdentifier	int
-------------	----------------------------	-----

NAME	DESCRIPTION
LeftEyeBack	
LeftEyeBottom	
LeftEyeFront	
LeftEyeLeft	
LeftEyeRight	
LeftEyeTop	
RightEyeBack	
RightEyeBottom	
RightEyeFront	
RightEyeLeft	
RightEyeRight	
RightEyeTop	
Unknown	

Struct FallbackCubefaceInfoStruct

Advanced API.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public struct FallbackCubefaceInfoStruct

Fields

fallbackCubefaceIdentifier

Declaration

public FallbackCubefaceIdentifier fallbackCubefaceIdentifier

Field Value

TYPE	DESCRIPTION
FallbackCubefaceIdentifier	

height

Declaration

public Int32 height

Field Value

TYPE	DESCRIPTION
Int32	

topLeftX

Declaration

public Int32 topLeftX

Field Value

TYPE	DESCRIPTION
Int32	

topLeftY

Declaration

public Int32 topLeftY

Field Value

TYPE	DESCRIPTION
Int32	

width

Declaration

public Int32 width

Field Value

TYPE	DESCRIPTION
Int32	

Methods

ToString()

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	

Class FallbackLayout

Advanced API, do not use.

Inheritance

System.Object

FallbackLayout

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class FallbackLayout : object

Fields

fallbackCubefaceInfoStructs

Declaration

public List<System.Object> fallbackCubefaceInfoStructs

Field Value

TYPE	DESCRIPTION
List <system.object></system.object>	

fallbackLayoutldentifier

Declaration

public System. Object fallback Layout Identifier

Field Value

ТҮРЕ	DESCRIPTION
System.Object	

overlapX

Declaration

public Int32 overlapX

Field Value

TYPE	DESCRIPTION
Int32	

overlapY

Declaration

public Int32 overlapY

Field Value

TYPE	DESCRIPTION
Int32	

Methods

ToString()

Declaration

public override String ToString()

ТҮРЕ	DESCRIPTION
String	

Enum FallbackLayoutIdentifier

Advanced enum, do not use.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum FallbackLayoutIdentifier : int

NAME	DESCRIPTION
CubemapThreeByTwo	
CubemapTwoByOne	
Unknown	

Class FeedInfo

Helper class that holds the info of one feed. For detailed information about the ContentInfo, FeedInfo, VideoTrackInfo, AudioTrackInfo, SubtitleTrackInfo objects and their relationship, refer to the ContentItem documentation.

Inheritance

System.Object

FeedInfo

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class FeedInfo: object

Properties

audioTracks

array that contains all the audio tracks that this feed contains.

Declaration

public AudioTrackInfo[] audioTracks { get; }

Property Value

ТҮРЕ	DESCRIPTION
AudioTrackInfo[]	

feedIndex

The index of the feed in the ContentInfo's list of feeds.

Declaration

public int feedIndex { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

subtitleTracks

ara, ma comano an mo casmo nacio ma mo reca comano

Declaration

public SubtitleTrackInfo[] subtitleTracks { get; }

Property Value

TYPE	DESCRIPTION
SubtitleTrackInfo[]	

url

Get the URL

Declaration

public String url { get; }

Property Value

TYPE	DESCRIPTION
String	

videoTracks

array that contains all the video tracks that this feed contains.

Declaration

public VideoTrackInfo[] videoTracks { get; }

Property Value

TYPE	DESCRIPTION
VideoTrackInfo[]	

Methods

GetActiveAudioTrack()

Returns the currently active audio track.

Declaration

public AudioTrackInfo GetActiveAudioTrack()

Returns

TYPE	DESCRIPTION
AudioTrackInfo	An AudioTrackInfo object which represents the active audio track, or null if no audio track is active.

GetActiveSubtitleTrack()

Returns the currently active subtitle track.

Declaration

public SubtitleTrackInfo GetActiveSubtitleTrack()

Returns

TYPE	DESCRIPTION
SubtitleTrackInfo	A SubtitleTrack object which represents the active subtitle track, or null if no subtitle track is active.

GetActiveVideoTrack()

Returns the currently active video track in this Feed, or null if no Video track is active.

Declaration

public VideoTrackInfo GetActiveVideoTrack()

Returns

TYPE	DESCRIPTION
VideoTrackInfo	The currently active VideoTrack, or null if none are active.

GetActiveVideoTracks()

Returns the currently active video track(s).

Declaration

public VideoTrackInfo[] GetActiveVideoTracks()

TYPE	DESCRIPTION
VideoTrackInfo[]	An array of all active video tracks as VideoTrackInfo objects. The size can be 0, but this API will never return null.

GetNumberOfAudioTracks()

Returns the number of AudioTracks in this Feed. This returns the same value as audioTracks.Length.

Declaration

public int GetNumberOfAudioTracks()

Returns

TYPE	DESCRIPTION
System.Int32	The number of audio tracks.

GetNumberOfSelectableAudioTracks()

Returns the number of selectable audio tracks on this Feed. SeeGetSelectableAudioTrackIDs for important details on the definition of a "selectable" audio track.

Declaration

public int GetNumberOfSelectableAudioTracks()

Returns

TYPE	DESCRIPTION
System.Int32	Returns the number of selectable audio tracks as a non-negative integer value. 0 means that there are no selectable audio tracks.

GetNumberOfSubtitleTracks()

Returns the number of SubtitleTracks in this Feed. This returns the same value as subtitle.Length.

Declaration

public int GetNumberOfSubtitleTracks()

TYPE	DESCRIPTION
System.Int32	The number of subtitle tracks.

GetNumberOfVideoTracks()

Returns the number of VideoTracks in this Feed. This returns the same value as videoTracks.Length.

Declaration

public int GetNumberOfVideoTracks()

Returns

TYPE	DESCRIPTION
System.Int32	The number of video tracks.

GetSelectableAudioTrackIDs()

Returns a list of selectable audio tracks. Selectable audio tracks are defined as:

- 1. the currently active audio track
- 2. the audiotracks that are *not* bound to any video track (e.g. they are not multiplexed together with another (A/V) track).
- 3. the audiotracks that are bound to a video track that is currently active.

This list will never contain duplicates.

1 NOTE

If you are looking for the subtitle track metadata, see the GetSelectableAudioTracks() API.

Declaration

public TrackID[] GetSelectableAudioTrackIDs()

TYPE

TYPE	DESCRIPTION
TrackID[]	The output list will contains zero, one or more TrackIDs objects which uniquely identify selectable audio tracks. This API will never return null.

GetSelectableAudioTracks()

Returns a list of AudioTrackInfos of selectable audio tracks within this Feed. Selectable audio tracks are defined as:

- 1. the currently active audio track
- 2. the audiotracks that are *not* bound to any video track (e.g. they are not multiplexed together with another (A/V) track).
- 3. the audiotracks that are bound to a video track that is currently active.

This list will never contain duplicates.

Declaration

public AudioTrackInfo[] GetSelectableAudioTracks()

Returns

TYPE	DESCRIPTION
AudioTrackInfo[]	The output list will contains zero, one or more AudioTrackInfo objects. Each AudioTrackInfo is unique identified by its audioTrackID field. This API will never return null.

GetSelectableSubtitlesTrackIDs()

Returns a list of selectable subtitle tracks. This list will never contain duplicates.



If you are looking for the subtitle track metadata, see the GetSelectableSubtitlesTracks() API.

Declaration

public TrackID[] GetSelectableSubtitlesTrackIDs()

TYPE	DESCRIPTION
TrackID[]	The output list will contains zero, one or more TrackIDs objects which will uniquely identify selectable subtitle tracks. This API will never return null.

Get Selectable Subtitles Tracks ()

Returns a list of selectable subtitle tracks. This list will never contain duplicates.

Declaration

public SubtitleTrackInfo[] GetSelectableSubtitlesTracks()

Returns

TYPE	DESCRIPTION
SubtitleTrackInfo[]	The output list will contains zero, one or more TrackIDs objects which will uniquely identify selectable subtitle tracks. This API will never return null.

ToString()

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	

Enum FishEyeCameraAndLensTypes

Legacy enum that was used for presets for known camera types.

A WARNING

The FishEyeCameraAndLensTypes enum is deprecated. One must use the FisheyePresets enum instead.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum FishEyeCameraAndLensTypes : int

NAME	DESCRIPTION
BlackmagicUrsa12KCanon8158Mmf48K169	Use the preset for the Blackmagic Ursa 12K Camera with Canon 815 lens at 8 mm, focal length 4 8K sensor in 16:9.
BlackmagicURSAMiniCanon815	use the preset for the blackmagic URSA Mini Canon 815
CustomCameraAndLens	No preset used for the fish eye camera. Can be used if a custom camera and lens type combination is used not covered by any of the provided presets.
RedKomodo6KCanon812	Use the preset for the Red Komodo 6K Camera with Canon 815 lens
RedVRaptor8KCanon81510	Use the preset for the Red V-Raptor 8K Camera with Canon 815 lens at 10 mm, focal length 4 8K sensor in 16:9.
RedVRaptor8KCanon81513	Use the preset for the Red V-Raptor 8K Camera with Canon 815 lens at 13 mm, focal length 4 8K sensor in 16:9.

NAME	DESCRIPTION
RedVRaptor8KCanon8158	Use the preset for the Red V-Raptor 8K Camera with Canon 815 lens at 8 mm, focal length 4 8K sensor in 16:9.
ZCamK1ProlzugarMkx22mft	Use the preset for the Z cam K1 pro with Pro Izugar Mkx 22mft sensor
ZCamK2ProlzugarMkx200	Use the preset for the Z cam K2 pro with Pro Izugar Mkx 200 sensor

Enum FishEyeLensTypes

Lens types for fish eye cameras

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum FishEyeLensTypes : int

NAME	DESCRIPTION
DefaultFisheyeLens	Default lens for the fish eye camera
Equidistant	Equidistant lens for the fish eye camera
Equisolid	Equisolid lens for the fish eye camera
	Polynomial lens model for the fish eye camera
Polynomial	▲ WARNING This value is not yet supported. Setting this value will result in undefined behaviour. Support will be added in a future update.

Enum FisheyePresets

Fish-eye settings presets for known camera types

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum FisheyePresets : int

NAME	DESCRIPTION
BlackmagicUrsa12Kcanon8_15_8Mm_8K_16_9	Use the preset for the Blackmagic Ursa 12K Camera with Canon 815 lens at 8 mm, focal length 4 8K sensor in 16:9.
BlackmagicUrsaMiniCanon8_15_10Dot5Mm	Use the preset for the blackmagic URSA Mini Canon 8-15 at 10.5mm.
BlackmagicUrsaMiniCanon8_15_10Mm	Use the preset for the blackmagic URSA Mini Canon 8-15 at 10mm.
BlackmagicUrsaMiniCanon8_15_11Dot5Mm	Use the preset for the blackmagic URSA Mini Canon 8-15 at 11.5mm.
BlackmagicUrsaMiniCanon8_15_11Mm	Use the preset for the blackmagic URSA Mini Canon 8-15 at 11mm.
BlackmagicUrsaMiniCanon8_15_12Dot5Mm	Use the preset for the blackmagic URSA Mini Canon 8-15 at 12.5mm.
BlackmagicUrsaMiniCanon8_15_12Mm	Use the preset for the blackmagic URSA Mini Canon 8-15 at 12mm.

NAME	DESCRIPTION
BlackmagicUrsaMiniCanon8_15_8Dot5Mm	Use the preset for the blackmagic URSA Mini Canon 8-15 at 8.5mm.
BlackmagicUrsaMiniCanon8_15_8Mm	Use the preset for the blackmagic URSA Mini Canon 8-15 at 8mm.
BlackmagicUrsaMiniCanon8_15_9Dot5Mm	Use the preset for the blackmagic URSA Mini Canon 8-15 at 9.5mm.
BlackmagicUrsaMiniCanon8_15_9Mm	Use the preset for the blackmagic URSA Mini Canon 8-15 at 9mm.
Custom	No preset used for the fish eye camera. Can be used if a custom camera and lens type combination is used not covered by any of the provided presets.
RedKomodo6Kcanon8_12_8Mm	Use the preset for the Red Komodo 6K Camera with Canon 815 lens.
RedVraptor8Kcanon8_15_10Mm	Use the preset for the Red V-Raptor 8K Camera with Canon 815 lens at 10 mm, focal length 4 8K sensor in 16:9.
RedVraptor8Kcanon8_15_13Mm	Use the preset for the Red V-Raptor 8K Camera with Canon 815 lens at 13 mm, focal length 4 8K sensor in 16:9.
RedVraptor8Kcanon8_15_8Mm	Use the preset for the Red V-Raptor 8K Camera with Canon 815 lens at 8 mm, focal length 4 8K sensor in 16:9.

NAME	DESCRIPTION
Zcamk1ProIzugarMkx22Mft_3Dot25Mm	Use the preset for the Z cam K1 pro with Pro Izugar Mkx 22mft sensor.
Zcamk2ProIzugarMkx200_3Dot8Mm	Use the preset for the Z cam K2 pro with Pro Izugar Mkx 200 sensor.

Class FishEyeSettings

Describes the exact fish eye lens parameters. A number of pre-defined lenses are available, see FisheyePresets.

Inheritance

System.Object

FishEyeSettings

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class FishEyeSettings: object

Constructors

FishEyeSettings(FishEyeCameraAndLensTypes, FishEyeLensTypes, Single, Single, Int32, Int32)

The ClearVR SDK supports various well-known camera and lens-type combinations out of the box, saving you from the trouble to specify all parameters manually. If you set this value to anything other than Custom, all other fields will be ignored. Typically, it suffices to only specify the first argument (argFishEyeCameraAndLensType). You can leave all other arguments at their default value.

▲ WARNING

Specifying a custom camera and lens type is considered an expert feature of this SDK. If you run into problems, please contact Tiledmedia.

A WARNING

The FishEyeCameraAndLensTypes enum is deprecated. One must use the FisheyePresets enum instead.

Declaration

public FishEyeSettings(FishEyeCameraAndLensTypes argFishEyeCameraAndLensType, FishEyeLensTypes argFishEyeLensType = default(FishEyeLensTypes), float argFocalLength = null, float argSensorPixelDensity = null, int argReferenceWidth = 0, int argReferenceHeight = 0)

Parameters

ТҮРЕ	NAME	DESCRIPTION
------	------	-------------

TYPE	NAME	DESCRIPTION
FishEyeCameraAndLensTypes	argFishEyeCameraAndLensType	Select a particular camera and lens combination for default settings, or specify everything with CUSTOM
FishEyeLensTypes	argFishEyeLensType	The shape of the fisheye lens"
System.Single	argFocalLength	The effective focal length for the lens in millimeters
System.Single	argSensorPixelDensity	The pixel density for the sensor per square millimeter
System.Int32	argReferenceWidth	The Reference width for which density parameter is correct. Defaults to the available pixels, but might be unequal if a zoom is applied
System.Int32	argReferenceHeight	The Reference height for which density parameter is correct. Defaults to the available pixels, but might be unequal if a zoom is applied

FishEyeSettings(FisheyePresets, FishEyeLensTypes, Single, Single, Int32, Int32)

Declaration

public FishEyeSettings(FisheyePresets fisheyePreset, FishEyeLensTypes fisheyeLensType =
 default(FishEyeLensTypes), float focalLength = null, float sensorPixelDensity = null, int referenceWidth = 0, int
 referenceHeight = 0)

Parameters

TYPE	NAME	DESCRIPTION
FisheyePresets	fisheyePreset	

TYPE	NAME	DESCRIPTION
FishEyeLensTypes	fisheyeLensType	
System.Single	focalLength	
System.Single	sensorPixelDensity	
System.Int32	referenceWidth	
System.Int32	referenceHeight	

Methods

Copy()

Declaration

public FishEyeSettings Copy()

Returns

TYPE	DESCRIPTION
FishEyeSettings	

ToString()

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	

Enum InitializeFlags

Flags specific to initializing the ClearVRPlayer.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum InitializeFlags : int

Fields

NAME	DESCRIPTION
LongBuffer	Force long buffer for more stable playout (not recommended).
NoCachePrewarming	Disable cache prewarming, reduces initialization time.
None	Default, no flags raised

Enum InteractionModes

This enum is used to help distinguish between different interaction modes depending on the content type.

Namespace: com.tiledmedia.clearv	1
Assembly: Assembly-CSharp.dll	

Syntax

public enum InteractionModes : int

Fields

NAME	DESCRIPTION
OmniDirectional	
Planar	
Rectilinear	
Unknown	

Class LayoutParameters

Parameters to specify the feed to display object mapping for the SetFeedLayout API This object has a mutable builder pattern.

Inheritance

System.Object

LayoutParameters

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class LayoutParameters : object

Constructors

LayoutParameters(LayoutParameters)

Copy constructor

Declaration

public LayoutParameters(LayoutParameters argOtherLayoutParameters)

Parameters

TYPE	NAME	DESCRIPTION
LayoutParameters	argOtherLayoutParameters	

LayoutParameters(TrackID, TrackID, Boolean)

Default constructor. Initialize the configuration to an empty mapping with default audio and subtitle selection

Declaration

public LayoutParameters(TrackID argAudioTrackID = null, TrackID argSubtitleTrackID = null, bool argDisableFallbackSwitch = false)

TYPE	NAME	DESCRIPTION
TrackID	argAudioTrackID	The audio TrackID. Default value: null (meaning: autoselect the first available audio track.

TYPE	NAME	DESCRIPTION
TrackID	argSubtitleTrackID	The subtitle TrackID. Default value: null (meaning: autoselect the first available subtitle track.
System.Boolean	argDisableFallbackSwitch	Whether to disable fallback-based fast switching. Default value: false. You are strongly recommended to not change this value

Properties

audioTrackID

Specifies the audio feed index and track index.

Declaration

public TrackID audioTrackID { get; set; }

Property Value

TYPE	DESCRIPTION
TrackID	

disableFallbackSwitch

This parameter applies to mosaic playback. Imagine you have:

- 1. one fullscreen display object 0 showing feed 0 in highest quality
- 2. one thumbnail display object 1 showing feed 1 in thumbnail quality

Now you switch feed 0 and 1, showing feed 1 on display object 0 and feed 0 on display object 1.

Switching behaviour depends on the value of this argument:

If disableFallbackSwitch == false (default value): the switch is immediate but you will briefly see feed 1 in thumbnail quality on the fullscreen display object 0. If disableFallbackSwitch == true: the switch will complete once feed 1 is available in highest quality. The responsiveness depends on the network conditions, but the user will always see the highest quality.

▲ WARNING

This is an advanced parameter, you are strongly recommened to NOT change its default value

Declaration

public bool disableFallbackSwitch { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

displayObjectMappings

Declaration

public List<DisplayObjectMapping> displayObjectMappings { get; set; }

Property Value

TYPE	DESCRIPTION
List <displayobjectmapping></displayobjectmapping>	

name

Declaration

public string name { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
System.String	

preferredAudioLanguage

The preferred audio language as an ISO-639 language code.

Declaration

public String preferredAudioLanguage { get; set; }

Property Value

TYPE	DESCRIPTION
String	

preferredSubtitlesLanguage

Declaration

public String preferredSubtitlesLanguage { get; set; }

Property Value

TYPE	DESCRIPTION
String	

subtitleTrackID

Specifies the subtitle feed index and track index.

Declaration

public TrackID subtitleTrackID { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
TrackID	

Methods

Clone()

Creates and returns a deep-copy of the object.

Declaration

public LayoutParameters Clone()

Returns

TYPE	DESCRIPTION
LayoutParameters	

GetFullScreenDisplayObjectMapping()

Returns a the DisplayObjectMapping object that has displayObjectClassType set to FullScreen. Remember that there can be only 0 or 1 DisplayObjectMappings with this DisplayObjectClassType.

Declaration

public DisplayObjectMapping GetFullScreenDisplayObjectMapping()

Returns

ТҮРЕ	DESCRIPTION	
DisplayObjectMapping	Returns null if not present, the appropriate DisplayObjectMapping otherwise.	

ToString()

Declaration

public override String ToString()

Returns

ТҮРЕ	DESCRIPTION
String	

Enum LogComponents

wamespace	:: com.tileamedia.clearv
Assembly:	Assembly-CSharp.dll

Syntax

public enum LogComponents : int		
---------------------------------	--	--

Fields

NAME	DESCRIPTION
MediaFlow	
Nrp	
Sdk	
SigmaAudio	
TmCore	
Unknown	

Class Logger

Class that handles the ClearVR and App logging.

Inheritance

System.Object

Logger

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class Logger: object

Constructors

Logger(String, Int64)

Declaration

public Logger(string argStrLogFileName, long argMaxLogFileSize = null)

Parameters

TYPE	NAME	DESCRIPTION
System.String	argStrLogFileName	
System.Int64	argMaxLogFileSize	

Properties

AppLogger

Logger instance that logs the general app messages.

Declaration

public static Logger AppLogger { get; }

Property Value

TYPE	DESCRIPTION
Logger	

Methods

Log(Exception)

(c) Tiledmedia B.V. 2017-2022

Writes exceptions to log files

Declaration

public void Log(Exception ex)

Parameters

TYPE	NAME	DESCRIPTION
Exception	ex	Pass the exception ex as parameter.

Log(String, String, LogType)

Log a Unity message.

Declaration

public void Log(string strMessage, string stackTrace, LogType logType)

TYPE	NAME	DESCRIPTION
System.String	strMessage	message to log.
System.String	stackTrace	log's stack trace.
LogType	logType	log's type (Warning, Info, Error, etc.)

Class LoggingConfiguration

This class is used to configure the log level and logging output of each component of Tiledmedia SDK. Do not forget to call ClearVRPlayer.EnableLogging(loggingConfiguration) to apply the configuration.

A WARNING

Take special care to keep only the default constructed instance of this class in the ClearVRPlayer to avoid performance impacts

Inheritance

System.Object

LoggingConfiguration

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class LoggingConfiguration : object

Properties

coreLogFileName

To override the core log file name

Declaration

public string coreLogFileName { get; set; }

Property Value

TYPE	DESCRIPTION
System.String	

coreLogLevel

To override the core log level

Declaration

public LogLevels coreLogLevel { get; set; }

Property Value

TYPE	DESCRIPTION
LogLevels	

coreLogToMemory

To override whether the core component performs in-memory logging or not. Default value: false.

Declaration

public bool coreLogToMemory { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

coreLogToStderr

To override whether the core component logs to stderr or not. Default value: false.

Declaration

public bool coreLogToStderr { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

globalLogFileName

Specify the output file name of the logs. Empty string means stdout and is the default value. A nonempty string means logging to a file located at the path specified in globalLogPath. By default all components log to the same output if not overriden

Declaration

public string globalLogFileName { get; set; }

Property Value

TYPE	DESCRIPTION
System.String	

globalLogFolder

Specify the output folder of the logs of all components.

Declaration

public string globalLogFolder { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
System.String	

globalLogLevel

globalLogLevel specify the default log verbosity for all Tiledmedia SDK components. Component log level can be overriden by setting explicitly the log level of that component In production this value should always be LogLevels. Warn

Declaration

public LogLevels globalLogLevel { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
LogLevels	

globalLogToMemory

Specify whether all components should perform in-memory logging or not. Default value: false.

Declaration

public bool globalLogToMemory { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

globalLogToStderr

Specify whether all components should log to stderr or not. Default value: false (meaning that they will log to file).

Declaration

public bool globalLogToStderr { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

interactionEventRecorderFileName

Specify the output file name of the interaction recorder. Empty string means interaction recorder is diabled. The output folder is the same as for the global log files.

Declaration

public string interactionEventRecorderFileName { get; set; }

Property Value

TYPE	DESCRIPTION
System.String	

logNetwork

Enable network logging. Default value: false.

Declaration

public bool logNetwork { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

mfLogFileName

To override the MediaFlow log file name

Declaration

public string mfLogFileName { get; set; }

Property Value

TYPE	DESCRIPTION
System.String	

mfLogLevel

To override the MediaFlow log level

Declaration

public LogLevels mfLogLevel { get; set; }

Property Value

TYPE	DESCRIPTION
LogLevels	

mfLogToMemory

To override whether the MediaFlow component performs in-memory logging or not. Default value: false.

Declaration

public bool mfLogToMemory { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

mfLogToStderr

To override whether the MediaFlow component logs to stderr or not. Default value: false.

Declaration

public bool mfLogToStderr { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

nrpLogFileName

To override the NRP log file name

Declaration

public string nrpLogFileName { get; set; }

Property Value

TYPE	DESCRIPTION
System.String	

nrpLogLevel

To override the NRP log level

Declaration

public LogLevels nrpLogLevel { get; set; }

Property Value

TYPE	DESCRIPTION
LogLevels	

nrpLogToMemory

To override whether the NRP component performs in-memory logging or not. Default value: false.

Declaration

public bool nrpLogToMemory { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

nrpLogToStderr

To override whether the NRP component logs to stderr or not. Default value: false.

Declaration

public bool nrpLogToStderr { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

sdkLogFileName

To override the SDK log file name

Declaration

public string sdkLogFileName { get; set; }

Property Value

TYPE	DESCRIPTION
System.String	

sdkLogLevel

To override the SDK log level

Declaration

public LogLevels sdkLogLevel { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
LogLevels	

${\bf sdkLogToMemory}$

To override whether the SDK component performs in-memory logging or not. Default value: false.

Declaration

public bool sdkLogToMemory { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
System.Boolean	

sdkLogToStderr

To override whether the SDK component logs to stderr or not. Default value: false.

Declaration

public bool sdkLogToStderr { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

sigmaAudioLogFileName

To override SigmaAudio log file name

Declaration

public string sigmaAudioLogFileName { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
System.String	

sigmaAudioLogLevel

To override SigmaAudio log level

Declaration

public LogLevels sigmaAudioLogLevel { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
LogLevels	

sigmaAudioLogToMemory

To override whether the SigmaAudio component performs in-memory logging or not. Default value: false.

Declaration

public bool sigmaAudioLogToMemory { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

sigmaAudioLogToStderr

To override whether the SigmaAudio component logs to stderr or not. Default value: false.

Declaration

public bool sigmaAudioLogToStderr { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

GetDefaultLoggingConfiguration()

Static getter for a Logging configuration configured with default values:

- Debug log level
- Dump logs into file clearvr.tmlog
- Dump interaction recording into recorder.tmerp (recorder.tmerj is used for debug) > [!NOTE] >
 Can not be called statically or from a constructor.

Declaration

public static LoggingConfiguration GetDefaultLoggingConfiguration()

Returns

TYPE	DESCRIPTION
LoggingConfiguration	A LoggingConfiguration instance with default values

GetDefaultLoggingFolder()

Declaration

public static String GetDefaultLoggingFolder()

Returns

ТҮРЕ	DESCRIPTION
String	

ToString()

Declaration

public override String ToString()

Returns

ТҮРЕ	DESCRIPTION
String	

Enum LogLevels

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum LogLevels : int	
-----------------------------	--

Fields

NAME	DESCRIPTION
Debug	
Error	
Fatal	
Info	
Warn	

Interface MediaControllerInterface

The controller interface is used to control media playback. APIs like Pause(), Unpause() and Seek() are all part of this interface.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public interface MediaControllerInterface

Methods

GetAudioGain()

Retrieve current audio gain. Notes: Returns 0 if audio gain could not be queried.

Declaration

float GetAudioGain()

Returns

ТҮРЕ	DESCRIPTION
System.Single	The current audio gain.

GetAudioTrack()

Get the currently selected audio track index.

Declaration

int GetAudioTrack()

Returns

TYPE	DESCRIPTION
System.Int32	=0 if an audio track is selected, -1 if audio is explicitely disabled, -2 is no audio track is (yet) selected, -3 if it was impossible to retrieve the selected audio track (should be treated as an error).

GetCurrentContentTimeInMilliseconds()

Get the current content position in milliseconds.

Declaration

long GetCurrentContentTimeInMilliseconds()

Returns

TYPE	DESCRIPTION
System.Int64	The current position in the content, in milliseconds.

GetCurrentWallclockContentTimeInMilliseconds()

Get the current wallclock content position in milliseconds. This API is especially useful when playing back live, synchronized, content.

Declaration

long GetCurrentWallclockContentTimeInMilliseconds()

Returns

TYPE	DESCRIPTION
System.Int64	The current wallclock content position in the content, in milliseconds.

GetIsAudioMuted()

Check whether audio is muted or not.

Declaration

bool GetIsAudioMuted()

Returns

TYPE	DESCRIPTION	
System.Boolean	True is audio is muted, false otherwise.	

GetPlaybackRate()

The rate at which the media is being played back.

Declaration

float GetPlaybackRate()

Returns

TYPE	DESCRIPTION
System.Single	the playback rate as Float value. Returns 0.0f if playback rate can't be queried.

GetTimingReport(TimingTypes)

Gets a comprehensive overview of the current content position and the seekable range.

This API is particularly convenient when playing a live event and one wants to determine the seekable range of that live event and compare it to the current position. One can use one of the following TimingTypes to determine how to interpret the returned values.

- ContentTime query timing properties in reference to the ContentTime. Applicable to VOD and LIVE.
- WallclockTime query timing properties in reference to the ContentTime. Applicable to LIVE content only.
- RelativeTime not applicable, do not use.
- LiveEdge not applicable, do not use.
- Seamless not applicable, do not use.

The TimingReport's getIsSucces() method must be used to determine if the query was successfully handled or not. In case of no success, the clearVRMessage field can be used to query details on the failure.

Declaration

TimingReport GetTimingReport(TimingTypes argTimingType)

Parameters

ТҮРЕ	NAME	DESCRIPTION
TimingTypes	argTimingType	The TimingType, as defined above.

Returns

TYPE	DESCRIPTION
TimingReport	a TimingReport in case of success AND failure. This method never returns null, so in case of failure the TimingReport will contain undefined values

GetTimingReport(System.Object)

Gets a comprehensive overview of the current content position and the seekable range.

This API is particularly convenient when playing a live event and one wants to determine the seekable range of that live event and compare it to the current position. One can use one of the following {@link SeekFlags} to determine how to interpret the returned values, but one can NEVER use a combination of {@link SeekFlags}. SeekFlags.ContentTime: query timing properties in reference to the ContentTime. Applicable to VOD and LIVE. SeekFlags.WallclockTime: query timing properties in reference to the ContentTime. Applicable to LIVE content only. SeekFlags.RelativeTime: not applicable, do not use. SeekFlags.LiveEdge: not applicable, do not use. SeekFlags.Seamless: not applicable, do not use.

The TimingReport's getIsSucces() method can be used to determine if the query was successfully handled or not. In case of no success, the clearVRMessage field can be used to query details on the failure.

Declaration

TimingReport GetTimingReport(System.Object argSeekFlag)

Parameters

TYPE	NAME	DESCRIPTION
System.Object	argSeekFlag	A SINGLE SeekFlag as defined above.

Returns

TYPE	DESCRIPTION
TimingReport	a TimingReport in case of success AND failure. This method never returns null.

Pause(Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Pause playback.

6 NOTE

Due to the nature of spatio-temporal video streaming, pausing is only possible on a GOP boundary and might thus take a short while. Callback event: StateChangedPausing --> StateChangedPaused

Declaration

void Pause(Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] optionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onSuccess	
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	
System.Object[]	optionalArguments	

Pause(Action<ClearVREvent, ClearVRPlayer>, Object[])

Declaration

void Pause(Action<ClearVREvent, ClearVRPlayer> argCbClearVRAsyncRequestResponseReceived, params
object[] argOptionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	argCbClearVRAsyncRequestResponseReceived	
System.Object[]	argOptionalArguments	

Seek(SeekParameters, Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Seek to a new position in the content. The new position should be provided in milliseconds and will be bound between 0 and the content duration. Notes:

- 1. Due to the nature of spatio-temporal streaming, seek will jump to the closest I-frame.
- 2. This method can only be called in Running/Pausing/Paused/Buffering state.

Declaration

void Seek(SeekParameters seekParameters, Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] optionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
SeekParameters	seekParameters	This object specifies the arguments that should be passed to the seek request.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onSuccess	An optional callback that should be triggered after the request was successfully completed. You are highly encouraged to implement the callback, but it can be null.
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	onFailure	An optional callback that should be triggered in case the request has failed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	optionalArguments	Any optional argument that you would like to pass inside the callback.

Seek(SeekParameters, Action<ClearVREvent, ClearVRPlayer>, Object[])

Seek to a new position in the content. The new position should be provided in milliseconds and will be bound between 0 and the content duration. Notes:

- 1. Due to the nature of spatio-temporal streaming, seek will jump to the closest I-frame.
- 2. This method can only be called in Running/Pausing/Paused/Buffering state.

Declaration

void Seek(SeekParameters argSeekParameters, Action<ClearVREvent, ClearVRPlayer> argCbClearVRAsyncRequestResponseReceived, params object[] argOptionalArguments)

TYPE NAME DESCRIPTION	
-----------------------	--

TYPE	NAME	DESCRIPTION
SeekParameters	argSeekParameters	This object specifies the arguments that should be passed to the seek request.
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	argCbClearVRAsyncRequestResponseReceived	An optional callback that should be triggered after the request completed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	argOptionalArguments	Any optional argument that you would like to pass inside the callback.

SetAudioGain(Single)

Change audio gain. Expected scale: [0, 1]

Declaration

void SetAudioGain(float argGain)

Parameters

TYPE	NAME	DESCRIPTION
System.Single	argGain	The new gain to set.

SetAudioTrack(AudioTrackAndPlaybackParameters, Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Sets the active audio track index. Use -1 to select no audio track (effectively disabling audio playback). Note that this call is asynchronous and one must wait for the AudioTrackSwitched event before the switch is done. Callback event: AudioTrackSwitched

Declaration

void SetAudioTrack(AudioTrackAndPlaybackParameters audioTrackAndPlaybackParameters,
Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params
object[] optionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
AudioTrackAndPlaybackParameters	audioTrackAndPlaybackParameters	What audio track to select, and vaudio decoder/playback engine use. When unsure, just use an c from the default constructor new AudioTrackAndPlaybackParame
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onSuccess	An optional callback that should triggered after the request was successfully completed. You are encouraged to implement the callback, but it can be null.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	An optional callback that should triggered in case the request has failed. You are highly encourage implement the callback, but it canull.
System.Object[]	optionalArguments	Any optional argument that you like to pass along.

Set Audio Track (Audio Track And Playback Parameters, Action < Clear VREvent, Clear VRP layer >, Object [])

Sets the active audio track index. Use -1 to select no audio track (effectively disabling audio playback). Note that this call is asynchronous and one must wait for the AudioTrackSwitched event before the switch is done. Callback event: AudioTrackSwitched

Declaration

void SetAudioTrack(AudioTrackAndPlaybackParameters argAudioTrackAndPlaybackParameters,
Action<ClearVREvent, ClearVRPlayer> argCbClearVRAsyncRequestResponseReceived, params object[]
argOptionalArguments)

ТҮРЕ	NAME	DESCRIPTION
AudioTrackAndPlaybackParameters	argAudioTrackAndPlaybackParameters	What audio track to audio decoder/play use. When unsure, from the default con AudioTrackAndPlay
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	argCbClearVRAsyncRequestResponseReceived	Optional callback the triggered upon cominull
System.Object[]	argOptionalArguments	Any optional argum like to pass along.

SetLoopContent(Boolean)

Enable or disable content looping. Note that you can only call this API after the player has reached StateChangedPlaying. If you would like to configure content looping before loading the first ContentItem, please refer to platformOptions.loopContent.

Declaration

bool SetLoopContent(bool argIsContentLoopEnabled)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	argIsContentLoopEnabled	

Returns

TYPE	DESCRIPTION
System.Boolean	

SetMuteAudio(Boolean)

Mute or unmute audio playback. Check the specification of the return type in the notes below!

Notes.

1. this method returns false if it successfully UNMUTED the audio OR failed to execute the mute/unmute request.

Declaration

bool SetMuteAudio(bool argIsMuted)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	argIsMuted	True if audio should be muted, false is audio should be unmuted.

Returns

TYPE	DESCRIPTION
System.Boolean	True if audio is muted, false if audio is unmuted OR if the call failed.

SetPlaybackRate(Single)

Set the rate at which the media is being played back.

Notes: Only values between 0.5 and 2.0 are valid.

Declaration

void SetPlaybackRate(float value)

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	the new playback rate

StartPlayout(Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

After initialization and content preparation, use this method to actually commence playback. Typically, playback should start more or less immediately after calling this method as video is supposed to be cached by then.

A WARNING

During the lifecycle of a ClearVRPlayer object one can only call this method once. To unpause playback, use Unpause() instead. Callback event: StateChangedPlaying

Declaration

void StartPlayout(Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] optionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onSuccess	
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	
System.Object[]	optionalArguments	

StartPlayout(Action<ClearVREvent, ClearVRPlayer>, Object[])

Declaration

void StartPlayout(Action<ClearVREvent, ClearVRPlayer> argCbClearVRAsyncRequestResponseReceived, params object[] argOptionalArguments)

Parameters

ТҮРЕ	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	argCbClearVRAsyncRequestResponseReceived	
System.Object[]	argOptionalArguments	

Stop()

Declaration

ClearVRAsyncRequest Stop()

Returns

TYPE	DESCRIPTION
ClearVRAsyncRequest	

Stop(Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Stop the player. One MUST wait for the StateChangedStopped and MUST clean-up the ClearVRPlayer object once this callback is received OR (preferable) wait for the StateChangedStopped event. Please note that the player can stop itself for whatever reason at whatever moment in time. In this case, one will have to perform final clean-up in StateChangedStopped anyway. Callback event: StateChangedStopped

Declaration

void Stop(Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] optionalArguments)

Parameters

ТҮРЕ	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onSuccess	
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	
System.Object[]	optionalArguments	

Stop(Action<ClearVREvent, ClearVRPlayer>, Object[])

Declaration

void Stop(Action<ClearVREvent, ClearVRPlayer> argCbClearVRAsyncRequestResponseReceived, params object[] argOptionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	argCbClearVRAsyncRequestResponseReceived	
System.Object[]	argOptionalArguments	

SwitchContent(SwitchContentParameters, Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

This API is used to switch to a new ContentItem.

Declaration

void SwitchContent(SwitchContentParameters switchContentParameters, Action<ClearVREvent,
ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[]
optionalArguments)

Parameters

ТҮРЕ	NAME	DESCRIPTION
SwitchContentParameters	switchContentParameters	Specifies required parameters to switch content.
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	onSuccess	An optional callback that should be triggered after the request was succesfully completed. You are highly encouraged to implement the callback, but it can be null.
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	onFailure	An optional callback that should be triggered in case the request has failed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	optionalArguments	Any optional argument that you would like to pass inside the callback.

SwitchContent(SwitchContentParameters, Action<ClearVREvent, ClearVRPlayer>, Object[])

This API is used to switch to a new ContentItem.

Declaration

void SwitchContent(SwitchContentParameters argSwitchContentParameters, Action<ClearVREvent,
ClearVRPlayer> argCbClearVRAsyncRequestResponseReceived, params object[] argOptionalArguments)

TYPE	NAME	DESCRIPTION
SwitchContentParameters	argSwitchContentParameters	Specifies required parameters to switch content.

TYPE	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	argCbClearVRAsyncRequestResponseReceived	An optional callback that should be triggered after the request completed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	argOptionalArguments	Any optional argument that you would like to pass inside the callback.

TogglePause(Action<ClearVREvent, ClearVRPlayer>, Object[])

Convenience method to easily toggle between pause and unpause. See Pause() and Unpause() for details.

Declaration

void TogglePause(Action<ClearVREvent, ClearVRPlayer> argCbClearVRAsyncRequestResponseReceived, params object[] argOptionalArguments)

Parameters

ТҮРЕ	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	argCbClearVRAsyncRequestResponseReceived	
System.Object[]	argOptionalArguments	

Unpause(Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Unpause/resume playback. Callback event: StateChangedPlaying

Declaration

void Unpause(Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] optionalArguments)

TYPE	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onSuccess	
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	
System.Object[]	optionalArguments	

Unpause(Action<ClearVREvent, ClearVRPlayer>, Object[])

Declaration

void Unpause(Action<ClearVREvent, ClearVRPlayer> argCbClearVRAsyncRequestResponseReceived, params
object[] argOptionalArguments)

Parameters

ТҮРЕ	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	argCbClearVRAsyncRequestResponseReceived	
System.Object[]	argOptionalArguments	

Unpause(TimingParameters, Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Since v9.1 Unpause/resume playback. This API allows you to specify where to resume playback by setting the argTimingParameters argument.

Declaration

void Unpause(TimingParameters argTimingParameters, Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] optionalArguments)

ТҮРЕ	NAME	DESCRIPTION
TimingParameters	argTimingParameters	Set to null for default behaviour: VOD = resume from last played position, LIVE = resume from last played position unless that fell out of the live window. In the latter case, playback will resume from the live edge.

TYPE	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onSuccess	Callback triggered when the request was successfully serviced.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	Callback triggered when the request failed to complete.
System.Object[]	optionalArguments	Any optional argument you would like to receive in the onSuccess and onFailure callback.

Interface MediaInfoInterface

This interface specifies various convenience methods that allws quick access to the most important media stream properties.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public interface MediaInfoInterface

Methods

GetContentDurationInMilliseconds()

Query the content duration. Notes:

- 1. for VOD content, this method returns the duration of the clip
- 2. for live content, this method returns 0
- 3. if called at an illegal moment (e.g. when no content has yet been loaded), this method returns 0
- 4. this is considered an "expensive" method, one is encouraged to call this method too often, but rather cache its value.

Declaration

long GetContentDurationInMilliseconds()

Returns

TYPE	DESCRIPTION
System.Int64	The content duration in milliseconds, or 0 otherwise (see notes)

GetContentFormat()

Returns the content format. Can be used to figure out whether the content is 360 or 180 degrees and if it is monoscopic or stereoscopic. Notes.

1. if, for whatever reason, the content format changes, the callback event ContentFormatChanged will be invoked.

Declaration

ContentFormat GetContentFormat()

Returns

TYPE	DESCRIPTION
ContentFormat	The current content format.

GetContentInfo(ContentItem)

Get information (e.g. the resolution of available ABR representations, number of audio tracks, etc.) about the specified contentItem and all the feeds it contains. This information can only be fetched after the ContentPreparedForPlayout state with a ContentItem that is known (is prepared/playing right now, or has been prepared/played by using the SwitchContent API)

Declaration

ContentInfo GetContentInfo(ContentItem contentItem = null)

Parameters

TYPE	NAME	DESCRIPTION
ContentItem	contentitem	The content item you would like to retrieve this information from, can be null. If set to null, the currently active content item will be queried.

Returns

TYPE	DESCRIPTION
ContentInfo	a ContentInfo object, which contains one or multiple FeedInfo which in turn contains VideoTrackInfo, AudioTrackInfo and SubtitleTrackInfo (if available). This API returns null if the call fails or a ContentInfo object with 0 feeds if the queried content item is not loaded/active.

GetCurrentResolutionAndFramerate()

Returns the resolution and framerate of the currently active representaion as a formatted string: [width]x[height]p[framerate] or 0x0p0 if unknown.

Declaration

String GetCurrentResolutionAndFramerate()

Returns

TYPE	DESCRIPTION
String	The current quality as a formatted string.

GetEventType()

Query the event type of the currently active content item. Note that this might change after every SwitchContent(), so one is encouraged to requery this parameter in the ClearVREventTypes.ContentSwitched event.

Declaration

EventTypes GetEventType()	
---------------------------	--

Returns

TYPE	DESCRIPTION
EventTypes	Returns the appropriate EventTypes enum.

GetHighestQualityResolutionAndFramerate()

Returns the highest available resolution and framerate as a formatted string: [width]x[height]p[framerate] or 0x0p0 if unknown. Helpfull if one wants to determine the highest available representation

A WARNING

This API has been deprecated and removed from the SDK. Monitor the ActiveTracksChanged event instead. The video tracks per feed in the content info are ordered in descending order, highest quality first.

Declaration

String GetHighestQualityResolutionAndFramerate()

Returns

TYPE	DESCRIPTION
String	The highest available quality as a formatted string.

GetIsContentFormatStereoscopic()

Convenience method that returns whether the current content is stereoscopic or not.

Declaration

bool GetIsContentFormatStereoscopic()

Returns

TYPE	DESCRIPTION
System.Boolean	True if the content is stereoscopic, false if monoscopic.

GetNumberOfAudioTracks()

Returns the number of audio tracks Notes:

1. Returns 0 even if the number of audio tracks could not be queried. > [!NOTE] > This API has been removed in v9.x. Instead, refer to the ActiveTracksChanged event and its ContentInfo instead.

Declaration

int GetNumberOfAudioTracks()

Returns

ТҮРЕ	DESCRIPTION
System.Int32	The number of available audio tracks.

Interface MediaPlayerInterface

The MediaPlayer interface specifies interactions with the MediaPlayer object.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public interface MediaPlayerInterface

Methods

EnableOrDisableViewportTracking(Boolean)

Since v4.1.2 Enable or disable viewport tracking. This is an advanced API that allows you to disable viewport tracking, effectively fixating the high quality viewport regardless of camera orientation. By default, viewport tracking is enabled.

Notes.

1. This is an advanced API, typicaly only used in exceptional cases.

Declaration

void EnableOrDisableViewportTracking(bool argIsEnabledOrDisabled)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	argIsEnabledOrDisabled	True to enable tracking, false otherwise.

GetCanPerformanceMetricesBeQueried()

Convenience method which helps check whether one can query performance metrics or not.

Declaration

bool GetCanPerformanceMetricesBeQueried()

Returns

TYPE	DESCRIPTION
System.Boolean	True if metrics can be queried, false otherwise.

GetClearVRCoreArrayParameter(String, Int32)

Since v4.3

The ClearVRCore exposes a large number of parameters which can be queried when the player is in a certain state. Please refer to the ClearVRCore documentation for an exhaustive list of all parameters. This method can be used to query parameters that take a mandatory array index.

Notes.

- 1. Some common day-to-day-use parameters are exposed through convenience methods.
- 2. Note that many parameters can only be queried at a specific moment in time.
- 3. Most parameters do not take an array index. See GetClearVRCoreParameter() and the ClearVRCore documentation for details.
- 4. One needs to make sure that your query is properly bounded.

Declaration

String GetClearVRCoreArrayParameter(String argKey, int argIndex)

Parameters

TYPE	NAME	DESCRIPTION
String	argKey	The key to query.
System.Int32	argIndex	The index in the array to query.

Returns

TYPE	DESCRIPTION
String	The value of the queried key, or an empty string if it could not be queried.

GetClearVRCoreParameter(String)

The ClearVRCore exposes a large number of parameters which can be queried when the player is in a certain state. Please refer to the ClearVRCore documentation for an exhaustive list of all parameters. Notes.

- 1. Some common day-to-day-use parameters are exposed through convenience methods.
- 2. Note that many parameters can only be queried at a specific moment in time.
- 3. Some parameters can only be queried with an array index. See

GetClearVRCoreArrayParameter() and the ClearVRCore documentation for details.

Declaration

String GetClearVRCoreParameter(String argKey)

Parameters

TYPE	NAME	DESCRIPTION
String	argKey	The key to query.

Returns

TYPE	DESCRIPTION
String	The value of the queried key, or an empty string if it could not be queried.

GetDefaultViewportPose()

GetDefaultViewportPose return the position and the orientation that the default camera should have. For omnidirectional content this pose is at the origin with no rotation. For planar content this pose should be at the totally unzoomed position with no rotation.

Declaration

Pose GetDefaultViewportPose()

Returns

TYPE	DESCRIPTION
Pose	

GetDeviceAppld()

Returns the unique device app id as a string. The device app is defined as a MD5-hashed combination of a unique user id and your application's id. It will return an empty string if queried in an invalid state, or if the device app id could not be generated. Notes.

1. One is recommended to query the device app id when one receives the ClearVREventTypes.StateChangedCorePrepared event.

Declaration

String GetDeviceAppId()

Returns

TYPE	DESCRIPTION
String	The unique id as a string or an empty string if unavailable.

GetIsInFinishedState()

Convenience method to check whether the player is in Finished state. The player will only reach Finished state in case it reached the end of the clip AND content looping is disabled. Notes.

1. When the player reaches this state, the viewport can no longer be updated and you cannot switch to Pause state. You can call Seek() though and playback will resume immediately if you would do so.

Since: v4.1.2

Declaration

bool GetIsInFinishedState()

Returns

TYPE	DESCRIPTION
System.Boolean	True if in Finished state, false otherwise.

GetIsInitialized()

Whether the media player was initialized or not. Note that "initialized" does not neccessarily mean that it is already playing content.

Declaration

bool GetIsInitialized()

Returns

ТҮРЕ	DESCRIPTION
System.Boolean	True if the underlying media player is at least initialized, false otherwise.

GetIsInPausedState()

Convenience method to check whether the player is in paused state.

Declaration

bool GetIsInPausedState()

Returns

TYPE	DESCRIPTION
System.Boolean	True if in paused state, false otherwise.

GetIsInPausingOrPausedState()

Convenience method to check whether the player is in pausing/paused state.

Declaration

bool GetIsInPausingOrPausedState()

Returns

TYPE	DESCRIPTION
System.Boolean	True if in pausing or paused state, false otherwise.

GetIsInPlayingState()

Convenience method to check whether the player is in playing state.

Declaration

bool GetIsInPlayingState()

Returns

TYPE	DESCRIPTION
System.Boolean	True if in playing state, false if in any other state.

GetIsInStoppedState()

Convenience method to check whether the player is in Stopped state. When in Stopped state, the player can no longer be used and should be destroyed. Notes.

1. Before the Stopped state, the mediaPlayer will first transition through the Stopping state. This is a transient state and one MUST wait for it to complete before destroying the mediaPlayer. Failing to do so will result in unexpected behaviour and resources might leak.

Since: v4.1.2

Declaration

bool GetIsInStoppedState()

Returns

TYPE	DESCRIPTION
System.Boolean	True if in Stopped state, false otherwise.

GetIsMediaInfoParsed()

Advanced API to determine whether media info was parsed or not.

Declaration

bool GetIsMediaInfoParsed()

Returns

TYPE	DESCRIPTION
System.Boolean	True if media info was parsed, false otherwise.

GetIsPlayerBusy()

Whether the player is busy playing a content item.

Declaration

bool GetIsPlayerBusy()

Returns

ТҮРЕ	DESCRIPTION
System.Boolean	True if busy, false otherwise.

GetPlatformOptions()

Returns the currently active platform options. Note that you can only set these prior to initialization. Changing any of its parameters after initialization results in undefined baheviour and must be avoided.

Declaration

PlatformOptionsBase GetPlatformOptions()

Returns

TYPE	DESCRIPTION
PlatformOptionsBase	The currently active PlatformOptions. Cast to the platform-specific PlatformOptions to access any platform-specific fields.

GetRecommendedZoomRange(out Single, out Single)

GetRecommendedZoomRange return the recommanded zoom range. min correspond to the maximal zoom, closest to the display object, and max correspond to the minimal zoom.



This API has been deprecated in v9.x and can no longer be used. Refer to the LayoutManager instead.

Declaration

void GetRecommendedZoomRange(out float min, out float max)

Parameters

TYPE	NAME	DESCRIPTION
System.Single	min	
System.Single	max	

GetRenderMode()

Gets the currently active RenderMode.

Declaration

RenderModes GetRenderMode()

Returns

TYPE	DESCRIPTION
RenderModes	The active RenderMode.

PopulateMediaInfo(PopulateMediaInfoParameters, Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Populate media info on a specific ContentItem. Note that populating media info does not load the content item and does not prepare it for immediate playout. For that, one needs to Initialize() or SwitchContent() instead. This is an advanced API and should typically not be used.

Declaration

void PopulateMediaInfo(PopulateMediaInfoParameters populateMediaInfoParameters, Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] optionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
PopulateMediaInfoParameters	populateMediaInfoParameters	The parameters required for this API
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onSuccess	An optional callback that should be triggered after the request was succesfully completed. You are highly encouraged to implement the callback, but it can be null.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	An optional callback that should be triggered in case the request has failed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	optionalArguments	Any optional argument to pass along in the provided callback.

PopulateMediaInfo(PopulateMediaInfoParameters, Action<ClearVREvent, ClearVRPlayer>, Object[])

Populate media info on a specific ContentItem. Note that populating media info does not load the content item and does not prepare it for immediate playout. For that, one needs to Initialize() or SwitchContent() instead. This is an advanced API and should typically not be used.

Declaration

void PopulateMediaInfo(PopulateMediaInfoParameters argPopulateMediaInfoParameters, Action<ClearVREvent, ClearVRPlayer> argCbClearVRAsyncRequestResponseReceived, params object[] argOptionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
PopulateMediaInfoParameters	argPopulateMediaInfoParameters	The parameters required for this API
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	argCbClearVRAsyncRequestResponseReceived	The callback to trigger upon completion.
System.Object[]	argOptionalArguments	Any optional argument to pass along in the provided callback.

PrepareContentForPlayout(PrepareContentParameters, Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Prepares a ContentItem for playout. This is an asynchronous call and one must wait for the StateChangedContentPreparedForPlayout callback to check whether it was successful or not. Callback event: StateChangedContentPreparedForPlayout

Declaration

void PrepareContentForPlayout(PrepareContentParameters prepareContentParameters, Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] optionalArguments)

|--|

ТҮРЕ	NAME	DESCRIPTION
PrepareContentParameters	prepareContentParameters	The parameters used when preparing the content for playout.
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	onSuccess	An optional callback that should be triggered after the request was succesfully completed. You are highly encouraged to implement the callback, but it can be null.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	An optional callback that should be triggered in case the request has failed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	optionalArguments	Any optional argument to pass along in the provided callback.

PrepareContentForPlayout(PrepareContentParameters, Action<ClearVREvent, ClearVRPlayer>, Object[])

Prepares a ContentItem for playout. This is an asynchronous call and one must wait for the StateChangedContentPreparedForPlayout callback to check whether it was successful or not. Callback event: StateChangedContentPreparedForPlayout

Declaration

void PrepareContentForPlayout(PrepareContentParameters argPrepareContentParameters,
Action<ClearVREvent, ClearVRPlayer> argCbClearVRAsyncRequestResponseReceived, params object[]
argOptionalArguments)

TYPE	NAME	DESCRIPTION
PrepareContentParameters	argPrepareContentParameters	The parameters used when preparing the content for playout.

TYPE	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	argCbClearVRAsyncRequestResponseReceived	
System.Object[]	argOptionalArguments	

PrewarmCache(PrewarmCacheParameters, Action<ClearVREvent, ClearVRPlayer>, Object[])

The PrewarmCache API allows one to preload initialization data for a specific clip prior to actually using it. Note that the cache is linked to the ClearVRPlayer instance. If you destroy the ClearVRPlayer, the cache will be flushed as well. This is typically used together with the SwitchContent() API.

As each cache warmup will require up to 1 or 2 megabyte of data (varying per content item), please be considerate when to actually call this API as downloading extra data might interfere with regular playback on limited internet connections. For example, paused state might be a good moment to prewarm the cache on an item or two.

Warning: please do NOT hammer this API (e.g. by looping over your entire content list as fast as possible).

Callback event: PrewarmCacheCompleted

Declaration

void PrewarmCache(PrewarmCacheParameters argPrewarmCacheParameter, Action<ClearVREvent, ClearVRPlayer> argCbClearVRAsyncRequestResponseReceived, params object[] argOptionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
PrewarmCacheParameters	argPrewarmCacheParameter	The parameters that should be passed along to prewarm the cache on a specific ContentItem.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	argCbClearVRAsyncRequestResponseReceived	
System.Object[]	argOptionalArguments	

ResetViewportAndDisplayObjectToDefaultPoses()

ResetViewportAndDisplayObjectToDefaultPoses move the display object and the camera to it default pose based on the current content type.



This API has been deprecated in v9.x and can no longer be used. Refer to the LayoutManager instead.

Declaration

void ResetViewportAndDisplayObjectToDefaultPoses()

SetClearVRCoreParameter(String, String)

Set a specific ClearVRCore parameter. See also GetClearVRCoreParameter() for more details.

Declaration

bool SetClearVRCoreParameter(String argKey, String argValue)

Parameters

TYPE	NAME	DESCRIPTION
String	argKey	The key to set.
String	argValue	The value to set the key to.

Returns

TYPE	DESCRIPTION
System.Boolean	

SetLayout(LayoutParameters, Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Since v9.x This API is used to change the feed to display object mapping within a given content. The provided LayoutParameters will overwrite the existing LayoutParameters on the ClearVRLayoutManager, or will be added in case they are not yet present.

Declaration

void SetLayout(LayoutParameters argLayoutParameters, Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] argOptionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
LayoutParameters	argLayoutParameters	Specifies required parameters to change the feeds layout.
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	onSuccess	An optional callback that should be triggered after the request was successfully completed. You are highly encouraged to implement the callback, but it can be null.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	An optional callback that should be triggered in case the request has failed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	argOptionalArguments	Any optional argument that you would like to pass inside the callback.

SetRenderMode(RenderModes)

This API is used to switch between monoscopic and stereoscopic rendering (if the content and player allows for it). Note that this only changes how the content is *rendered*, NOT how it is retrieved from the network. If you want to switch between monoscopic or stereoscopic content retrieval, you should use the SetStereoMode() API instead. It is especially suited for temporarily disabling stereoscopic rendering when you show an in-player menu (to prevent depth-fighting from happening) as this API is instantaneous while the SetStereoMode() API will take a bit of time to respond, reducing user experience. Callback event: RenderModeChanged

Declaration

void SetRenderMode(RenderModes argNewRenderMode)

Parameters

TYPE	NAME	DESCRIPTION
RenderModes	argNewRenderMode	The new render mode to switch to.

SetRenderModeOnAllDisplayObjects(RenderModes)

Since v9.x Convenience API that allows one to set the Render Mode on all active DisplayObjects at (c) Tiledmedia B.V. 2017-2022

once.

Declaration

void SetRenderModeOnAllDisplayObjects(RenderModes argNewRenderMode)

Parameters

TYPE	NAME	DESCRIPTION
RenderModes	argNewRenderMode	The RenderMode to set.

SetRenderModeOnAllDisplayObjectsConditionally(RenderModes, RenderModes)

Since v9.x Convenience API that allows one to set the Render Mode on all active DisplayObjects at once, under the condition that they are in the Render Mode as specified by argRequired Current Render Mode.

Declaration

void SetRenderModeOnAllDisplayObjectsConditionally(RenderModes argNewRenderMode, RenderModes argRequiredCurrentRenderMode)

Parameters

TYPE	NAME	DESCRIPTION
RenderModes	argNewRenderMode	The RenderMode to set.
RenderModes	argRequiredCurrentRenderMode	The RenderMode will only be changed on DisplayObjects that are in this RenderMode.

SetStereoMode(Boolean)

The SetStereoMode() API can be used to switch between monoscopic and stereoscopic playback if the content supports it. Note that this actually toggles the retrieval of the right-eye tiles, so enabling it might take a bit of time and might result in buffering depending on network conditions. If one would only temporarily switch between monoscopic and stereoscopic rendering (e.g. because an in-player menu pops-up and you want to make sure there is no depth-fighting between UI and video), one is advised to use the SetRenderMode() API instead as that only changes how the video rendered, not how it is retrieved from the network. Callback event: StereoModeChanged. Optionally, one will receive the RenderModeChanged event if a switch in rendering was required (e.g. when you went from stereoscopic to monoscopic).

Declaration

ClearVRAsyncRequest SetStereoMode(bool argStereo)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	argStereo	true if you want to switch to stereoscopic rendering, false if monoscopic rendering is requested.

Returns

ТҮРЕ	DESCRIPTION
ClearVRAsyncRequest	

SetStereoMode(Boolean, Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

The SetStereoMode() API can be used to switch between monoscopic and stereoscopic playback if the content supports it. Note that this actually toggles the retrieval of the right-eye tiles, so enabling it might take a bit of time and might result in buffering depending on network conditions. If one would only temporarily switch between monoscopic and stereoscopic rendering (e.g. because an in-player menu pops-up and you want to make sure there is no depth-fighting between UI and video), one is advised to use the SetRenderMode() API instead as that only changes how the video rendered, not how it is retrieved from the network. Callback event: StereoModeChanged. Optionally, one will receive the RenderModeChanged event if a switch in rendering was required (e.g. when you went from stereoscopic to monoscopic).

O NOTE

This is considered an advanced API and should be used with care. Also, it only applies to ClearVR content playback.

Declaration

void SetStereoMode(bool stereo, Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] optionalArguments)

TYPE	NAME	DESCRIPTION
System.Boolean	stereo	true if you want to switch to stereoscopic rendering, false if monoscopic rendering is requested.
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	onSuccess	An optional callback that should be triggered after the request was successfully completed. You are highly encouraged to implement the callback, but it can be null.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	An optional callback that should be triggered in case the request has failed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	optionalArguments	Any optional argument to pass along in the provided callback.

SetStereoMode(Boolean, Action<ClearVREvent, ClearVRPlayer>, Object[])

The SetStereoMode() API can be used to switch between monoscopic and stereoscopic playback if the content supports it. Note that this actually toggles the retrieval of the right-eye tiles, so enabling it might take a bit of time and might result in buffering depending on network conditions. If one would only temporarily switch between monoscopic and stereoscopic rendering (e.g. because an in-player menu pops-up and you want to make sure there is no depth-fighting between UI and video), one is advised to use the SetRenderMode() API instead as that only changes how the video rendered, not how it is retrieved from the network. Callback event: StereoModeChanged. Optionally, one will receive the RenderModeChanged event if a switch in rendering was required (e.g. when you went from stereoscopic to monoscopic).

Declaration

void SetStereoMode(bool argStereo, Action<ClearVREvent, ClearVRPlayer>
argCbClearVRAsyncRequestResponseReceived, params object[] argOptionalArguments)

TYPE	NAME	DESCRIPTION
------	------	-------------

TYPE	NAME	DESCRIPTION
System.Boolean	argStereo	true if you want to switch to stereoscopic rendering, false if monoscopic rendering is requested.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	argCbClearVRAsyncRequestResponseReceived	
System.Object[]	argOptionalArguments	

TelemetryUpdateCustomData(TelemetryUpdateCustomData, Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Since v9.2 Allows you send custom key/value pairs to the Telemetry target(s) of choice. Note that the Telemetry target(s) must have been configured a-priori on platformOptions.telemetryConfiguration or on the (rarely used and advanced API) ContentSupportedTesterParameters. Example:

```
clearVRPlayer.mediaPlayer.TelemetryUpdateCustomData(new TelemetryUpdateCustomData(new
List<TelemetryUpdateTargetCustomData>() {
    new TelemetryUpdateTargetCustomData(0, new List<KeyValuePair<string, string>>() {
        new KeyValuePair<string, string>("Hello", "This is a test")
        })}),
        onSuccess: (clearVREvent, clearVRPlayer) => UnityEngine.Debug.Log("Telemetry custom data update:
success"),
        onFailure: (clearVREvent, clearVRPlayer) => UnityEngine.Debug.Log(String.Format("Telemetry custom data update: failed. Details: {0}", clearVREvent))
        );
```

Declaration

void TelemetryUpdateCustomData(TelemetryUpdateCustomData telemetryUpdateCustomData,
Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params
object[] optionalArguments)

ТҮРЕ	NAME	DESCRIPTION
TelemetryUpdateCustomData	telemetryUpdateCustomData	

ТҮРЕ	NAME	DESCRIPTION
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	onSuccess	An optional callback that should be triggered after the request was succesfully completed. You are highly encouraged to implement the callback, but it can be null.
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	onFailure	An optional callback that should be triggered in case the request has failed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	optionalArguments	

UpdateOverrideUserAgent(String)

This API allows you to update the user agent included in every HTTP request. One is strongly discouraged to setting a user agent in the first place because of network overhead. The user agent is a global setting, impacting all HTTP requests from that point onward. For example, if one would configure a user agent "ABC" on the ContentSupportedTesterParameters, but would call this API to change it, than the user agent might change while the content supported tester is running. To set a user agent for the TestIsContentSupported API, refer to ContentSupportedTesterParameters. To set a user agent for regular playback, refer to PlatformOptionsbase.overrideUserAgent. Since v7.3.2

Declaration

void UpdateOverrideUserAgent(String argNewUserAgent)

TYPE	NAME	DESCRIPTION
String	argNewUserAgent	The new user agent to include in each HTTP request.

Enum MeshTextureModes

An internal enum that is not of any use to the integrator.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum MeshTextureModes: int

Fields

NAME	DESCRIPTION
OVROverlay	OVR overlay texture rendering mode. Only available on Android
Sprite	Sprite texture rendering mode.
Unknown	Unknown is a forbidden value
UnmanagedMesh	Unmanaged mesh rendering mode. The customer has full control over the mesh. This mode does not support ClearVR content playback.
UVShuffling	Normal texture rendering mode. The ClearVRPlayer is in full control over the mesh and its updates.

Class OVROverlayOptions

Holds OVROverlay playback specific settings.

Inheritance

System.Object

OVROverlayOptions

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class OVROverlayOptions : object

Fields

isOverlay

Configure whether the video should be rendered as an Overlay (true) or Underlay (false). As Underlay mode has a severe performance penalty, Overlay is strongly recommended. Please refer to the documentation at https://developer.oculus.com/documentation/unity/unity-ovroverlay/#understanding-ovroverlay-script-configurations for details.

Default value: true



You can change this value on the fly by setting ClearVRDisplayObjectControllerOVROverlay.isOverlay

Declaration

public bool isOverlay

Field Value

ТҮРЕ	DESCRIPTION
System.Boolean	

noDepthBufferTesting

Disable depth buffer testing on video OVROverlay. Default value: false Notes:

1. it is strongly recommended to not change this default value.

Declaration

public bool noDepthBufferTesting

Field Value

TYPE	DESCRIPTION
System.Boolean	

reservedIndices

Specify the preferred reserved OVROverlay indices. By default, the last two allowed indices (index 13 and 14 (0-based)) are selected. If you want to modify the composition depth, one should set videoCompositionDepth.

Notes:

1. You MUST make sure that your application will NEVER use these two OVROverlay layer indices. In practice, this means that you are limited to using up to 12 OVROverlays yourself.

Declaration

public Int32[] reservedIndices

Field Value

TYPE	DESCRIPTION
Int32[]	

videoCompositionDepth

Specifies the composition depth of the video texture.

Default value: 0.

Declaration

public int videoCompositionDepth

Field Value

TYPE	DESCRIPTION
System.Int32	

Interface PerformanceInterface

The performance interface exposes various methods to query the performance of the ClearVRCoreWrapper and the underlying ClearVRCore. Note that the ClearVRCore exposes many more metrics, refer to the ClearVRCore documentation and the "perf.*" parameter keys for details.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public interface PerformanceInterface

Properties

clearVRCoreWrapperStatistics

The ClearVRCoreWrapper class exposes various runtime performance metrics. Use this object to access them. Note that this *must* be destroyed in order to not leak any memory. Typically, this is being taken care of by the underlying layers. You typically get a reference to this object once for every ClearVRPlayer object you instantiate. The object stays valid thourhgout the entire lifetime of said object, but will be destroyed if the underlying ClearVRPlayer object is destroyed.

Declaration

StatisticsBase clearVRCoreWrapperStatistics { get; }

Property Value

TYPE	DESCRIPTION
StatisticsBase	A ClearVRCoreWrapperStatistics object that can be queried for metrics.

Methods

GetAverageBitrateInMbps()

Convience method exposing the ClearVRCore "perf.network.current_avg_kbps" metric. Notes.

1. This is considered an "expensive" operation and one is encouraged *not* to call this method every frame.

Declaration

float GetAverageBitrateInMbps()

Returns

TYPE	DESCRIPTION
System.Single	The current network throughput in megabit per second.

Class PlatformOptionsAndroid

Android platform specific platform options. Currently, there are NO additional options/fields to set.

Inheritance

System.Object

PlatformOptionsBase

PlatformOptionsAndroid

Implements

PlatformOptionsInterface

Inherited Members

PlatformOptionsBase.licenseFileBytes

PlatformOptionsBase.isVRDevicePresent

PlatformOptionsBase.parentGameObject

PlatformOptionsBase.preferredRenderMode

PlatformOptionsBase.autoPlay

PlatformOptionsBase.loopContent

PlatformOptionsBase.autoPrepareContentItem

PlatformOptionsBase.audioPlaybackEngine

PlatformOptionsBase.audioDecoder

PlatformOptionsBase.applicationFocusAndPauseHandling

PlatformOptionsBase.audioFocusChangedHandlingType

PlatformOptionsBase.enableAutomaticRenderModeSwitching

PlatformOptionsBase.screenWidth

PlatformOptionsBase.screenHeight

PlatformOptionsBase.deviceType

PlatformOptionsBase.deviceParameters

PlatformOptionsBase.clearVRCoreVerbosity

PlatformOptionsBase.clearVRCoreLogToFile

PlatformOptionsBase.enableAutomaticRenderModeSwitchingOnContentFormatChanged

PlatformOptionsBase.initializeFlags

Platform Options Base. prepare Content For Playout Time out In Millise conds

PlatformOptionsBase.isTransparencySupportEnabled

PlatformOptionsBase.trackingTransform

PlatformOptionsBase.httpProxyParameters

PlatformOptionsBase.httpsProxyParameters

PlatformOptionsBase.renderCamera

PlatformOptionsBase.textureBlitMode

PlatformOptionsBase.overrideColorSpace

PlatformOptionsBase.vrAPIType

PlatformOptionsBase.ovrOverlayOptions

(c) Tiledmedia B.V. 2017-2022

PlatformOptionsBase.contentProtectionRobustnessLevel

PlatformOptionsBase.cameraAndContentPlacementMode

PlatformOptionsBase.overrideUserAgent

PlatformOptionsBase.initialAudioGain

PlatformOptionsBase.enableABR

PlatformOptionsBase.abrStartMode

PlatformOptionsBase.allowDecoderContraintsInducedStereoToMono

PlatformOptionsBase.prepareContentParameters

PlatformOptionsBase.telemetryConfiguration

PlatformOptionsBase.Print()

PlatformOptionsBase.ToString()

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class PlatformOptionsAndroid: PlatformOptionsBase, PlatformOptionsInterface

Fields

isUseOESFastPathEnabled

Enables OES Fast Path mode. Video frames are directly streamed from the decoder onto the mesh, skipping one texture copy. An important limitation is that this mode does not support picture-in-picture.

Declaration

public bool isUseOESFastPathEnabled

Field Value

TYPE	DESCRIPTION
System.Boolean	

platform

Declaration

public RuntimePlatform platform

Field Value

ТҮРЕ	DESCRIPTION
RuntimePlatform	

Methods

Verify(ClearVRLayoutManager)

Verifies if Android platform specific options are valid or not.

Declaration

public override bool Verify(ClearVRLayoutManager argClearVRLayoutManager)

Parameters

TYPE	NAME	DESCRIPTION
ClearVRLayoutManager	argClearVRLayoutManager	

Returns

TYPE	DESCRIPTION
System.Boolean	true if everything is OK, false otherwise.

Overrides

PlatformOptionsBase.Verify(ClearVRLayoutManager)

Implements

PlatformOptionsInterface

Class PlatformOptionsBase

Configuring your ClearVRPlayer object can be achieved using the platformOptions class.

A WARNING

Reading or changing any field *after* calling ClearVRPlayer.Initialize() is not allowed and can result in undefined behaviour. The only exception to this rule is when using the **applicationRegainedFocusDelegate** delegate to customize player behaviour during an application suspend/resume cycle.

Inheritance

System.Object

PlatformOptionsBase

PlatformOptionsAndroid

PlatformOptionsIOS

PlatformOptionsPC

Implements

PlatformOptionsInterface

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class PlatformOptionsBase : object, PlatformOptionsInterface

Fields

abrStartMode

Define the start mode of the ABR algorithm. SeeABRStartModes for details. Default value: Default

Declaration

public ABRStartModes abrStartMode

Field Value

TYPE	DESCRIPTION
ABRStartModes	

allowDecoderContraintsInducedStereoToMono

In some cases, a stereoscopic clip cannot be rendered stereoscopic due to insufficient decoder capabilities. For example, playing a 12K stereoscopic clip requires a hardware decoder that can

decode at least 6K. If you would've tried to play this clip on a 4K decoder, playback would simply fail. If you enable this option (e.g. set it to true), the player will attempt to render the clip in monoscopic in case insufficient decoder capacity is detected for stereoscopic playback, permitting that the decoder has sufficient capacity to do so. Default value: false

A WARNING

Note that this can only be configured once, either when calling the ClearVRPlayer.TestIsContentSupported() API or clearVRPlayer.Initialize(), whichever comes first. From that point onward, it cannot be changed until you completely stop the ClearVRPlayer.

Declaration

public bool allowDecoderContraintsInducedStereoToMono

Field Value

TYPE	DESCRIPTION
System.Boolean	

applicationFocusAndPauseHandling

How to handle application pause/unpause and suspend/resume. See ApplicationFocusAndPauseHandlingTypes for details.

Declaration

public ApplicationFocusAndPauseHandlingTypes applicationFocusAndPauseHandling

Field Value

TYPE	DESCRIPTION
ApplicationFocusAndPauseHandlingTypes	

audioDecoder

The AudioDecoder to use. Must be set to its default value null.

Declaration

public AudioDecoder audioDecoder

Field Value

TYPE	DESCRIPTION
AudioDecoder	

audioFocusChangedHandlingType

How to handle the loss and gain of the audio focus. SeeAudioFocusChangedHandlingTypes for details.

Declaration

public AudioFocusChangedHandlingTypes audioFocusChangedHandlingType

Field Value

TYPE	DESCRIPTION
AudioFocusChangedHandlingTypes	

audioPlaybackEngine

The AudioPlaybackEngine to use. Must be set to its default value null.

Declaration

public AudioPlaybackEngine audioPlaybackEngine

Field Value

TYPE		DESCRIPTION
AudioPla	ybackEngine	

autoPlay

Whether content playback should automatically start after content has been loaded. Default value: true

Declaration

public bool autoPlay

Field Value

TYPE	DESCRIPTION
System.Boolean	

autoPrepareContentItem

If specified, this ContentItem will be automatically loaded after the ClearVRPlayer has completed initialization. By default, we highly recommend you to set this to your ContentItem of choice.

Declaration

public ContentItem autoPrepareContentItem

Field Value

TYPE	DESCRIPTION
ContentItem	

cameraAndContentPlacementMode

Specify the behavior of the ClearVRPlayer object regarding the camera and display object automatic placement after a content switch. Default value: CameraAndContentPlacementModes.Default You can set this to Disabled to not move the Camera and/or mesh to their default/recommended position.

Declaration

public CameraAndContentPlacementModes cameraAndContentPlacementMode

Field Value

TYPE	DESCRIPTION
CameraAndContentPlacementModes	

clearVRCoreLogToFile

Instead of writing to stdout, the core ClearVRCore log will be written to the specified file.

Declaration

public String clearVRCoreLogToFile

Field Value

TYPE	DESCRIPTION
String	

clearVRCoreVerbosity

ClearVRCore verbosity level. Please keep at 0 at all times.

Declaration

public int clearVRCoreVerbosity

Field Value

TYPE	DESCRIPTION
System.Int32	

contentProtectionRobustnessLevel

Set the required robustness of playback of protected content. Default value:

ContentProtectionRobustnessLevels.Unprotected Note that this toggle is effective throughout the lifetime of your ClearVRPlayer object.

Declaration

public ContentProtectionRobustnessLevels contentProtectionRobustnessLevel

Field Value

TYPE	DESCRIPTION
ContentProtectionRobustnessLevels	

deviceParameters

Allows you to manually override the device type and screen dimensions. This is an advanced API, and under normal circumstances there should never be the need to change this from its default value: null. Since v8.0.

6 NOTE

This field replaces the screenWidth/screenHeight and deviceType fields that were available in v7 and older. If you are targeting a custom headset that is not correctly auto-detected by the SDK, you can use this field to deviceType and screen dimensions.

Declaration

public DeviceParameters deviceParameters

Field Value

TYPE	DESCRIPTION
DeviceParameters	

deviceType

The device type. For optimal performance it is essential to set the correct device type. Set/keep at its default value DeviceTypes.Unknown for auto-detection. If you fail to set the correct value, poor performance, poor visual quality or stereoscopic playback may break.

Declaration

public DeviceTypes deviceType

Field Value

TYPE	DESCRIPTION
DeviceTypes	

enableABR

Enable or disable ABR. Default value: true

Declaration

public bool enableABR

Field Value

TYPE	DESCRIPTION
System.Boolean	

enableAutomaticRenderModeSwitching

ClearVR's SwitchContent() API allows you to switching between monoscopic and stereoscopic clips. This feature requires careful management of the active RenderMode. This option applies only when targeting HMDs, as playback will always be monoscopic if no HMD is detected. When set to true (default value), the ClearVRPlayer will automatically configure the RenderMode (monoscopic/stereoscopic) based on the active content item.

▲ WARNING

Setting this value to false will require the developer to implement all RenderMode switching logic himself. This can be a daunting task given the many permutations possible. You are discouraged to do so.

Declaration

public bool enableAutomaticRenderModeSwitching

Field Value

TYPE	DESCRIPTION
System.Boolean	

enableAutomaticRenderModeSwitchingOnContentFormatChanged

Declaration

public bool enableAutomaticRenderModeSwitchingOnContentFormatChanged

Field Value

TYPE	DESCRIPTION
System.Boolean	

httpProxyParameters

Since v5.1 Specify the HTTP proxy settings. Note that the lower-level SDK will attempt to detect proxy host and port automatically if host and port are at their default values ("<auto>" and -1 respectively). Due to platform security constraints, we cannot detect username and password automatically. We must rely on the application to provide those. See SetProxyParameters for details.

Declaration

public readonly ProxyParameters httpProxyParameters

Field Value

TYPE	DESCRIPTION
ProxyParameters	

httpsProxyParameters

Since v5.1 Specify the HTTPS proxy settings. Note that the lower-level SDK will attempt to detect proxy host and port automatically if host and port are at their default values ("<auto>" and -1 respectively). Due to platform security constraints, we cannot detect username and password automatically. We must rely on the application to provide those. See SetProxyParameters for details.

Declaration

public readonly ProxyParameters httpsProxyParameters

Field Value

TYPE	DESCRIPTION
ProxyParameters	

initialAudioGain

Allows one to override the initial audio gain [0, 1]. Since: v8.0

Declaration

public float initialAudioGain

Field Value

TYPE	DESCRIPTION
System.Single	

initializeFlags

Declaration

public long initializeFlags

Field Value

TYPE	DESCRIPTION
System.Int64	

isTransparencySupportEnabled

Setting this to true (default: false) allows the implementor to change the transparency of the sphere by using the mediaPlayer.SetMainColor() api. For common performance reasons not specifically related to ClearVR streaming, you should carefully assess the potential negative performance impact of using transparency-enabled shaders on mobile devices. Please note that you need to manually include the required shaders into your project (via Edit -> Project Settings -> Graphics -> Always Included Shaders).

Declaration

public bool isTransparencySupportEnabled

TYPE	DESCRIPTION
System.Boolean	

isVRDevicePresent

Holds whether a VR headset is present or not. Note that this value is ALWAYS overwritten based on the deviceType field. Setting this value yourself does not have any effect.

Declaration

public bool isVRDevicePresent

Field Value

TYPE	DESCRIPTION
System.Boolean	

licenseFileBytes

The ClearVRPlayer requires a license file to function. Provide the binary data here.

Declaration

public byte[] licenseFileBytes

Field Value

ТҮРЕ	DESCRIPTION
System.Byte[]	

loopContent

Whether content should loop when it has reached the end of the clip. Default value: true. Note that live content does not support looping.

Declaration

public bool loopContent

ТҮРЕ	DESCRIPTION
System.Boolean	

overrideColorSpace

Specifies the active color space. Default value is based on QualitySettings.activeColorSpace You can use ColorSpacesMethods.ConvertUnityColorSpace() to convert a Unity ColorSpace enum, into a ClearVR ColorSpaces enum.

Declaration

public ColorSpaces overrideColorSpace

Field Value

TYPE	DESCRIPTION
ColorSpaces	

overrideUserAgent

This field allows you to override the user agent field in each video-streaming related HTTP request. As viewport-adaptive streaming results in many HTTP requests, one is STRONGLY discouraged to set this field as it will increase network overhead.. If needed, be sure to keep this String as short as possible. Also, this string cannot be changed after the ClearVRPlayer has initialized. Default value: "" (an empty string)

Declaration

public String overrideUserAgent

Field Value

TYPE	DESCRIPTION
String	

ovrOverlayOptions

You can tweak OVROverlay specific settings to your preference on this struct. See OVROverlayOptions for details.

This applies to Android only currently.

Declaration

public OVROverlayOptions ovrOverlayOptions

TYPE	DESCRIPTION
OVROverlayOptions	

parentGameObject

The parent gameobject of the ClearVRPlayer. If set to null (default value) it will be set to the GameObject the ClearVRPlayer script is attached to. This is preferred default behaviour.

Declaration

public GameObject parentGameObject

Field Value

TYPE	DESCRIPTION
GameObject	

platform

Automatically inferred, do not overwrite.

Declaration

public readonly RuntimePlatform platform

Field Value

ТҮРЕ	DESCRIPTION
RuntimePlatform	

preferredRenderMode

The preferred RenderMode of the content. Default value: RenderModes. Native.

NOTE

The value of this field will be ignored if enableAutomaticRenderModeSwitching = false.

Declaration

public RenderModes preferredRenderMode

TYPE	DESCRIPTION
RenderModes	

prepareContentForPlayoutTimeoutInMilliseconds

Any value <= 0 will result in the default timeout to be used. This default value is currently 30000 milliseconds. This timeout will only be triggered if loading a specific content item took longer than the specified amount of time. In this case, one will receive a ClearVRCoreWrapperContentLoadingTimeout message.

Declaration

public int prepareContentForPlayoutTimeoutInMilliseconds

Field Value

TYPE	DESCRIPTION
System.Int32	

prepareContentParameters

Specifies what content should be loaded and at what position playout should start. Cannot be null. This replaces the deprecated autoPrepareContentItem field.

Since: v8.0

1 NOTE

Before v8.0, one could set autoPrepareContentItem to null to only bootstrap the player. This feature has been removed in v8.0 because speed gain was very minimal (less then 50 msec)

Declaration

public PrepareContentParameters prepareContentParameters

Field Value

TYPE	DESCRIPTION
PrepareContentParameters	

renderCamera

Since v6.0 Specifies the active camera rendering to the screen. Default value: Camera.main

Declaration

public Camera renderCamera

Field Value

TYPE	DESCRIPTION
Camera	

screenHeight

Declaration

public short screenHeight

Field Value

TYPE	DESCRIPTION
System.Int16	

screenWidth

The width of the screen in pixels. Keep at the default value 0 for auto-detection.

Declaration

public short screenWidth

Field Value

TYPE	DESCRIPTION
System.Int16	

telemetryConfiguration

Configure telemetry services reporting. See TelemetryConfiguration for details. Default value: null.

Declaration

public TelemetryConfiguration telemetryConfiguration

TYPE	DESCRIPTION
TelemetryConfiguration	

textureBlitMode

Since v6.0 Specifies how the video texture should be rendered.

- Android (OpenGLES 2): TextureBlitModes.UVShufflingZeroCopy (default) and TextureBlitModes.UVShufflingCopy are allowed. The latter should ONLY be used if targetting Picture in Picture mode using another third party video player at the same time as when using ClearVR.
- Android (OpenGLES 3): TextureBlitModes.UVShufflingZeroCopy (default),
 TextureBlitModes.UVShufflingCopy and TextureBlitModes.OVROverlayZeroCopy are allowed.
 TextureBlitModes.UVShufflingCopy should ONLY be used if targetting Picture in Picture mode using another third party video player at the same time as when using ClearVR.
 TextureBlitModes.OVROverlayZeroCopy implies the use of OVROverlay on Oculus hardware and is ONLY enabled if the ClearVR Oculus VR Extensions have been enabled in the ClearVR drop down menu.
- iOS (OpenGLES 2 and 3): only TextureBlitModes.UVShufflingZeroCopy (default) is allowed. OpenGLES 2 and 3 are only supported on Unity versions 2020.2 and older.
- iOS (Metal): only TextureBlitModes.UVShufflingCopy (default) is allowed.
- Win64 (OpenGLCore): only TextureBlitModes.UVShufflingZeroCopy (default) is allowed.
- Win64 (Direct3D11): only TextureBlitModes.UVShufflingZeroCopy (default) is allowed.
- Linux (OpenGLCore): only TextureBlitModes.UVShufflingZeroCopy (default) is allowed.

6 NOTE

TextureBlitModes.OVROverlayZeroCopy support is ONLY supported on Oculus Android hardware (e.g. the Quest 1 and 2). It supports playback of non-ClearVR encoded content like rectilinear ("16x9") feeds, and traditional ERP360, ERP180 and Cubemap content.

Declaration

public TextureBlitModes textureBlitMode

Field Value

TYPE	DESCRIPTION
TextureBlitModes	

trackingTransform

Since v4.1.2 Supersedes the deprecated camera field. This transform is used to track the highquality viewport. Typically, one would set this field to the activeCamera.transform field. One could also use the transform of for example a random gameobject and let the viewport track the orientation of it.

Declaration

public Transform trackingTransform

Field Value

ТҮРЕ	DESCRIPTION
Transform	

vrAPIType

Specifies the currently active VR API. By default this will be auto-detected, but you can override it my manually specifying. The latter is, however, not recommended. Notes.

- 1. Auto detection is currently limited to Oculus VR and OpenVR only. Any other VRApi is detected as Unknown.
- 2. One is not encouraged to rely on this value for your own application logic.

Declaration

public VRAPITypes vrAPIType

Field Value

TYPE	DESCRIPTION
VRAPITypes	

Methods

Print()

Declaration

public void Print()

ToString()

Declaration

public override String ToString()

Returns

ТҮРЕ	DESCRIPTION
String	

Verify(ClearVRLayoutManager)

Used internally to verify whether the specified options are supported by the current platform. Do not use yourself! In case of failure, an error message will be logged to the console. The callee should stop execution if false is returned from this method, as library behaviour is undefined otherwise.

Declaration

public virtual bool Verify(ClearVRLayoutManager argClearVRLayoutManager)

Parameters

TYPE	NAME	DESCRIPTION
ClearVRLayoutManager	argClearVRLayoutManager	

Returns

TYPE	DESCRIPTION
System.Boolean	True in case of success, false otherwise.

Implements

PlatformOptionsInterface

Interface PlatformOptionsInterface

This is an internal interface and should not be used by the end-user.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public interface PlatformOptionsInterface

Methods

Verify(ClearVRLayoutManager)

Verifies whether the PlatformOptions are correct.

Declaration

bool Verify(ClearVRLayoutManager argClearVRLayoutManager)

Parameters

TYPE	NAME	DESCRIPTION
ClearVRLayoutManager	argClearVRLayoutManager	

Returns

TYPE	DESCRIPTION
System.Boolean	True if verification succeeded, false otherwise.

Class PlatformOptionsIOS

iOS platform specific platform options. Currently, there are NO additional options/fields to set.

Inheritance

System.Object

PlatformOptionsBase

PlatformOptionsIOS

Implements

PlatformOptionsInterface

Inherited Members

PlatformOptionsBase.licenseFileBytes

PlatformOptionsBase.isVRDevicePresent

PlatformOptionsBase.parentGameObject

PlatformOptionsBase.preferredRenderMode

PlatformOptionsBase.autoPlay

PlatformOptionsBase.loopContent

PlatformOptionsBase.autoPrepareContentItem

PlatformOptionsBase.audioPlaybackEngine

PlatformOptionsBase.audioDecoder

PlatformOptionsBase.applicationFocusAndPauseHandling

PlatformOptionsBase.audioFocusChangedHandlingType

PlatformOptionsBase.enableAutomaticRenderModeSwitching

PlatformOptionsBase.screenWidth

PlatformOptionsBase.screenHeight

PlatformOptionsBase.deviceType

PlatformOptionsBase.deviceParameters

PlatformOptionsBase.clearVRCoreVerbosity

PlatformOptionsBase.clearVRCoreLogToFile

PlatformOptionsBase.enableAutomaticRenderModeSwitchingOnContentFormatChanged

PlatformOptionsBase.initializeFlags

Platform Options Base. prepare Content For Playout Time out In Millise conds

PlatformOptionsBase.isTransparencySupportEnabled

PlatformOptionsBase.trackingTransform

PlatformOptionsBase.httpProxyParameters

PlatformOptionsBase.httpsProxyParameters

PlatformOptionsBase.renderCamera

PlatformOptionsBase.textureBlitMode

PlatformOptionsBase.overrideColorSpace

PlatformOptionsBase.vrAPIType

PlatformOptionsBase.ovrOverlayOptions

PlatformOptionsBase.contentProtectionRobustnessLevel

PlatformOptionsBase.cameraAndContentPlacementMode

PlatformOptionsBase.overrideUserAgent

PlatformOptionsBase.initialAudioGain

PlatformOptionsBase.enableABR

PlatformOptionsBase.abrStartMode

PlatformOptionsBase.allowDecoderContraintsInducedStereoToMono

PlatformOptionsBase.prepareContentParameters

PlatformOptionsBase.telemetryConfiguration

PlatformOptionsBase.Print()

PlatformOptionsBase.ToString()

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class PlatformOptionsIOS: PlatformOptionsBase, PlatformOptionsInterface

Fields

isUseOESFastPathEnabled

Declaration

public bool isUseOESFastPathEnabled

Field Value

TYPE	DESCRIPTION
System.Boolean	

platform

Declaration

public RuntimePlatform platform

Field Value

ТҮРЕ	DESCRIPTION
RuntimePlatform	

Methods

Verify(ClearVRLayoutManager)

vennes il 103 pianonni specino opnons ale vana di non

Declaration

public override bool Verify(ClearVRLayoutManager argClearVRLayoutManager)

Parameters

TYPE	NAME	DESCRIPTION
ClearVRLayoutManager	argClearVRLayoutManager	

Returns

TYPE	DESCRIPTION
System.Boolean	true if everything is OK, false otherwise.

Overrides

PlatformOptionsBase.Verify(ClearVRLayoutManager)

Implements

PlatformOptionsInterface

Class PlatformOptionsPC

PC (Linux/Windows) platform specific platform options. Currently, there are NO additional options/fields to set.

Inheritance

System.Object

PlatformOptionsBase

PlatformOptionsPC

Implements

PlatformOptionsInterface

Inherited Members

PlatformOptionsBase.licenseFileBytes

PlatformOptionsBase.isVRDevicePresent

PlatformOptionsBase.parentGameObject

PlatformOptionsBase.preferredRenderMode

PlatformOptionsBase.autoPlay

PlatformOptionsBase.loopContent

PlatformOptionsBase.autoPrepareContentItem

PlatformOptionsBase.audioPlaybackEngine

PlatformOptionsBase.audioDecoder

PlatformOptionsBase.applicationFocusAndPauseHandling

PlatformOptionsBase.audioFocusChangedHandlingType

PlatformOptionsBase.enableAutomaticRenderModeSwitching

PlatformOptionsBase.screenWidth

PlatformOptionsBase.screenHeight

PlatformOptionsBase.deviceType

PlatformOptionsBase.deviceParameters

PlatformOptionsBase.clearVRCoreVerbosity

PlatformOptionsBase.clearVRCoreLogToFile

PlatformOptionsBase.enableAutomaticRenderModeSwitchingOnContentFormatChanged

PlatformOptionsBase.initializeFlags

PlatformOptionsBase.prepareContentForPlayoutTimeoutInMilliseconds

PlatformOptionsBase.isTransparencySupportEnabled

PlatformOptionsBase.trackingTransform

PlatformOptionsBase.httpProxyParameters

PlatformOptionsBase.httpsProxyParameters

PlatformOptionsBase.renderCamera

PlatformOptionsBase.textureBlitMode

PlatformOptionsBase.overrideColorSpace

PlatformOptionsBase.vrAPIType

(c) Tiledmedia B.V. 2017-2022

PlatformOptionsBase.ovrOverlayOptions

PlatformOptionsBase.contentProtectionRobustnessLevel

PlatformOptionsBase.cameraAndContentPlacementMode

PlatformOptionsBase.overrideUserAgent

PlatformOptionsBase.initialAudioGain

PlatformOptionsBase.enableABR

PlatformOptionsBase.abrStartMode

PlatformOptionsBase.allowDecoderContraintsInducedStereoToMono

PlatformOptionsBase.prepareContentParameters

PlatformOptionsBase.telemetryConfiguration

PlatformOptionsBase.Print()

PlatformOptionsBase.ToString()

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class PlatformOptionsPC: PlatformOptionsBase, PlatformOptionsInterface

Fields

platform

Declaration

public RuntimePlatform platform

Field Value

TYPE	DESCRIPTION
RuntimePlatform	

Methods

Verify(ClearVRLayoutManager)

Declaration

public override bool Verify(ClearVRLayoutManager argClearVRLayoutManager)

Parameters

TYPE	NAME	DESCRIPTION
ClearVRLayoutManager	argClearVRLayoutManager	

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

PlatformOptionsBase.Verify(ClearVRLayoutManager)

Implements

PlatformOptionsInterface

Class PlaybackParameters

Inheritance

System.Object

PlaybackParameters

PrepareContentParameters

SwitchContentParameters

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class PlaybackParameters : object

Constructors

PlaybackParameters(PlaybackParameters)

Declaration

protected PlaybackParameters(PlaybackParameters argOtherPlaybackParameters)

Parameters

TYPE	NAME	DESCRIPTION
PlaybackParameters	argOtherPlaybackParameters	

Properties

approximateDistanceFromLiveEdgeInMilliseconds

Approximate offset from live edge (in milliseconds). Ignored for non-live (VOD) content. The default value is 0 (see note below). The actual value depends on whether a minimum offset is enforced and on the encoding parameters of the live feed (e.g. segment duration). Since: v8.0

6 NOTE

A minimum offset might be enforced.

A WARNING

Changing this value directly impacts the camera-lens-to-client-device latency. This is an advanced parameter, changing it is strongly discouraged. Please leave at its default value 0 unless you know exactly what you are doing.

Declaration

public long approximateDistanceFromLiveEdgeInMilliseconds { get; set; }

Property Value

TYPE	DESCRIPTION
System.Int64	The approximate distance from the live edge, in milliseconds

contentItem

The ContentItem to load, cannot be null.

Declaration

public ContentItem contentItem { get; set; }

Property Value

TYPE	DESCRIPTION
ContentItem	

layoutParameters

Describes the FeedConfiguration, where the video(s) will be positioned.

A WARNING

Although accepted, passing null is NOT recommend. Instead, use the LayoutManager to configure your audio/video layout. Legacy mesh positioning will be performed when set to null, but this behaviour will be removed after 2023-03-30.

Declaration

public LayoutParameters layoutParameters { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
LayoutParameters	The feed configuration

preferredAudioLanguage

If set to an ISO-639 language code, an audio track matching this value will be selected by default.

Declaration

public String preferredAudioLanguage { get; }

Property Value

TYPE	DESCRIPTION
String	The ISO-639 language code as a string.

preferredAudioTrackLanguage

If set to an ISO-639 language code, an audio track matching this value will be selected by default.

Declaration

public String preferredAudioTrackLanguage { get; }

Property Value

TYPE	DESCRIPTION
String	The ISO-639 language code as a string.

preferred Subtitles Language

If set to an ISO-639 language code, a subtitles track matching this value will be selected by default.

Declaration

public String preferredSubtitlesLanguage { get; }

Property Value

TYPE	DESCRIPTION
String	The ISO-639 language code as a string.

syncSettings

Settings for sync.

Declaration

public SyncSettings syncSettings { get; set; }

Property Value

TYPE	DESCRIPTION
SyncSettings	

timingParameters

The TimingParameters, e.g. the initial playout start position in milliseconds and how to interpet this position. If set to null, default values will be used.

Declaration

public TimingParameters timingParameters { get; set; }

Property Value

TYPE	DESCRIPTION
TimingParameters	

Methods

ToString()

Declaration

public override String ToString()

Returns

TYPE	DESCRIPTION
String	

Class PopulateMediaInfoParameters

The PopulateMediaInfoParameters class is used in conjunction with the PopulateMediaInfo() API.

Inheritance

System.Object

PopulateMediaInfoParameters

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class PopulateMediaInfoParameters : object

Constructors

PopulateMediaInfoParameters(ContentItem, DeviceParameters)

Populate media info retrieves the provided content item and parses its media info (available audo tracks, DRM properties, etc.)

IP IMPORTANT

This is considered an advanced API, and one should typically refrain from using it.

Declaration

public PopulateMediaInfoParameters(ContentItem argContentItem, DeviceParameters argDeviceParameters = null)

Parameters

TYPE	NAME	DESCRIPTION
ContentItem	argContentItem	The ContentItem to parse.
DeviceParameters	argDeviceParameters	The DeviceParameters that should be taken into account when parsing the content item.

Properties

contentItem

The Content item to populate its media info.

Declaration

public ContentItem contentItem { get; }

Property Value

TYPE	DESCRIPTION
ContentItem	The content item

deviceParameters

The device parameters that must be taken into consideration when populating the media info.

Declaration

public DeviceParameters deviceParameters { get; }

Property Value

TYPE	DESCRIPTION
DeviceParameters	The device parameters.

Class PrepareContentParameters

Parameters used to prepare a ClearVRPlayer for content playout.

Inheritance

System.Object

PlaybackParameters

PrepareContentParameters

Inherited Members

PlaybackParameters.contentItem

PlaybackParameters.timingParameters

PlaybackParameters.syncSettings

PlaybackParameters.approximateDistanceFromLiveEdgeInMilliseconds

PlaybackParameters.preferredAudioTrackLanguage

PlaybackParameters.preferredAudioLanguage

PlaybackParameters.preferredSubtitlesLanguage

PlaybackParameters.layoutParameters

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class PrepareContentParameters : PlaybackParameters

Constructors

PrepareContentParameters(ContentItem, TimingParameters, LayoutParameters)

The default constructor taking the minimum subset of arguments required.

A WARNING

Since v9.x Although accepted, passing null as argLayoutParameters is NOT recommend. Instead, use the LayoutManager to configure your audio/video feed layout. Legacy Display Object positioning will be performed hen set to null, but this behaviour will be removed after 2023-03-31. When playing Mosaic content, this parameter *MUST* be specified and cannot be null.

Declaration

public PrepareContentParameters(ContentItem argContentItem, TimingParameters argTimingParameters, LayoutParameters argLayoutParameters)

TYPE	NAME	DESCRIPTION
ContentItem	argContentItem	The content item to play
TimingParameters	argTimingParameters	The timing parameters, specifying how to start playback. Setting this to null will trigger default behaviour and is a valid value, see TimingParameters for details.
LayoutParameters	argLayoutParameters	The LayoutParameters specifying which DisplayObject will render what Feed. For backwards compatibility with SDKs prior to v9.0, you can set this argument to null. This will trigger default display object placement behaviour, but this mode will be removed after 2023-03-31.

PrepareContentParameters(ContentItem, TimingParameters, SyncSettings, Int64, Int32, Int64, LayoutParameters)

Specifies the parameters used for loading the ContentItem during ClearVRPlayer initialization.

Declaration

public PrepareContentParameters(ContentItem argContentItem, TimingParameters argTimingParameters = null, SyncSettings argSyncSettings = null, long argFlags = null, int argTimeoutInMilliseconds = 0, long argApproximateDistanceFromLiveEdgeInMilliseconds = null, LayoutParameters argLayoutParameters = null)

TYPE	NAME	DESCRIPTION
ContentItem	argContentItem	The content item to load. Can no
TimingParameters	argTimingParameters	Configures when playback shoul provided ContentItem. If set to the VOD clips will start playback at public clips will start at the live edge.
SyncSettings	argSyncSettings	Enable sync with supplied setting null (e.g. sync is disabled)

ТҮРЕ	NAME	DESCRIPTION
System.Int64	argFlags	The initialization flags for prepari Default value: 0. Keep at 0 for no
System.Int32	argTimeoutInMilliseconds	Any value <= 0 will result in the cused. This default value is 30000 timeout will only be triggered if Iccontent item took longer than the time. In this case, one will receiv ClearVRCoreWrapperContentLomessage.
System.Int64	argApproximateDistanceFromLiveEdgeInMilliseconds	Specifies the approximate offset from the live edge. Default value a non-0 minimum value might be the default value is strongly discrapproximateDistanceFromLiveEndetails.
LayoutParameters	argLayoutParameters	Configure the layout of the streat [LayoutManager] (xref:com.tiledmedia.clearvr.Cleat for details.

PrepareContentParameters(ContentItem, TimingParameters, SyncSettings, Int64, Int32, Int64, String)

Specifies the parameters used for loading the ContentItem during ClearVRPlayer initialization.

Declaration

public PrepareContentParameters(ContentItem argContentItem, TimingParameters argTimingParameters, SyncSettings argSyncSettings, long argFlags, int argTimeoutInMilliseconds, long argApproximateDistanceFromLiveEdgeInMilliseconds, String argPreferredAudioTrackLanguage)

TYPE	NAME	DESCRIPTION
ContentItem	argContentItem	The content item to load. Can no

TYPE	NAME	DESCRIPTION
TimingParameters	argTimingParameters	Configures when playback shoul provided ContentItem. If set to the null, VOD clips will start playback while LIVE clips will start at the li
SyncSettings	argSyncSettings	Enable sync with supplied setting null (e.g. sync is disabled)
System.Int64	argFlags	The initialization flags for prepari item. Default value: 0. Keep at 0
System.Int32	argTimeoutInMilliseconds	Any value <= 0 will result in the c be used. This default value is 30 This timeout will only be triggere specific content item took longer specified amount of time. In this receive a ClearVRCoreWrapperContentLo message.
System.Int64	argApproximateDistanceFromLiveEdgeInMilliseconds	Specifies the approximate offset kept from the live edge. Default vector Note that a non-0 minimum value enforced. Changing the default vector discouraged, see also approximateDistanceFromLiveEnfor details.
String	argPreferredAudioTrackLanguage	Override the preferred audio trac setting this to the preferred ISO-code. Only works if language code mbedded in the source stream. which is interpreted as "automati audio track".

Properties

flags

Flags can be used to tune content loading behaviour. Currently, no flags are supported. This will allways be set to the default value: 0.

Declaration

<pre>public long flags { get; set; }</pre>	
--	--

Property Value

TYPE	DESCRIPTION
System.Int64	

timeoutInMilliseconds

The time it takes before loading the specified clip is considered to have timed out. Any negative value or 0 will be considered as "use the default value", which is 30000 msec.

Declaration

public int timeoutInMilliseconds { get; set; }

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

ToString()

Declaration

public override String ToString()

Returns

TYPE	DESCRIPTION
String	

Overrides

PlaybackParameters.ToString()

Enum PrewarmCacheFlags

Flags related to prewarming cache. See ClearVRPlayer.PrewarmCache() for details.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum PrewarmCacheFlags: int

Fields

NAME	DESCRIPTION
None	Default, no flags raised.

Class PrewarmCacheParameters

Specifies what content to pre-load. This class is deprecated and can no longer be used. Cache prewarming has been removed in v8 of the SDK

Inheritance

System.Object

PrewarmCacheParameters

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class PrewarmCacheParameters : object

Constructors

PrewarmCacheParameters(ContentItem)

Declaration

public PrewarmCacheParameters(ContentItem argContentItem)

Parameters

TYPE	NAME	DESCRIPTION
ContentItem	argContentItem	

PrewarmCacheParameters(ContentItem, Int64)

Declaration

public PrewarmCacheParameters(ContentItem argContentItem, long argFlags)

ТҮРЕ	NAME	DESCRIPTION
ContentItem	argContentItem	
System.Int64	argFlags	

Enum ProjectionTypes

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

nublic	enum	Projection ⁻	Tynes	٠	int
public	CHUIII	i rojection	i ypcs	٠	HIL

Fields

NAME	DESCRIPTION
Erp180Mono	
Erp180StereoSideBySide	
Erp360Mono	
Erp360StereoTopBottom	
FisheyeMono	
FisheyeStereoSideBySide	
MeshBoxMono	
MeshBoxStereo	
RectilinearMono	
RectilinearStereoSideBySide	
RectilinearStereoTopBottom	
Unknown	

Class ProxyParameters

Helper class to define HTTP/HTTPS proxy parameters.

Inheritance

System.Object

ProxyParameters

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class ProxyParameters : object

Constructors

ProxyParameters(ProxyTypes)

Default constructor. Note that you first construct a ProxyParameters class, and subsequently call SetProxyParameters() to configure it accordingly.

Declaration

public ProxyParameters(ProxyTypes argProxyType)

Parameters

TYPE	NAME	DESCRIPTION
ProxyTypes	argProxyType	The proxy type.

Properties

host

Get the host as a string

Declaration

public String host { get; }

Property Value

TYPE	DESCRIPTION
String	The host

password

Get the password as a string.

Declaration

public String password { get; }

Property Value

TYPE	DESCRIPTION
String	The password

port

Get the port as an integer.

Declaration

public int port { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	The port

proxyType

Get proxy type

Declaration

public ProxyTypes proxyType { get; }

Property Value

ТҮРЕ	DESCRIPTION
ProxyTypes	The type of this ProxyParameters class.

username

Get the username as a string.

(c) Tiledmedia B.V. 2017-2022

Declaration

public String username { get; }

Property Value

TYPE	DESCRIPTION
String	The username

Methods

SetProxyParameters(String, Int32, String, String)

Set the proxy settings. At minimal, host and port need to specified if a proxy is active. You can either opt for auto-detection (setting them to "<auto>" and -1 respectively or disable proxy detection completely (by setting them to "" and 0 respectively). Optionally a username and password can be set. Leave username and password empty (e.g. an empty string) in case an anonymous proxy is used. The proxy host. Set to its default value "<auto>" for auto-detection. Set to an empty string to display the proxy. The proxy port. Set to its default value -1 for auto-detection, set to 0 for disable the proxy. The proxy username. Default value: "<auto>". Note that we can ONLY auto-detect the username on iOS, so you have to provide it on any other platform. Set this value in case the proxy does not allow anonymous access. The proxy password. Default value: "". Note that we cannot auto-detect the password, it must always be provided in case the proxy does not allow anonymous access.

Declaration

public void SetProxyParameters(String argHost, int argPort, String argUsername = null, String argPassword = null)

Parameters

ТҮРЕ	NAME	DESCRIPTION
String	argHost	
System.Int32	argPort	
String	argUsername	
String	argPassword	

ToString()

public override String ToString()

Returns

TYPE	DESCRIPTION
String	

Enum ProxyTypes

Enum that distinguishes between different supported proxy types.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum ProxyTypes : int

NAME	DESCRIPTION
Http	HTTP proxy
Https	HTTPS proxy
Unknown	Unknown, not used.

Enum RenderModes

This enum specifies how the video is rendered to the display.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum RenderModes: int

NAME	DESCRIPTION
ForcedMonoscopic	Since v8.1 Force monoscopic playback, even if content is stereoscopic. When set to this mode, content playback will always stick to monoscopic whatever the clip you SwitchContent() to.
Monoscopic	Rendering will be forced to monoscopic: • Stereoscopic clips will be rendered as monoscopic on all devices. • Monoscopic clips will be rendered as monoscopic on all devices.
	▲ WARNING The RenderMode will reset to the behaviour as described under Native when calling SwitchContent()! This means that if you used the SetRenderMode() API to switch to monoscopic rendering during playback, rendering will switch to Stereoscopic again when switching to a stereoscopic clip. If you want to enforce monoscopic rendering even after a SwitchContent, refer to ForcedMonoscopic instead.
Native	 Rendering will follow the content's format. This is the default behaviour. Stereoscopic clips will be rendered as stereoscopic on headsets and monoscopic on non-HMD devices. Monoscopic clips will be rendered as monoscopic on all devices.

NAME	DESCRIPTION
	Rendering will be forced to Stereoscopic on HMD if the current playing content is stereoscopic. This has no effect on non-HMD devices.
Stereoscopic	The RenderMode will reset to the behaviour as described under Native when calling SwitchContent()!

Enum SDKTypes

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum SDKTypes : int	
----------------------------	--

NAME	DESCRIPTION
Native	
Unity	
Unknown	
Web	

Enum SeekFlags

Allows one to specify how the new content position in a Seek() or SwitchContent() call (specified in milliseconds) should be interpreted.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum SeekFlags: int

NAME	DESCRIPTION
LiveEdge	When specified, seek to live edge of the currently playing ContentItem. This flag is only valid when playing Live content.
None	Default, no flags raised.
RelativeTime	Use the RelativeTime flag to seek based on the current content position. This saves you from grabbing the current content position and adding the required offset yourself.
Seamless	Use the Seamless flag to seamlessly switch between two content items. This is typically used when switching between cameras in a synchronized multi-camera event. This does not apply to seeking, only switch content.
WallclockTime	When specified, seek to specified wallclock time. Normally, one would seek based on content time. This flag is typically used when playing back synchronized live broadcast with multiple cameras.

Class SeekParameters

Parameters to specify seek operation. Note that frame accurate seek is not guaranteed.

Inheritance

System.Object

SeekParameters

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class SeekParameters: object

Constructors

SeekParameters(TimingParameters, TransitionTypes)

Default constructor, allowing you to specify the new seek position, and how to interpret this new seek position.

Declaration

public SeekParameters(TimingParameters argTimingParameters = null, TransitionTypes argTransitionType =
default(TransitionTypes))

Parameters

TYPE	NAME	DESCRIPTION
TimingParameters	argTimingParameters	Describes the new position (in milliseconds) and how this new position should be interpreted. If set to null, VOD clips will seet to position 0, LIVE clips will seek to the live edge.
TransitionTypes	argTransitionType	Defines how the seek should be handled: Default value: Fast. Currently, no other value is supported and you will always default to this value when doing a seek.

SeekParameters(Int64)

Simplified constructor, defaulting to no SeekFlags.

Declaration

public SeekParameters(long argNewPositionInMilliseconds)

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	argNewPositionInMilliseconds	The position in milliseconds to seek to. It will depend on the type of content how accurate the seek will be.

SeekParameters(Int64, Int64)

Default constructor

Declaration

public SeekParameters(long argNewPositionInMilliseconds, long argFlags)

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	argNewPositionInMilliseconds	The position in milliseconds to seek to. It will depend on the type of content how accurate the seek will be.
System.Int64	argFlags	Binary OR-ed set of flags, or 0 (SeekFlags.None) in casee of no flags.

Class SharedPointersWithSDK

Inheritance

System.Object

SharedPointersWithSDK

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class SharedPointersWithSDK: object

Fields

debugDisplayObjectDescriptorPtr

Declaration

public IntPtr debugDisplayObjectDescriptorPtr

Field Value

TYPE	DESCRIPTION
IntPtr	

displayObjectDescriptorHeaderPtr

Declaration

public IntPtr displayObjectDescriptorHeaderPtr

Field Value

TYPE	DESCRIPTION
IntPtr	

dynamicDisplayObjectDescriptorPtr

Declaration

public IntPtr dynamicDisplayObjectDescriptorPtr

Field Value

TYPE	DESCRIPTION
IntPtr	

$static {\tt DisplayObject Descriptor Ptr}$

Declaration

public IntPtr staticDisplayObjectDescriptorPtr

Field Value

TYPE	DESCRIPTION
IntPtr	

Class StatisticsBase

Inheritance

System.Object

StatisticsBase

Implements

ClearVRCoreWrapperStatisticsInterface

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public abstract class StatisticsBase : object, ClearVRCoreWrapperStatisticsInterface

Fields

audioStatistics

Declaration

public StatisticsBase.AudioStatisticsBase audioStatistics

Field Value

TYPE	DESCRIPTION
StatisticsBase.AudioStatisticsBase	

isDestroyed

Declaration

protected bool isDestroyed

Field Value

TYPE	DESCRIPTION
System.Boolean	

videoStatistics

Declaration

public StatisticsBase.VideoStatisticsBase videoStatistics

Field Value

TYPE	DESCRIPTION
StatisticsBase.VideoStatisticsBase	

Properties

pipelineLatencyInNanoseconds

Declaration

public virtual long pipelineLatencyInNanoseconds { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	

Methods

Destroy()

Declaration

public virtual void Destroy()

Implements

 ${\bf Clear VR Core Wrapper Statistics Interface}$

Class StatisticsBase.AudioStatisticsBase

Inheritance

System.Object

StatisticsBase.AudioStatisticsBase

Implements

ClearVRCoreWrapperAudioStatisticsInterface

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public abstract class AudioStatisticsBase : object, ClearVRCoreWrapperAudioStatisticsInterface

Properties

framesDropped

Declaration

public virtual long framesDropped { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	

framesRendered

Declaration

public virtual long framesRendered { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	

playbackUnderrunCount

Declaration

public virtual int playbackUnderrunCount { get; }

TYPE	DESCRIPTION
System.Int32	

Methods

Destroy()

Declaration

public virtual void Destroy()

Implements

 ${\bf Clear VRCore Wrapper Audio Statistics Interface}$

Class StatisticsBase.VideoStatisticsBase

Inheritance

System.Object

StatisticsBase.VideoStatisticsBase

Implements

ClearVRCoreWrapperVideoStatisticsInterface

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public abstract class VideoStatisticsBase : object, ClearVRCoreWrapperVideoStatisticsInterface

Properties

averageDecoderInputQueueSize

Declaration

public virtual float averageDecoderInputQueueSize { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

average Decoder Output Queue Size

Declaration

public virtual float averageDecoderOutputQueueSize { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

endToEndFrameLatencyMean

Declaration

public virtual float endToEndFrameLatencyMean { get; }

TYPE	DESCRIPTION
System.Single	

end To End Frame Latency Standard Deviation

Declaration

public virtual float endToEndFrameLatencyStandardDeviation { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

frameReleaseQuality

Declaration

public virtual float frameReleaseQuality { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

framesDropped

Declaration

public virtual long framesDropped { get; }

Property Value

ТҮРЕ	DESCRIPTION
System.Int64	

framesRendered

Declaration

public virtual long framesRendered { get; }

TYPE	DESCRIPTION
System.Int64	

interFrameApplicationLatencyMean

Declaration

public virtual float interFrameApplicationLatencyMean { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

inter Frame Application Latency Standard Deviation

Declaration

public virtual float interFrameApplicationLatencyStandardDeviation { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

interFrameDecoderLatencyMean

Declaration

public virtual float interFrameDecoderLatencyMean { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

inter Frame Decoder Latency Standard Deviation

Declaration

public virtual float interFrameDecoderLatencyStandardDeviation { get; }

ТҮРЕ	DESCRIPTION
System.Single	

interFrameRenderLatencyMean

Declaration

public virtual float interFrameRenderLatencyMean { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

inter Frame Render Latency Standard Deviation

Declaration

public virtual float interFrameRenderLatencyStandardDeviation { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

vsyncQuality

Declaration

public virtual float vsyncQuality { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

Methods

Destroy()

Declaration

public virtual void Destroy()

GetEndToEndFrameLatencyAsPrettyString()

(c) Tiledmedia B.V. 2017-2022

Declaration

public virtual String GetEndToEndFrameLatencyAsPrettyString()

Returns

TYPE	DESCRIPTION
String	

GetInterFrameApplicationLatencyAsPrettyString()

Declaration

public virtual String GetInterFrameApplicationLatencyAsPrettyString()

Returns

TYPE	DESCRIPTION
String	

GetInterFrameApplicationRateInFramesPerSecond()

Declaration

public virtual float GetInterFrameApplicationRateInFramesPerSecond()

Returns

ТҮРЕ	DESCRIPTION
System.Single	

GetInterFrameDecoderLatencyAsPrettyString()

Declaration

public virtual String GetInterFrameDecoderLatencyAsPrettyString()

Returns

ТҮРЕ	DESCRIPTION
String	

GetInterFrameDecoderRateInFramesPerSecond()

Declaration

public virtual float GetInterFrameDecoderRateInFramesPerSecond()

Returns

TYPE	DESCRIPTION
System.Single	

GetInterFrameRenderLatencyAsPrettyString()

Declaration

public virtual String GetInterFrameRenderLatencyAsPrettyString()

Returns

TYPE	DESCRIPTION
String	

GetInterFrameRenderRateInFramesPerSecond()

Declaration

public virtual float GetInterFrameRenderRateInFramesPerSecond()

Returns

TYPE	DESCRIPTION
System.Single	

Implements

ClearVRCoreWrapperVideoStatisticsInterface

Class SubtitleTrackInfo

Helper class that holds information of one subtitle track. For detailed information about the ContentInfo, FeedInfo, VideoTrackInfo, AudioTrackInfo, SubtitleTrackInfo objects and their relationship, refer to the ContentItem documentation.

Inheritance

System.Object
SubtitleTrackInfo

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class SubtitleTrackInfo: object

Properties

boundToVideoTrackIndex

Some subtitle tracks cannot be individually accessed and played as they are bound to a specific video track. This is for example the case when the subtitle track and video track are multiplexed together into the same file. If this is the case, this API will return the index of that video track. If the subtitle track can be individually accessed, this API will return -1. This returns the index of the video track this subtitle track is bound to, or -1 if this subtitle track is not bound to any specific video track and can be individually accessed.

Declaration

public int boundToVideoTrackIndex { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

feedIndex

The Feed index associated with this Subtitle Track.

Declaration

public int feedIndex { get; }

TYPE	DESCRIPTION
System.Int32	

isActive

Whether this subtitle track is active or not.

Declaration

public bool isActive { get; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

language

Get the Language

Declaration

public String language { get; }

Property Value

TYPE	DESCRIPTION
String	

name

The name of the subtitle track, can be an empty string if not specified.

Declaration

public String name { get; }

TYPE	DESCRIPTION
String	For example "EN hearing impaired"

trackID

Returns the TrackID, a unique identifier for this track and the FeedInfo it belongs to. It can be used when constructing your FeedConfiguration.

Declaration

public TrackID trackID { get; }

Property Value

TYPE	DESCRIPTION
TrackID	The TracKID, it cannot be null.

trackIndex

The index of the subtitle track as int.

Declaration

public int trackIndex { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

url

Get the URL

Declaration

public String url { get; }

Property Value

TYPE	DESCRIPTION
String	

Methods

ToString()

Declaration

r	uhlic	override	etrina	ToString()	١
L	Jublic	overnue	Sumu	105011100	ł

Returns

TYPE	DESCRIPTION
System.String	

Enum SwitchContentFlags

SwitchContent specific flags.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum SwitchContentFlags : int

NAME	DESCRIPTION	
None	Default, no flags raised.	

Class SwitchContentParameters

Parameters required for switching to a different ContentItem.

Inheritance

System.Object

PlaybackParameters

SwitchContentParameters

Inherited Members

PlaybackParameters.contentItem

PlaybackParameters.timingParameters

PlaybackParameters.syncSettings

PlaybackParameters.approximateDistanceFromLiveEdgeInMilliseconds

PlaybackParameters.preferredAudioTrackLanguage

PlaybackParameters.preferredAudioLanguage

PlaybackParameters.preferredSubtitlesLanguage

PlaybackParameters.layoutParameters

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class SwitchContentParameters : PlaybackParameters

Constructors

SwitchContentParameters(ContentItem, TimingParameters, LayoutParameters)

The default constructor taking the minimum subset of arguments required.

A WARNING

Since v9.x Although accepted, passing null as argLayoutParameters is NOT recommend. Instead, use the LayoutManager to configure your audio/video feed layout. Legacy Display Object positioning will be performed hen set to null, but this behaviour will be removed after 2023-03-31. When playing Mosaic content, this parameter *MUST* be specified and cannot be null.

Declaration

public SwitchContentParameters(ContentItem argContentItem, TimingParameters argTimingParameters, LayoutParameters argLayoutParameters)

Parameters

TYPE	NAME	DESCRIPTION
ContentItem	argContentItem	The content item to play
TimingParameters	argTimingParameters	The timing parameters, specifying how to start playback. Setting this to null will trigger default behaviour and is a valid value, see TimingParameters for details.
LayoutParameters	argLayoutParameters	The LayoutParameters specifying which DisplayObject will render what Feed. For backwards compatibility with SDKs prior to v9.0, you can set this argument to null. This will trigger default display object placement behaviour, but this mode will be removed after 2023-03-31.

SwitchContentParameters(ContentItem, TimingParameters, TransitionTypes, AudioTrackAndPlaybackParameters, SyncSettings, Int64, Int64, String)

Default Constructor. Under typical conditions one should only set argContentItem and argTimingParameters and keep ther other fields at their default values.

Declaration

public SwitchContentParameters(ContentItem argContentItem, TimingParameters argTimingParameters = null,
TransitionTypes argTransitionType = default(TransitionTypes), AudioTrackAndPlaybackParameters
argAudioTrackAndPlaybackParameters = null, SyncSettings argSyncSettings = null, long argFlags = null, long
argApproximateDistanceFromLiveEdgeInMilliseconds = null, String argPreferredAudioTrackLanguage = null)

Parameters

TYPE	NAME	DESCRIPTION
ContentItem	argContentItem	The content ite switch to. Can null.

TYPE	NAME	DESCRIPTION
TimingParameters	argTimingParameters	The timing parameters, deplayback start position and he should be interwith respect to content. If set the default value in playback will set the beginning of clip (VOD) or a live edge (LIVE)
TransitionTypes	argTransitionType	Determines ho transition from current content to the specified content item sl be performed. Default value: TransitionType
AudioTrackAndPlaybackParameters	argAudioTrackAndPlaybackParameters	The audio dec and playback of to use after sw content. Keep default value: I
SyncSettings	argSyncSettings	Enable sync w supplied settin Default value: (sync = disable Note that spec SyncSettings i yet supported iOS!

TYPE	NAME	DESCRIPTION
System.Int64	argFlags	Customize swi content API behaviour. Cui unused. Leave default value: (
System.Int64	argApproximateDistanceFromLiveEdgeInMilliseconds	Specifies the approximate or that should be from the live or Default value: (msec). Note the non-0 minimur might be enfor Changing the or value is strong discouraged.
String	argPreferredAudioTrackLanguage	Override the preferred audic language by sethis to the preferred is code. Only wo language code embedded in the source stream Default value: which is interplas "automatica an audio track"

Properties

flags

Customize switch content behaviour, currently unused.

Declaration

public long flags { get; }

TYPE	DESCRIPTION
System.Int64	

transitionType

The transition type. See TransitionTypes for more information.

Declaration

public TransitionTypes transitionType { get; set; }

Property Value

TYPE	DESCRIPTION
TransitionTypes	

Methods

ToString()

Declaration

public override String ToString()

Returns

TYPE	DESCRIPTION
String	

Overrides

PlaybackParameters.ToString()

Interface SyncInterface

The sync interface can be used to control and view synchronization for live events.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public interface SyncInterface

Methods

DisableSync(Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Disable the sync algorithm.

Declaration

void DisableSync(Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] optionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	onSuccess	An optional callback that should be triggered after the request was successfully completed. You are highly encouraged to implement the callback, but it can be null.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	An optional callback that should be triggered in case the request has failed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	optionalArguments	Any optional argument that you would like to pass inside the callback.

DisableSync(Action<ClearVREvent, ClearVRPlayer>, Object[])

Disable the sync algorithm.

Declaration

void DisableSync(Action<ClearVREvent, ClearVRPlayer> argCbClearVRAsyncRequestResponseReceived,
params object[] argOptionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	argCbClearVRAsyncRequestResponseReceived	The response event handler for this call
System.Object[]	argOptionalArguments	Any optional argument that you would like to pass inside the callback.

EnableSync(SyncSettings, Action<ClearVREvent, ClearVRPlayer>, Action<ClearVREvent, ClearVRPlayer>, Object[])

Enable the sync algorithm. This is only supported for live content.

Declaration

void EnableSync(SyncSettings syncSettings, Action<ClearVREvent, ClearVRPlayer> onSuccess, Action<ClearVREvent, ClearVRPlayer> onFailure, params object[] optionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
SyncSettings	syncSettings	Configure advanced sync options.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onSuccess	An optional callback that should be triggered after the request was succesfully completed. You are highly encouraged to implement the callback, but it can be null.
Action <clearvrevent, clearvrplayer=""></clearvrevent,>	onFailure	An optional callback that should be triggered in case the request has failed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	optionalArguments	Any optional argument that you would like to pass inside the callback.

EnableSync(SyncSettings, Action<ClearVREvent, ClearVRPlayer>, Object[])

Enable the sync algorithm. This is only supported for live content.

Declaration

void EnableSync(SyncSettings argSyncSettings, Action<ClearVREvent, ClearVRPlayer>
argCbClearVRAsyncRequestResponseReceived, params object[] argOptionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
SyncSettings	argSyncSettings	Configure advanced sync options.
Action <clearvrevent, ClearVRPlayer></clearvrevent, 	argCbClearVRAsyncRequestResponseReceived	The response event handler for this call
System.Object[]	argOptionalArguments	Any optional argument that you would like to pass inside the callback.

PollSyncStatus(Action<SyncStatus, ClearVRPlayer, Object[]>, Action<ClearVRMessage, ClearVRPlayer, Object[]>, Object[])

Poll the current sync status. This can be called independently of sync being currently enabled or disabled.

Declaration

void PollSyncStatus(Action<SyncStatus, ClearVRPlayer, object[]> onSuccess, Action<ClearVRMessage,
ClearVRPlayer, object[]> onFailure, params object[] optionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
Action <syncstatus, clearvrplayer,="" system.object[]=""></syncstatus,>	onSuccess	An optional callback that should be triggered after the request was successfully completed. You are highly encouraged to implement the callback, but it can be null.

TYPE	NAME	DESCRIPTION
Action <clearvrmessage, clearvrplayer,="" system.object[]=""></clearvrmessage,>	onFailure	An optional callback that should be triggered in case the request has failed. You are highly encouraged to implement the callback, but it can be null.
System.Object[]	optionalArguments	Any optional argument that you would like to pass inside the callback.

PollSyncStatus(Action<SyncStatus, ClearVRPlayer>, Object[])

Poll the current sync status. This can be called independently of sync being currently enabled or disabled.

Declaration

void PollSyncStatus(Action<SyncStatus, ClearVRPlayer> argCbClearVRAsyncRequestResponseReceived,
params object[] argOptionalArguments)

Parameters

TYPE	NAME	DESCRIPTION
Action <syncstatus, ClearVRPlayer></syncstatus, 	argCbClearVRAsyncRequestResponseReceived	The response event handler for this call
System.Object[]	argOptionalArguments	Any optional argument that you would like to pass inside the callback.

Enum SyncModes

The mode of the synchronization algorithm

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum SyncModes : int

NAME	DESCRIPTION
PlaybackRate	The sync algorithm wil try to reach the sync target by changing the playback rate.
Seek	The sync algorithm wil try to reach the sync target by seeking inside the buffer.

Class SyncSettings

This object is used for advanced configuration of the livestream sync feature.

Inheritance

System.Object

SyncSettings

Namespace: com.tiledmedia.clearvr

Assembly: Assembly-CSharp.dll

Syntax

public class SyncSettings : object

Constructors

SyncSettings(Int32, Int32, Single, Single, Single, Boolean, SyncModes)

SyncSettings constructor

Synchronisation is enabled by passing a SyncSettings object != null into the PrepareContentParameters and SwitchContentParameters constructor. By passing null, synchronisation is disabled. LIVE ContentItems might have a *sync live edge* which is not equal to the regular *live edge* to accommodate for propagation latencies across CDNs.

Declaration

public SyncSettings(int argClientLatency = 0, int argMaxTargetLag = 0, float argMaxPlaybackRate = 1.2F, float argMinPlaybackRate = 0.8F, float argMaxPlaybackRateChange = 0.1F, bool argDisableInitialSeek = false, SyncModes syncMode = SyncModes.PlaybackRate)

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	argClientLatency	(ms) Latency the client will stay behind the content item specific live sync target. Value should be between 0 and 300000.
System.Int32	argMaxTargetLag	(ms) Maximum difference between the client sync target and the current content time. A seek will be triggered if outside this range. If set the value should be at least 5000. If unset (0) this feature is disabled.

TYPE	NAME	DESCRIPTION
System.Single	argMaxPlaybackRate	Maximum value for the playback rate the sync algorithm selects. Value should be between 1 and 1.5. If unset (0) defaults to 1.2.
System.Single	argMinPlaybackRate	Minimum value for the playback rate the sync algorithm selects. Value should be between 0.5 and 1. If unset (0) defaults to 0.8.
System.Single	argMaxPlaybackRateChange	Maximum change for the sync algorithm playback rate per second. Value should be positive. If unset (0) defaults to 0.1.
System.Boolean	argDisableInitialSeek	Disable the initial seek that might be triggered upon enabling sync. If unset defaults to false.
SyncModes	syncMode	Specifies the way sync is achieved. Supported modes: Seek and PlaybackRate. Default value: PlaybackRate.

Properties

disableInitialSeek

Declaration

public bool disableInitialSeek { get; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

maxPlaybackRate

Declaration

public float maxPlaybackRate { get; }

TYPE	DESCRIPTION
System.Single	

maxPlaybackRateChange

Declaration

public float maxPlaybackRateChange { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

maxTargetLag

Declaration

public int maxTargetLag { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

minPlaybackRate

Declaration

public float minPlaybackRate { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

syncMode

Declaration

public SyncModes syncMode { get; }

Property Value

TYPE	DESCRIPTION
SyncModes	

Methods

ToString()

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	

Enum SyncState

The state of the synchronization algorithm

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum SyncState : int

Fields

NAME	DESCRIPTION
Disabled	The sync algorithm is disabled, no synchronization actions will be taken until it is explicitely enabled.
InSync	The sync algorithm is enabled and the user is in sync.
Syncing	The sync algorithm is enabled and currently taking steps to get the user in sync.

Class SyncStateChanged

Inheritance

System.Object

SyncStateChanged

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class SyncStateChanged : object

Properties

syncState

The new sync state.

Declaration

public SyncState syncState { get; }

Property Value

TYPE	DESCRIPTION
SyncState	

Class SyncStatus

Inheritance

System.Object

SyncStatus

Namespace: com.tiledmedia.clearvr

Assembly: Assembly-CSharp.dll

Syntax

public class SyncStatus : object

Properties

actualTime

(epoch ms) Current time of the content in terms of the target time.

Declaration

public long actualTime { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	

clientLatency

(ms) Latency the client will stay behind the content item specific sync target.

Declaration

public long clientLatency { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	

maxPlaybackRate

Maximum value for the playback rate the sync algorithm selects.

Declaration

public float maxPlaybackRate { get; }

(c) Tiledmedia B.V. 2017-2022

Property Value

ТҮРЕ	DESCRIPTION
System.Single	

maxPlaybackRateChange

Maximum change for the sync algorithm playback rate per second.

Declaration

public float maxPlaybackRateChange { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

maxTargetLag

(ms) Maximum difference between the client sync target and the current content time. A seek will be triggered if outside this range.

Declaration

public long maxTargetLag { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	

minPlaybackRate

Minimum value for the playback rate the sync algorithm selects.

Declaration

public float minPlaybackRate { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

ntpTime

(epoch ms) Current network time protocol time.

Declaration

public long ntpTime { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	

optionalArguments

The optional arguments can be defined when calling the asynchronous PollSyncStatus API.

Declaration

public object[] optionalArguments { get; }

Property Value

TYPE	DESCRIPTION
System.Object[]	

playbackRate

Current content playback rate.

Declaration

public double playbackRate { get; }

Property Value

ТҮРЕ	DESCRIPTION
System.Double	

syncEdgeLatency

(ms) Current sync edge latency. Configured on the livestream. Defines the sync edge as ntp minus sync edge latency.

Declaration

public long syncEdgeLatency { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	

syncState

Indicator of the current state of sync.

Declaration

public SyncState syncState { get; }

Property Value

TYPE	DESCRIPTION
SyncState	

targetLag

(ms) Lag of the content behind the sync target, difference between target and actual.

Declaration

public long targetLag { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	

targetTime

(epoch ms) Current sync target for live content.

Declaration

public long targetTime { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	

Methods

ToString()

Declaration

public override String ToString()

ТҮРЕ	DESCRIPTION
String	

Class TelemetryConfiguration

This object contains the configuration of one or more Telemetry targets. SeeTelemetryTarget for details. You would set this on platformOptions.telemetryConfiguration

Example:

// Define telemetry target configuration (in this example for a NewRelic end-point)

TelemetryTargetConfigNewRelic telemetryTargetConfig = new

TelemetryTargetConfigNewRelic("YOUR_ACCOUNT_ID", "YOUR_LICENSE", "YOUR_END_POINT");

// Configure your telemetry target (you can have multiple)

TelemetryTarget telemetryTarget = new

TelemetryTarget(TelemetryIPSignallingTypes.TelemetryIpSignallingMasked, new

List<TelemetryTargetConfigBase>() { telemetryTargetConfig });

// Set the telemetry configuration.

platformOptions.telemetryConfiguration = new TelemetryConfiguration(new List<TelemetryTarget>() {
telemetryTarget });

Inheritance

System.Object

TelemetryConfiguration

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class TelemetryConfiguration : object

Constructors

TelemetryConfiguration(List<TelemetryTarget>, Boolean)

Configure your telemetry targets.

Declaration

public TelemetryConfiguration(List<TelemetryTarget> telemetryTargets, bool disableTiledmediaTelemetry =
false)

Parameters

TYPE	NAME	DESCRIPTION
List <telemetrytarget></telemetrytarget>	telemetryTargets	Your telemetry target(s). Can be null.

TYPE	NAME	DESCRIPTION
System.Boolean	disableTiledmediaTelemetry	Set to true if you want to disable sending telemetry data to the Tiledmedia backend. Default value: false

Properties

disableTiledmediaTelemetry

Set to true if you want to disable sending telemetry data to the Tiledmedia backend. Default value: false

Declaration

public bool disableTiledmediaTelemetry { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

disable Tiled media Telemetry To New Relic

Set to true if you want to disable sending telemetry data to the Tiledmedia New Relic backend.

Default value: false

Declaration

public bool disableTiledmediaTelemetryToNewRelic { get; set; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

telemetryTargets

Configure your Telemetry target(s). Can be null.

Declaration

public List<TelemetryTarget> telemetryTargets { get; set; }

Property Value

TYPE	DESCRIPTION
List <telemetrytarget></telemetrytarget>	

Methods

ToString()

Declaration

public override String ToString()

ТҮРЕ	DESCRIPTION
String	

Enum TelemetryIPSignallingTypes

Namespace	: com.tileamedia.clearvi
Assembly:	Assembly-CSharp.dll

Syntax

public enum TelemetryIPSignallingTypes : int
--

Fields

NAME	DESCRIPTION
TelemetryIpSignallingDisabled	
TelemetryIpSignallingFull	
TelemetryIpSignallingMasked	

Class Telemetry Target

Configure your Telemetry Target(s) of a specifictype.

Inheritance

System.Object

TelemetryTarget

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class TelemetryTarget : object

Constructors

TelemetryTarget(TelemetryIPSignallingTypes, List<TelemetryTargetConfigBase>)

Configure your Telemetry Target. You can have multiple Telemtry Targets Configurations, but each configuration must be of the same time (e.g. you can specify multiple New Relic targets).

Declaration

public TelemetryTarget(TelemetryIPSignallingTypes telemetryIPSignallingType, List<TelemetryTargetConfigBase> telemetryTargetConfigBase)

Parameters

TYPE	NAME	DESCRIPTION
TelemetryIPSignallingTypes	telemetryIPSignallingType	Whether IP adresses should signalled unmasked, masked, or not at all.
List <telemetrytargetconfigbase></telemetrytargetconfigbase>	telemetryTargetConfigBase	A list of configurations of the target. Note that all configurations must be of the same type. If you want to use different targets at the same time, you need to set multiple Targets in the TelemetryConfiguration.

Properties

telemetryIPSignallingType

Property Value

TYPE	DESCRIPTION
TelemetryIPSignallingTypes	

telemetry Target Configurations

Declaration

public List<TelemetryTargetConfigBase> telemetryTargetConfigurations { get; set; }

Property Value

TYPE	DESCRIPTION
List <telemetrytargetconfigbase></telemetrytargetconfigbase>	

Methods

ToString()

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	

Class TelemetryTargetConfigBase

Abstract base class for TelemetryTargetConfigurations. All Telemtry Targets should extend this base class.

Inheritance

System.Object

TelemetryTargetConfigBase

TelemetryTargetConfigNewRelic

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public abstract class TelemetryTargetConfigBase : object

Methods

ToString()

Declaration

public override String ToString()

ТҮРЕ	DESCRIPTION
String	

Class TelemetryTargetConfigNewRelic

The Telemetry Target configuration for New Relic metrics aggregation. See NewRelic.com for details.

Inheritance

System.Object

TelemetryTargetConfigBase

TelemetryTargetConfigNewRelic

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class TelemetryTargetConfigNewRelic: TelemetryTargetConfigBase

Constructors

TelemetryTargetConfigNewRelic(String, String, String)

Configure the NewRelic telemetry target parameters

Declaration

public TelemetryTargetConfigNewRelic(String accountID, String license, String url)

Parameters

TYPE	NAME	DESCRIPTION
String	accountID	Your NewRelic account id.
String	license	Your NewRelic license.
String	url	Your NewRelic url end-point.

Properties

accountID

Example: "3804000"

Declaration

public string accountID { get; set; }

Property Value

TYPE	DESCRIPTION
System.String	

license

Declaration

public string license { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
System.String	

url

Example: https://insights-collector.eu01.nr-data.net/v1/accounts/<ACCOUNT_ID>/events

Declaration

public string url { get; set; }

Property Value

ТҮРЕ	DESCRIPTION
System.String	

Methods

ToString()

Declaration

public override String ToString()

Returns

TYPE	DESCRIPTION
String	

Overrides

TelemetryTargetConfigBase.ToString()

Enum TelemetryTargetTypes

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum TelemetryTargetTypes : int	
--	--

Fields

NAME	DESCRIPTION
TelemetryTargetNewRelic	

Class TelemetryUpdateCustomData

Used to send custom key/value pair data to the configured Telemetry Targets. Refer to TelemetryUpdateTargetCustomData for details.

Inheritance

System.Object

TelemetryUpdateCustomData

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class TelemetryUpdateCustomData: object

Constructors

TelemetryUpdateCustomData(List<TelemetryUpdateTargetCustomData>)

Default constructor. Specify the custom data one wants to send to the telemetry target(s) of choice.

Declaration

public TelemetryUpdateCustomData(List<TelemetryUpdateTargetCustomData> targetSpecificCustomMetadata)

Parameters

TYPE	NAME	DESCRIPTION
List <telemetryupdatetargetcustomdata></telemetryupdatetargetcustomdata>	targetSpecificCustomMetadata	

Properties

telemetryUpdateTargetCustomData

The custom data that one wants to send.

Declaration

public List<TelemetryUpdateTargetCustomData> telemetryUpdateTargetCustomData { get; set; }

Property Value

TYPE	DESCRIPTION
List <telemetryupdatetargetcustomdata></telemetryupdatetargetcustomdata>	

Methods

ToString()

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	

Class TelemetryUpdateTargetCustomData

Data fields to set send custom data to the telemtry target of choice.

Inheritance

System.Object

TelemetryUpdateTargetCustomData

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class TelemetryUpdateTargetCustomData : object

Constructors

TelemetryUpdateTargetCustomData(Int32, List<KeyValuePair<String, String>>)

Construct the TelemetryUpdateTargetCustomData object.

Declaration

public TelemetryUpdateTargetCustomData(Int32 telemetryTargetIndex, List<KeyValuePair<String, String>>
customDatas)

Parameters

ТҮРЕ	NAME	DESCRIPTION
Int32	telemetryTargetIndex	The index of the telemetry target. The index matches the order in which you defined the TelemetryTargets in TelemetryConfiguration.telemetryTargets.
List <keyvaluepair<string, String>></keyvaluepair<string, 	customDatas	This list of Key/value pairs can not contain nulls.

Properties

customDatas

A set of Key/Value pairs. Key and Value are not allowed to be null. If either is set to null, the pair will be skipped.

Declaration

public List<KeyValuePair<String, String>> customDatas { get; set; }

Property Value

TYPE	DESCRIPTION
List <keyvaluepair<string, string="">></keyvaluepair<string,>	

telemetryTargetIndex

The index of the telemetry target. The index matches the order in which you defined the TelemetryTargets in TelemetryConfiguration.telemetryTargets.

Declaration

public Int32 telemetryTargetIndex { get; set; }

Property Value

TYPE	DESCRIPTION
Int32	

Methods

ToString()

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	

Enum TextureBlitModes

This enum specifies the texture blit mode, e.g. how the decoded video frame will be rendered to the screen.

A WARNING

W.r.t. TextureBlitModes.OVROverlayZeoCopy and when playing Widevine L1 protected content only:

Oculus provides two types of plugins: the traditional OVR plugin and the new OVR OpenXR plugin. Both are supported, but some versions of the traditional OVR plugin will spam logcat with the following message:

D/OVRPlugin: ovrpLayerFlag_ProtectedContent requested without a protected front-buffer.

This is a bug in this plugin as your content is being played back in a protected context (you can confirm this by taking a screenshot, your video will show up as black or green).

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum TextureBlitModes : int

Fields

NAME	DESCRIPTION
Default	Pick default option based on platform.
OVROverlayCopy	Android only. Render the video onto an OVROverlay. This mode is currently NOT supported.
OVROverlayZeroCopy	Android only. Render the video onto an OVROverlay. This mode only supports traditional (non-ClearVR) video playback like rectilinear, ERP360, ERP180 and cubemap. It does allow you to play Widevine L1 protected content on Oculus VR devices like the Quest 1 and Quest 2.
Unknown	

NAME	DESCRIPTION
UVShufflingCopy	 Classic, UV based, shuffling with texture copy. Specifically for the Android platform, the following holds: This mode is also known as "non-fast OES" mode. By default, UVShufflingZeroCopy is used on Android (see also point 3. below). This mode must be selected when using a third party player at the same time as the ClearVRPlayer. This mode must be selected when one or multiple Sprite meshes are found in the scene.
UVShufflingZeroCopy	Classic, UV based, shuffling without texture copy ("Fast OES" on Android).

Enum TextureTypes

TextureType is used to indicate how to use the planeIds (what color format is used and how the color planes are arraged)

Namespace: com.tiledmedia.clearv	r
Assembly: Assembly-CSharp.dll	

Syntax

Fields

NAME	DESCRIPTION
NV12	
RGBA	
YUV420P	

Class TimingParameters

The TimingParameters object is used when loading content, seeking and switching content. It defines the (new) start position (in milliseconds) and how this should be interpreted.

Inheritance

System.Object

TimingParameters

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class TimingParameters : object

Constructors

TimingParameters(Int64, TimingTypes)

Default constructor. For VOD content items, you can use startPosition = 0 and timingType = TimingTypes.ContentTime For LIVE content items, you can use startPosition = 0 and timingType = TimingTypes.LiveEdge to play from the live edge.

This object can also be used to control more complex content loading operations, like synchronized seamless camera switching when using the SwitchContent API.

A WARNING

When playing a live stream, the following has changed:

- 1. In pre-v8.0 versions of the ClearVR SDK you would seek to the Live Edge in a live stream regardless the SeekFlags you might have specified when calling Seek(0).
- 2. Since v8.0, in order to seek to the Live Edge, you MUST set TimingTypes.LiveEdge, e.g.: Seek(new TimingParameters(0, TimingTypes.LiveEdge)).
- 3. Seek(new TimingParameters(0, TimingTypes.ContentTime)) will seek to the earliest available content (which might not start at position 0 depending on the cache window on the Content Delivery Network).

Declaration

public TimingParameters(long argPositionInMilliseconds, TimingTypes argTimingType = default(TimingTypes))

Parameters

TYPE	NAME	DESCRIPTION
------	------	-------------

TYPE	NAME	DESCRIPTION
System.Int64	argPositionInMilliseconds	The (new) position, in milliseconds.
TimingTypes	argTimingType	How to interpret the (new) position. Default value: TimingTypes.ContentTime.

Methods

ToString()

Declaration

public override string ToString()

ТҮРЕ	DESCRIPTION
System.String	

Class TimingReport

This object holds detailed information about the position in the currently playing clip, as well as the lower and upper seek bounds. This object is especially useful when interested in the distance to the live edge in case of LIVE clip. Interpretation of the various timing fields should be based on the flag field.

IMPORTANT

The timing information on this object is only useful if the GetIsSuccess() API returns true. In case of false, the timing fields should be ignored as their values and meaning are undefined.

Inheritance

System.Object

TimingReport

Namespace: com.tiledmedia.clearvr

Assembly: Assembly-CSharp.dll

Syntax

public class TimingReport : object

Properties

clearVRMessage

The ClearVRMessage will be a GenericOK message in case of a valid report. It'sGetIsSuccess() method will return false if an error was reported. You can check the fields of the ClearVRMessage for details if that would be the case.

Declaration

public ClearVRMessage clearVRMessage { get; }

Property Value

TYPE	DESCRIPTION
ClearVRMessage	The ClearVRMessage.

contentDurationInMilliseconds

The content duration. The value is undefined if GetIsSuccess() returns false.

public long contentDurationInMilliseconds { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	A positive integer denoting the current duration.

currentPositionInMilliseconds

The current position with respect to the TimingType. The value is undefined if GetIsSuccess() returns false.

Declaration

public long currentPositionInMilliseconds { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	A positive integer denoting the current position.

eventType

Query whether this a Live, FinishedLive or VOD event. See EventTypes for details.

Declaration

public EventTypes eventType { get; }

Property Value

TYPE	DESCRIPTION
EventTypes	

flag

The SeekFlag indicates how the timing parameters on this object should be interpreted (e.g. as absolute content time or as wallclock synchronized time since epoch.) The value is undefined if GetIsSuccess() returns false.

public System.Enum flag { get; }

Property Value

TYPE	DESCRIPTION
System.Enum	The appropriate SeekFlag

lowerSeekBoundInMilliseconds

The lower bound for seeking with respect to the TimingType. For live content, this is also known as the max timeshift. The value is undefined if GetIsSuccess() returns false.

Declaration

public long lowerSeekBoundInMilliseconds { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	A positive integer denoting the earliest seekable position.

timingType

Determines how to interpret the reported values like position and upper/lower bound. See TimingTypes for details.

Declaration

public TimingTypes timingType { get; }

Property Value

TYPE	DESCRIPTION
TimingTypes	

upperSeekBoundInMilliseconds

The upper bound for seeking with respect to the TimingType. For live content, this is also known as the timeshift. The value is undefined if GetIsSuccess() returns false.

public long upperSeekBoundInMilliseconds { get; }

Property Value

TYPE	DESCRIPTION
System.Int64	A positive integer denoting the latest seekable position.

Methods

GetDistanceFromUpperBoundInMilliseconds()

The value is undefined if GetIsSuccess() returns false.

Declaration

public long GetDistanceFromUpperBoundInMilliseconds()

Returns

TYPE	DESCRIPTION
System.Int64	

GetIsSuccess()

Whether the query for the TimingReport was successfully handlded or not.

Declaration

public bool GetIsSuccess()

Returns

TYPE	DESCRIPTION
System.Boolean	true if the query was a success, false otherwise.

ToString()

Declaration

public override String ToString()

TYPE	DESCRIPTION
String	

Enum TimingTypes

Allows one to specify how the new content position in TimingParameters() should be interpreted.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum TimingTypes : int

Fields

NAME	DESCRIPTION
ContentTime	Default, interpret time as a content time.
LiveEdge	Only applicable to Live content. Indicates that the actual start position will be ignored and playback will start at the live edge.
None	Default, interpret time as a content time.
RelativeTime	Use RelativeTime to seek based on the current content position. This saves you from grabbing the current content position and adding the required offset yourself.
ScheduledOnDemand	Interpret the specified time position as the start time of a scheduled on demand content item. This applies to PrepareContentParameters and SwitchContentParameters, but cannot be used in conjunction with SeekParameters. Refer to ClearVRPlayer section Switching cameras / content for more details and an example on playing a VOD asset as ScheduledOnDemand.
Seamless	Use Seamless to seamlessly switch between two content items. This is typically used when switching between cameras in a synchronized multi-camera event. This does not apply to seeking, only switch content.
WallclockTime	Interpret the specified time position as Wallclock time. This is applicable to live broadcast streams.

Class TrackID

TrackID is an object containing the feed index and the track index of the track (video, audio or subtitle) The feed index tells you to which feed this track belongs (which can be found with clearVRPlayer.mediaInfo.getContentInfo()) The track index tells you which track this is in the list of tracks (useful when there are multiple video / audio / subtitle tracks) within the feed.

Note that this object in immutable after construction.

Inheritance

System.Object

TrackID

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class TrackID: object

Constructors

TrackID(Int32, Int32)

The default constructor. Used primarily when constructing yourLayoutParameters and configuring what audio and/or subtitle feed and track should be selected.

Declaration

public TrackID(Int32 argFeedIndex, Int32 argTrackIndex)

Parameters

TYPE	NAME	DESCRIPTION
Int32	argFeedIndex	The feed index, -2 means: automatically pick the first available feed.
Int32	argTrackIndex	The track index inside the specified feed, -2 means: automatically pick the first available track.

Properties

feedIndex

Declaration

public Int32 feedIndex { get; }

Property Value

TYPE	DESCRIPTION
Int32	

trackIndex

Declaration

public Int32 trackIndex { get; }

Property Value

TYPE	DESCRIPTION
Int32	

Methods

Equals(Object)

Declaration

public override bool Equals(object obj)

Parameters

TYPE	NAME	DESCRIPTION
System.Object	obj	

Returns

TYPE	DESCRIPTION
System.Boolean	

GetHashCode()

Declaration

public override int GetHashCode()

Returns

ТҮРЕ	DESCRIPTION
System.Int32	

ToString()

Declaration

public override String ToString()

Returns

ТҮРЕ	DESCRIPTION
String	

Enum TransitionTypes

When seeking or switching content, one can specify how this should be effectuated.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum TransitionTypes : int

NAME	DESCRIPTION
Continous	In case of a continuous transition, the current content will keep playing until the transition to the next position or content item has completed (e.g. until the internal buffers have been filled sufficiently to start playback at the new position (and new clip in case of switch content)l. This has been renamed to Continuous since v8.2.1.
Continuous	
Fast	When specifying Fast as transition type, playback will immediately halt and will only continue once the buffers at the requested content position (and in case of switch content in the new content item) have been sufficiently filled.

Class Utils

The utils class provides various convenience methods at your disposal. Public APIs on this class will maintained.

Inheritance

System.Object

Utils

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public static class Utils : object

Fields

OCULUS_XR_LOADER_NAME

Declaration

public static readonly String OCULUS_XR_LOADER_NAME

Field Value

TYPE	DESCRIPTION
String	

PICO_XR_LOADER_NAME

Declaration

public static readonly String PICO_XR_LOADER_NAME

Field Value

ТҮРЕ	DESCRIPTION
String	

SKYWORTH_XR_LOADER_NAME

Declaration

public static readonly String SKYWORTH_XR_LOADER_NAME

Field Value

ТҮРЕ	DESCRIPTION
String	

WAVE_XR_LOADER_NAME

Declaration

public static readonly String WAVE_XR_LOADER_NAME

Field Value

TYPE	DESCRIPTION
String	

Methods

ConvertMediaProjectionTypeToContentFormat(String)

This method facilitates converting a projection type represented as a string into a ContentFormat enum as supported by the ContentItem.overrideContentFormat field. This is especially useful when playing back non ClearVR 180/360/rectilinear content.

Declaration

public static ContentFormat ConvertMediaProjectionTypeToContentFormat(String argMediaProjectionType)

Parameters

TYPE	NAME	DESCRIPTION
String	argMediaProjectionType	The media projection type as a canonical string

Returns

TYPE	DESCRIPTION
ContentFormat	The matching ContentFormat, or ContentFormat.Unknown if the media projection type is not supported.

GetDeviceType()

Returns the detected Device Type. Auto-detection is only performed during the first time you call this API. If you want to re-detect the Device Type, or override the automatically detected Device Type,

see RedetectDeviceType().

Declaration

public static DeviceTypes GetDeviceType()

Returns

TYPE	DESCRIPTION
DeviceTypes	The current DeviceType

GetIsVrDevicePresent()

Declaration

public static bool GetIsVrDevicePresent()

Returns

TYPE	DESCRIPTION
System.Boolean	

GetListenerCount(UnityEngine.Events.UnityEventBase)

Declaration

public static int GetListenerCount(this UnityEngine.Events.UnityEventBase unityEvent)

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Events.UnityEventBase	unityEvent	

Returns

TYPE	DESCRIPTION
System.Int32	

GetType(String)

Declaration

public static Type GetType(string argTypeName)

Parameters

TYPE	NAME	DESCRIPTION
System.String	argTypeName	

Returns

TYPE	DESCRIPTION
Туре	

GetVRAPIType()

Declaration

public static VRAPITypes GetVRAPIType()

Returns

TYPE	DESCRIPTION
VRAPITypes	

PrintCurrentStackTrace()

Declaration

public static void PrintCurrentStackTrace()

RedetectDeviceType(DeviceTypes)

Redetect the device type. Call this API if you switched to a different mode (e.g. from traditional flat to cardboard). Use Utils.GetDeviceType() to get the newly detected device type. Notes:

- 1. This is a SLOW and EXPENSIVE call. We highly recommend you to query once and cache your result afterwards.
- 2. This does NOT return the same value as UnityEngine.XR.XRSettings.loadedDeviceName or UnityEngine.XR.XRDevice.system. Some VR platforms, notably WaveVR, do not expose any usable values on these Unity specific APIs.
- 3. This could theoretically change on run-time, for example if you would switch from flat (None) to cardboard (Cardboard) mode.

Declaration

public static void RedetectDeviceType(DeviceTypes argOverrideDeviceType = default(DeviceTypes))

Parameters

TYPE	NAME	DESCRIPTION
DeviceTypes	argOverrideDeviceType	

Enum VideoCodecProfiles

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum VideoCodecProfiles: int

NAME	DESCRIPTION
Av1High	
Av1Main	
Av1Professional	
H264Baseline	
H264ConstrainedBaseline	
H264ConstrainedHigh	
H264ExtendedProfile	
H264High	
H264High10	
H264High422	
H264High444	
H264Main	
H265Main	
H265Main10	
H265Main10Hdr10	

NAME	DESCRIPTION
H265Main10Hdr10Plus	
H265Main12	
H265MainStill	
Unknown	

Enum VideoCodecTypes

Namespace	:: com.tileamedia.clearv
Assembly:	Assembly-CSharp.dll

Syntax

public enum VideoCodecTypes : int

NAME	DESCRIPTION
Av1	
H264	
H265	
Unspecified	

Class VideoDecoderCapabilities

Helper object that is of little use to the integrator. Allows one to parse the

ClearVREventTypes.GenericMessage with

ClearVRMessageCodes.ClearVRCoreWrapperVideoDecoderCapabilities. This cotntains some information about the reported video decoder capabilities. Notably, it can be used to query the video decoder level

Inheritance

System.Object

VideoDecoderCapabilities

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class VideoDecoderCapabilities : object

Constructors

VideoDecoderCapabilities(String, String, String, String, String)

Declaration

protected VideoDecoderCapabilities(String argVideoDecoderName, String argMimetype, String argDecoderMaximumCodecLevelSupported, String argClearVRSdkMaximumCodecLevelSupported, String argMaximumCodecLevelInUse)

Parameters

TYPE	NAME	DESCRIPTION
String	argVideoDecoderName	
String	argMimetype	
String	argDecoderMaximumCodecLevelSupported	
String	argClearVRSdkMaximumCodecLevelSupported	
String	argMaximumCodecLevelInUse	

Properties

clearVRSdkMaximumCodecLevelSupported

The maximum codec level supported by the SDK, e.g. 5.2

(c) Tiledmedia B.V. 2017-2022

6 NOTE

In the past, the Android SDK supported HEVC decoders up to level 5.2. This limitation has since been lifted.

Declaration

public String clearVRSdkMaximumCodecLevelSupported { get; }

Property Value

ТҮРЕ	DESCRIPTION
String	The level

decoderMaximumCodecLevelSupported

The maximum codec level supported by the decoder, e.g. 5.2

Declaration

public String decoderMaximumCodecLevelSupported { get; }

Property Value

ТҮРЕ	DESCRIPTION
String	The level

maximumCodecLevelInUse

The codec level in use (= Math.min(maxSupportedByDecoder, maxSupportedBySDK))

Declaration

public String maximumCodecLevelInUse { get; }

Property Value

TYPE	DESCRIPTION
String	The level

mimetype

The mimetype of the video that is currently being decoded by this codec. Examples: video/hevc, video/avc

Declaration

public String mimetype { get; }

Property Value

TYPE	DESCRIPTION
String	The mimetype

videoDecoderName

Reported video decoder name. On Android, this will return a String that holds the MediaCodec decoder name, e.g. OMX.dec.hevc

Declaration

public String videoDecoderName { get; }

Property Value

TYPE	DESCRIPTION
String	The name as a canonical string.

Enum VideoStereoMode

Devices the type of a mesh. This allows you to infer the actual shape of the mesh

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum VideoStereoMode: int

NAME	DESCRIPTION
Mono	Monoscipic content
StereoBottomTop	Stereo right eye top, left eye bottom.
StereoComplexMapping	Stereo with a complex UV mapping
StereoSideBySide	Stereo left eye left, right eye right.
StereoTopBottom	Stereo left eye top, right eye bottom.
Unknown	Unknown stereo type, not allowed.

Class VideoTrackInfo

Helper class that holds information of one video track, like codec, dimensions and framerate. For detailed information about the ContentInfo, FeedInfo, VideoTrackInfo, AudioTrackInfo, SubtitleTrackInfo objects and their relationship, refer to the ContentItem documentation.

Inheritance

System.Object VideoTrackInfo

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class VideoTrackInfo: object

Properties

aspectRatio

Since v9.1.2 The aspect ratio is defined as width / height. If height = 0, this returns 0.

Declaration

public float aspectRatio { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

bitrateInKbps

Get the bitrate of this track in kilobits per second (kbps), as advertised in the manifest

1 NOTE

This will return 0 if no bitrate is advertised in the manifest file.

A WARNING

For ClearVR content, this will return the advertised bitrate for the entire (spatially segmented) video. The value should *not* be used as an indication of the required internet connectivity as only a small part of the entire video is ever fetched over the network.

Declaration

public Int32 bitrateInKbps { get; }

Property Value

TYPE	DESCRIPTION
Int32	

contentFormat

The content format of the video as ContentFormat.

Declaration

public ContentFormat { get; }

Property Value

TYPE	DESCRIPTION
ContentFormat	

drmType

The type of DRM used in this video track as DRMTypes.

Declaration

public DRMTypes drmType { get; }

Property Value

TYPE	DESCRIPTION
DRMTypes	

feedIndex

The Feed index associated with this VideoTrack.

Declaration

public int feedIndex { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

framerate

Get the framerate as a fraction. This is the same as calling framerateNum / framerateDenom yourself

Declaration

public float framerate { get; }

Property Value

TYPE	DESCRIPTION
System.Single	

framerateDenom

The denominator of the framerate integer fraction. framerate is a fraction: fps = num / denom

Declaration

public int framerateDenom { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

framerateNum

The numerator of the framerate integer fraction. The framerate is an integer fraction: fps = num / denom

Declaration

public int framerateNum { get; }

Property Value

ТҮРЕ	DESCRIPTION
System.Int32	

height

The height of the video as int.

Declaration

public int height { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

isActive

Whether this video track is active or not.

Declaration

public bool isActive { get; }

Property Value

TYPE	DESCRIPTION
System.Boolean	

supportedStatus

Whether the VideoTrack is supported by the device or not. This can be Unknown. Refer to ContentSupportedStatus for more more information.

Declaration

public ContentSupportedStatus supportedStatus { get; }

Property Value

ТҮРЕ	DESCRIPTION
ContentSupportedStatus	

trackID

Returns the TrackID, a unique identifier for this track and the FeedInfo it belongs to. It can be used when constructing your FeedConfiguration.

Declaration

public TrackID trackID { get; }

Property Value

TYPE	DESCRIPTION
TrackID	The TracKID, it cannot be null.

trackIndex

The index of the video track as int.

Declaration

public int trackIndex { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

url

Get the URL

Declaration

public String url { get; }

Property Value

ТҮРЕ	DESCRIPTION
String	

videoCodecType

Get the video codec type.

Declaration

public VideoCodecTypes videoCodecType { get; }

Property Value

TYPE	DESCRIPTION
VideoCodecTypes	

width

The width of the video as int.

Declaration

public int width { get; }

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

GetQualityDescriptor()

Returns a pretty formatted quality descriptor string, in the format of [width]x[height]p[framerate]. The framerate is truncated to three decimals.

Declaration

public String GetQualityDescriptor()

Returns

TYPE	DESCRIPTION
String	The pretty-printed quality descriptor.

ToString()

Declaration

public override string ToString()

Returns

TYPE	DESCRIPTION
System.String	

Enum VRAPITypes

VRAPITypes enum, currently, only OculusVR is supported.

A WARNING

This enum is of no use to the integrator and might repurposed, changed and or removed without notice. Do not use.

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public enum VRAPITypes: int

NAME	DESCRIPTION
OculusVR	
OpenVR	
Unknown	

Class VRKeyboard

Inheritance

System.Object

VRKeyboard

Namespace: com.tiledmedia.clearvr

Assembly: Assembly-CSharp.dll

Syntax

public class VRKeyboard : MonoBehaviour

Fields

ContentColor

Declaration

public Color ContentColor

Field Value

TYPE	DESCRIPTION
Color	

currentSelectedInputField

Declaration

public TMP_InputField currentSelectedInputField

Field Value

TYPE	DESCRIPTION
TMP_InputField	

hoverScaleMultiplier

Declaration

public float hoverScaleMultiplier

Field Value

ТҮРЕ	DESCRIPTION
System.Single	

Keys

Declaration

public List<VRKey> Keys

Field Value

TYPE	DESCRIPTION
List <vrkey></vrkey>	

KeysColor

Declaration

public Color KeysColor

Field Value

TYPE	DESCRIPTION
Color	

transitionTime

Declaration

public float transitionTime

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

BackSpace()

Declaration

public void BackSpace()

Confirm(Button)

(c) Tiledmedia B.V. 2017-2022

Declaration

public void Confirm(Button argConfirmButton)

Parameters

TYPE	NAME	DESCRIPTION
Button	argConfirmButton	

PrintChar(String)

Declaration

public void PrintChar(string argContent)

Parameters

TYPE	NAME	DESCRIPTION
System.String	argContent	

SelectInputField(TMP_InputField)

Declaration

public void SelectInputField(TMP_InputField argNewInputField)

Parameters

ТҮРЕ	NAME	DESCRIPTION
TMP_InputField	argNewInputField	

SelectInputFieldAndShowKeyboard(TMP_InputField)

Declaration

public void SelectInputFieldAndShowKeyboard(TMP_InputField argNewInputField)

Parameters

TYPE	NAME	DESCRIPTION
TMP_InputField	argNewInputField	

Shift()

Declaration

public void Shift()

SpecialCharKeyboardSwitch()

Declaration

public void SpecialCharKeyboardSwitch()

ToggleKeyboard(Boolean)

Declaration

public void ToggleKeyboard(bool argShowKeyboard)

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	argShowKeyboard	

UnselectInputFieldAndHideKeyboard()

Declaration

public void UnselectInputFieldAndHideKeyboard()

Class VRTagalong

Inheritance

System.Object VRTagalong

Namespace: com.tiledmedia.clearvr Assembly: Assembly-CSharp.dll

Syntax

public class VRTagalong: MonoBehaviour