

# Table of Contents

Uralstech.UXR.QuestCamera .....	2
CameraDevice .....	4
CameraDevice.ErrorCode .....	10
CameraFrameForwarder .....	11
CameraInfo .....	13
CameraInfo.CameraEye .....	17
CameraInfo.CameraIntrinsics .....	18
CameraInfo.CameraSource .....	21
CameraSupport .....	22
CaptureSessionObject<T> .....	24
CaptureTemplate .....	26
ContinuousCaptureSession .....	27
NativeWrapperState .....	32
OnDemandCaptureSession .....	33
UCameraManager .....	35
YUVToRGBAConverter .....	38

# Namespace Uralstech.UXR.QuestCamera

## Classes

### [CameraDevice](#)

A wrapper for a native Camera2 CameraDevice.

### [CameraFrameForwarder](#)

Forwards frame callbacks from the native Kotlin plugin to Unity.

### [CameraInfo](#)

Wrapper for Camera2's CameraCharacteristics.

### [CameraSupport](#)

Utility to check if the current Meta Quest device supports the Passthrough Camera API.

### [CaptureSessionObject<T>](#)

Simple class for grouping capture session related components to their GameObject.

### [ContinuousCaptureSession](#)

A wrapper for a native Camera2 CaptureSession and ImageReader.

### [OnDemandCaptureSession](#)

A wrapper for a native Camera2 CaptureSession and ImageReader.

### [UCameraManager](#)

Class for interfacing with the native Camera2 API on Android.

### [YUVToRGBAConverter](#)

The default YUV 4:2:0 to RGBA converter that uses a compute shader to convert the camera texture to RGBA.

## Structs

### [CameraInfo.CameraIntrinsics](#)

Defines the camera's intrinsic properties. All values are in pixels.

## Enums

### [CameraDevice.ErrorCode](#)

Error codes that can be returned by the native CameraDevice wrapper.

### [CameraInfo.CameraEye](#)

The camera eye.

### [CameraInfo.CameraSource](#)

The source of the camera feed.

### [CaptureTemplate](#)

Capture template to use when recording.

### [NativeWrapperState](#)

The current assumed state of a native wrapper.

# Class CameraDevice

Namespace: [Uralstech.UXR.QuestCamera](#)

A wrapper for a native Camera2 CameraDevice.

```
public class CameraDevice : MonoBehaviour
```

## Inheritance

object ← CameraDevice

## Fields

### OnDeviceClosed

Invoked when the CameraDevice is closed.

```
public UnityEvent OnDeviceClosed
```

Field Value

UnityEvent

### OnDeviceDisconnected

Invoked when the CameraDevice is disconnected.

```
public UnityEvent OnDeviceDisconnected
```

Field Value

UnityEvent

### OnDeviceErred

Invoked when the CameraDevice encounters an error.

```
public UnityEvent<CameraDevice.ErrorCode> OnDeviceErred
```

Field Value

UnityEvent<[ErrorCode](#)>

## OnDeviceOpened

Invoked when the CameraDevice is opened.

```
public UnityEvent OnDeviceOpened
```

Field Value

UnityEvent

## Properties

### CameraId

The ID of the camera being wrapped.

```
public string CameraId { get; }
```

Property Value

string

### CurrentState

The current assumed state of the native CameraDevice wrapper.

```
public NativeWrapperState CurrentState { get; }
```

Property Value

[NativeWrapperState](#)

## IsActiveAndUsable

Is the native CameraDevice wrapper active and usable?

```
public bool IsActiveAndUsable { get; }
```

Property Value

bool

## Methods

### CreateContinuousCaptureSession(Resolution, CaptureTemplate)

Creates a new repeating/continuous capture session for use.

```
public CaptureSessionObject<ContinuousCaptureSession>  
CreateContinuousCaptureSession(Resolution resolution, CaptureTemplate captureTemplate  
= CaptureTemplate.Preview)
```

Parameters

**resolution** Resolution

The resolution of the capture.

**captureTemplate** [CaptureTemplate](#)

The capture template to use for the capture

Returns

[CaptureSessionObject](#)<[ContinuousCaptureSession](#)>

A new capture session wrapper. May be null if the current camera device is not usable.

## Remarks

Once you have finished using the capture session, call [Destroy\(\)](#) to close the session and free up native and compute shader resources.

## CreateOnDemandCaptureSession(Resolution)

Creates a new on-demand capture session for use.

```
public CaptureSessionObject<OnDemandCaptureSession> CreateOnDemandCaptureSession(Resolution resolution)
```

## Parameters

**resolution** Resolution

The resolution of the capture.

## Returns

[CaptureSessionObject](#)<[OnDemandCaptureSession](#)>

A new capture session wrapper. May be null if the current camera device is not usable.

## Remarks

Once you have finished using the capture session, call [Destroy\(\)](#) to close the session and free up native and compute shader resources.

## Destroy()

Releases the CameraDevice's native resources, and destroys its GameObject.

```
public void Destroy()
```

## OnDestroy()

```
protected void OnDestroy()
```

## Release()

Releases the CameraDevice's native resources, and makes it unusable.

```
public void Release()
```

## WaitForInitialization()

Waits until the CameraDevice is open or erred out.

```
public IEnumerator WaitForInitialization()
```

Returns

`IEnumerator`

## WaitForInitializationAsync()

Waits until the CameraDevice is open or erred out.

```
public Awaitable<NativeWrapperState> WaitForInitializationAsync()
```

Returns

`Awaitable<NativeWrapperState>`

The current state of the CameraDevice.

Remarks

Requires Unity 6.0 or higher.



## \_onDeviceClosed(string)

```
public void _onDeviceClosed(string _)
```

### Parameters

\_ string

## \_onDeviceDisconnected(string)

```
public void _onDeviceDisconnected(string _)
```

### Parameters

\_ string

## \_onDeviceErred(string)

```
public void _onDeviceErred(string errorCodeStr)
```

### Parameters

errorCodeStr string

## \_onDeviceOpened(string)

```
public void _onDeviceOpened(string _)
```

### Parameters

\_ string

# Enum CameraDevice.ErrorCode

Namespace: [Uralstech.UXR.QuestCamera](#)

Error codes that can be returned by the native CameraDevice wrapper.

```
public enum CameraDevice.ErrorCode
```

## Fields

**CameraAccessException = 1000**

The native code encountered a CameraAccessException.

**CameraDeviceError = 4**

The camera device has encountered a fatal error.

**CameraDisabled = 3**

The camera device could not be opened due to a device policy.

**CameraInUse = 1**

The camera device is in use already.

**CameraServiceError = 5**

The camera service has encountered a fatal error.

**MaxCamerasInUse = 2**

The camera device could not be opened because there are too many other open camera devices.

**SecurityException = 1001**

The native code encountered a SecurityException.

**Unknown = 0**

Unknown error.

# Class CameraFrameForwarder

Namespace: [Uralstech.UXR.QuestCamera](#)

Forwards frame callbacks from the native Kotlin plugin to Unity.

```
public class CameraFrameForwarder : AndroidJavaProxy
```

## Inheritance

object ← CameraFrameForwarder

## Constructors

### CameraFrameForwarder()

```
public CameraFrameForwarder()
```

## Fields

### OnFrameReady

Callback for processing the YUV 4:2:0 frame.

```
public Action<nint, nint, nint, int, int, int, int, int, int> OnFrameReady
```

### Field Value

Action<nint, nint, nint, int, int, int, int, int, int>

### Remarks

Parameters	
yBuffer (IntPtr)	Pointer to the buffer containing Y (luminance) data of the frame.
uBuffer (IntPtr)	Pointer to the buffer containing U (color) data of the frame.

Parameters	
<b>vBuffer (IntPtr)</b>	Pointer to the buffer containing V (color) data of the frame.
<b>ySize (int)</b>	The size of yBuffer.
<b>uSize (int)</b>	The size of uBuffer.
<b>vSize (int)</b>	The size of vBuffer.
<b>yRowStride (int)</b>	The size of each row of the image in yBuffer in bytes.
<b>uvRowStride (int)</b>	The size of each row of the image in uBuffer and vBuffer in bytes.
<b>uvPixelStride (int)</b>	The size of a pixel in a row of the image in uBuffer and vBuffer in bytes.

## Methods

### GetBufferPointer(AndroidJavaObject)

Gets the pointer to a native buffer from a Java ByteBuffer object.

```
protected static nint GetBufferPointer(AndroidJavaObject byteBuffer)
```

#### Parameters

**byteBuffer** AndroidJavaObject

The Java ByteBuffer object.

#### Returns

nint

A pointer to the native buffer.

# Class CameraInfo

Namespace: [Uralstech.UXR.QuestCamera](#)

Wrapper for Camera2's CameraCharacteristics.

```
public class CameraInfo
```

## Inheritance

object ← CameraInfo

## Constructors

### CameraInfo(AndroidJavaObject)

```
public CameraInfo(AndroidJavaObject cameraInfo)
```

## Parameters

**cameraInfo** AndroidJavaObject

## Properties

### CameraId

The actual device ID of this camera.

```
public string CameraId { get; }
```

## Property Value

string

## Eye

(Meta Quest) The eye which the camera is closest to.

```
public CameraInfo.CameraEye Eye { get; }
```

Property Value

[CameraInfo.CameraEye](#)

## Intrinsics

The intrinsics for this camera.

```
public CameraInfo.CameraIntrinsics Intrinsics { get; }
```

Property Value

[CameraInfo.CameraIntrinsics](#)

## LensPoseRotation

The orientation of the camera relative to the sensor coordinate system.

```
public Quaternion LensPoseRotation { get; }
```

Property Value

Quaternion

## LensPoseTranslation

The position of the camera optical center.

```
public Vector3 LensPoseTranslation { get; }
```

Property Value

Vector3

## NativeCameraCharacteristics

The native CameraCharacteristics object.

```
public AndroidJavaObject NativeCameraCharacteristics { get; }
```

Property Value

AndroidJavaObject

## Source

(Meta Quest) The source of the camera feed.

```
public CameraInfo.CameraSource Source { get; }
```

Property Value

[CameraInfo.CameraSource](#)

## SupportedResolutions

The resolutions supported by this camera.

```
public Resolution[] SupportedResolutions { get; }
```

Property Value

Resolution[]

## Methods

ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

string

A string that represents the current object.

## Operators

implicit operator string(CameraInfo)

```
public static implicit operator string(CameraInfo camera)
```

Parameters

camera [CameraInfo](#)

Returns

string



# Enum CameraInfo.CameraEye

Namespace: [Uralstech.UXR.QuestCamera](#)

The camera eye.

```
public enum CameraInfo.CameraEye
```

## Fields

Left = 1

The leftmost camera.

Right = 2

The rightmost camera.

Unknown = 0

Unknown.

# Struct CameraInfo.CameraIntrinsics

Namespace: [Uralstech.UXR.QuestCamera](#)

Defines the camera's intrinsic properties. All values are in pixels.

```
public readonly struct CameraInfo.CameraIntrinsics
```

## Constructors

### CameraIntrinsics(Vector2, Vector2, Vector2, float)

```
public CameraIntrinsics(Vector2 resolution, Vector2 focalLength, Vector2 principalPoint, float skew)
```

#### Parameters

**resolution** Vector2

**focalLength** Vector2

**principalPoint** Vector2

**skew** float

## Fields

### FocalLength

Focal length in pixels.

```
public readonly Vector2 FocalLength
```

#### Field Value

Vector2

# PrincipalPoint

Principal point in pixels from the image's top-left corner.

```
public readonly Vector2 PrincipalPoint
```

Field Value

Vector2

# Resolution

Resolution in pixels.

```
public readonly Vector2 Resolution
```

Field Value

Vector2

# Skew

Skew coefficient for axis misalignment.

```
public readonly float Skew
```

Field Value

float

# Methods

## ToString()

Returns the fully qualified type name of this instance.

```
public override string ToString()
```

## Returns

string

The fully qualified type name.

# Enum CameraInfo.CameraSource

Namespace: [Uralstech.UXR.QuestCamera](#)

The source of the camera feed.

```
public enum CameraInfo.CameraSource
```

## Fields

**PassthroughRGB = 1**

Meta Quest Passthrough RGB cameras.

**Unknown = 0**

Unknown.

# Class CameraSupport

Namespace: [Uralstech.UXR.QuestCamera](#)

Utility to check if the current Meta Quest device supports the Passthrough Camera API.

```
public static class CameraSupport
```

## Inheritance

object ← CameraSupport

## Remarks

Requires the Meta XR Core SDK.

## Fields

### MINSUPPORTOSVERSION

```
public const int MINSUPPORTOSVERSION = 74
```

Field Value

int

## Properties

### HorizonOSVersion

Get the Horizon OS version number on the headset

```
public static int? HorizonOSVersion { get; }
```

Property Value

int?

## Remarks

Requires the Meta XR Core SDK.

## IsSupported

Returns true if the current headset supports Passthrough Camera API

```
public static bool IsSupported { get; }
```

## Property Value

bool

## Remarks

Requires the Meta XR Core SDK.

# Class CaptureSessionObject<T>

Namespace: [Uralstech.UXR.QuestCamera](#)

Simple class for grouping capture session related components to their GameObject.

```
public class CaptureSessionObject<T> where T : ContinuousCaptureSession
```

## Type Parameters

T

### Inheritance

object ← CaptureSessionObject<T>

## Fields

### CameraFrameForwarder

The camera frame forwarder.

```
public readonly CameraFrameForwarder CameraFrameForwarder
```

## Field Value

[CameraFrameForwarder](#)

## Remarks

You can add additional [YUVToRGBAConverter](#)s to this to have multiple streams of the same capture session.

### CaptureSession

The capture session wrapper.

```
public readonly T CaptureSession
```



Field Value

T

## GameObject

The GameObject containing the [CaptureSession](#) and [TextureConverter](#) components.

```
public readonly GameObject GameObject
```

Field Value

GameObject

## TextureConverter

The YUV to RGBA texture converter.

```
public readonly YUVToRGBAConverter TextureConverter
```

Field Value

[YUVToRGBAConverter](#)

## Methods

### Destroy()

Destroys the GameObject to release all native resources.

```
public void Destroy()
```

# Enum CaptureTemplate

Namespace: [Uralstech.UXR.QuestCamera](#)

Capture template to use when recording.

```
public enum CaptureTemplate
```

## Fields

**Default = 0**

Default value, do not use.

**Preview = 1**

Creates a request suitable for a camera preview window.

**Record = 3**

Creates a request suitable for video recording.

**StillCapture = 2**

Creates a request suitable for still image capture.

**VideoSnapshot = 4**

Creates a request suitable for still image capture while recording video.

**ZeroShutterLag = 5**

Creates a request suitable for zero shutter lag still capture.

# Class ContinuousCaptureSession

Namespace: [Uralstech.UXR.QuestCamera](#)

A wrapper for a native Camera2 CaptureSession and ImageReader.

```
public class ContinuousCaptureSession : MonoBehaviour
```

## Inheritance

object ← ContinuousCaptureSession

## Derived

[OnDemandCaptureSession](#)

## Remarks

This is different from [OnDemandCaptureSession](#) as it returns a continuous stream of images.

## Fields

### OnSessionConfigurationFailed

Called when the session could not be configured.

```
public UnityEvent<string> OnSessionConfigurationFailed
```

Field Value

UnityEvent<string>

### OnSessionConfigured

Called when the session has been configured.

```
public UnityEvent OnSessionConfigured
```

Field Value

UnityEvent

## OnSessionRequestFailed

Called when the session request could not be set.

```
public UnityEvent<string> OnSessionRequestFailed
```

Field Value

UnityEvent<string>

## OnSessionRequestSet

Called when the session request has been set.

```
public UnityEvent OnSessionRequestSet
```

Field Value

UnityEvent

## \_captureSession

The native capture session object.

```
protected AndroidJavaObject _captureSession
```

Field Value

AndroidJavaObject

## Properties

CurrentState

The current assumed state of the native CaptureSession wrapper.

```
public NativeWrapperState CurrentState { get; }
```

Property Value

[NativeWrapperState](#)

## IsActiveAndUsable

Is the native CaptureSession wrapper active and usable?

```
public bool IsActiveAndUsable { get; }
```

Property Value

bool

## Methods

### OnDestroy()

```
protected void OnDestroy()
```

### Release()

Releases the CaptureSession's native resources, and makes it unusable.

```
public void Release()
```

### WaitForInitialization()

Waits until the CaptureSession is open or erred out.

```
public IEnumerator WaitForInitialization()
```

Returns

IEnumerator

## WaitForInitializationAsync()

Waits until the CaptureSession is open or erred out.

```
public Awaitable<NativeWrapperState> WaitForInitializationAsync()
```

Returns

Awaitable<[NativeWrapperState](#)>

The current state of the CaptureSession.

Remarks

Requires Unity 6.0 or higher.

## \_onSessionConfigurationFailed(string)

```
public void _onSessionConfigurationFailed(string reason)
```

Parameters

**reason** string

## \_onSessionConfigured(string)

```
public void _onSessionConfigured(string _)
```

Parameters

`_` string

## `_onSessionRequestFailed(string)`

```
public void _onSessionRequestFailed(string reason)
```

### Parameters

`reason` string

## `_onSessionRequestSet(string)`

```
public void _onSessionRequestSet(string _)
```

### Parameters

`_` string

# Enum NativeWrapperState

Namespace: [Uralstech.UXR.QuestCamera](#)

The current assumed state of a native wrapper.

```
public enum NativeWrapperState
```

## Fields

**Closed = 2**

The native wrapper failed with an error, was disconnected or was closed normally.

**Initializing = 0**

The native wrapper is still initializing.

**Opened = 1**

The native wrapper is open and ready.



# Class OnDemandCaptureSession

Namespace: [Uralstech.UXR.QuestCamera](#)

A wrapper for a native Camera2 CaptureSession and ImageReader.

```
public class OnDemandCaptureSession : ContinuousCaptureSession
```

## Inheritance

object ← [ContinuousCaptureSession](#) ← OnDemandCaptureSession

## Inherited Members

[ContinuousCaptureSession.CurrentState](#) , [ContinuousCaptureSession.IsActiveAndUsable](#) ,  
[ContinuousCaptureSession.OnSessionConfigured](#) ,  
[ContinuousCaptureSession.OnSessionConfigurationFailed](#) ,  
[ContinuousCaptureSession.OnSessionRequestSet](#) , [ContinuousCaptureSession.OnSessionRequestFailed](#) ,  
[ContinuousCaptureSession.captureSession](#) , [ContinuousCaptureSession.OnDestroy\(\)](#) ,  
[ContinuousCaptureSession.WaitForInitialization\(\)](#) ,  
[ContinuousCaptureSession.WaitForInitializationAsync\(\)](#) , [ContinuousCaptureSession.Release\(\)](#) ,  
[ContinuousCaptureSession.onSessionConfigured\(string\)](#) ,  
[ContinuousCaptureSession.onSessionConfigurationFailed\(string\)](#) ,  
[ContinuousCaptureSession.onSessionRequestSet\(string\)](#) ,  
[ContinuousCaptureSession.onSessionRequestFailed\(string\)](#).

## Remarks

This is different from [ContinuousCaptureSession](#) as it only returns a frame from the native plugin when required. This is recommended for single-image capturing or on-demand capturing where you don't need a continuous stream of images.

Why does [OnDemandCaptureSession](#) inherit from [ContinuousCaptureSession](#)? Because under the hood, both do the same thing - a repeating capture session. A true on-demand capture results in a black image, so [OnDemandCaptureSession](#) runs a repeating capture request running on an dummy texture natively, and reads the actual image through an ImageReader only when requested to do so. This means that while the [ContinuousCaptureSession](#) processes each and every frame sent to it, converting it to RGBA, [OnDemandCaptureSession](#) only does it when required.

## Methods

# RequestCapture(CaptureTemplate)

Requests a new capture from the session.

```
public bool RequestCapture(CaptureTemplate captureTemplate = CaptureTemplate.StillCapture)
```

## Parameters

**captureTemplate** [CaptureTemplate](#)

The capture template to use for the capture

## Returns

**bool**

If the capture request was set successfully, [true](#)<sup>↗</sup>, otherwise, [false](#)<sup>↗</sup>.

# Class UCameraManager

Namespace: [Uralstech.UXR.QuestCamera](#)

Class for interfacing with the native Camera2 API on Android.

```
public class UCameraManager : DontCreateNewSingleton<UCameraManager>
```

## Inheritance

object ← UCameraManager

## Fields

### HeadsetCameraPermission

The permission required to access the Meta Quest's cameras.

```
public const string HeadsetCameraPermission = "horizonos.permission.HEADSET_CAMERA"
```

### Field Value

string

### YUVToRGBAComputeShader

The compute shader to use to convert the camera's YUV 4:2:0 images to RGBA.

```
public ComputeShader YUVToRGBAComputeShader
```

### Field Value

ComputeShader

## Properties

# Cameras

Returns all available cameras and their characteristics. This is a cached value.

```
public CameraInfo[] Cameras { get; }
```

Property Value

[CameraInfo\[\]](#)

## Methods

### Awake()

```
protected override void Awake()
```

### GetCamera(CameraEye)

Gets a camera device by the eye it is closest to.

```
public CameraInfo GetCamera(CameraInfo.CameraEye eye)
```

Parameters

eye [CameraInfo.CameraEye](#)

The eye.

Returns

[CameraInfo](#)

The camera's [CameraInfo](#), [null](#)  if not found.

### OnDestroy()

```
protected void OnDestroy()
```

## OpenCamera(string)

Opens a camera device for use.

```
public CameraDevice OpenCamera(string camera)
```

### Parameters

**camera** string

The ID of the camera. You can get it from [Cameras](#) or [GetCamera\(CameraEye\)](#).

### Returns

[CameraDevice](#)

A new camera device wrapper. May be null if the current object is disposed/unusable.

### Remarks

Once you have finished using the camera, either destroy its GameObject or call [Release\(\)](#) to close the camera and free up native resources.

# Class YUVToRGBAConverter

Namespace: [Uralstech.UXR.QuestCamera](#)

The default YUV 4:2:0 to RGBA converter that uses a compute shader to convert the camera texture to RGBA.

```
public class YUVToRGBAConverter : MonoBehaviour
```

## Inheritance

object ← YUVToRGBAConverter

## Fields

### OnFrameProcessed

Called when a frame has been converted from YUV 4:2:0 to RGBA.

```
public UnityEvent<RenderTexture> OnFrameProcessed
```

## Field Value

UnityEvent<RenderTexture>

## Shader

The shader used to convert YUV 4:2:0 to an RGBA RenderTexture. Uses [YUVToRGBAComputeShader](#) if not specified here.

```
public ComputeShader Shader
```

## Field Value

ComputeShader

## \_uComputeBuffer

Pointer to the buffer containing U (color) data of the frame being processed.

```
protected ComputeBuffer _uComputeBuffer
```

Field Value

ComputeBuffer

## \_vComputeBuffer

Pointer to the buffer containing V (color) data of the frame being processed.

```
protected ComputeBuffer _vComputeBuffer
```

Field Value

ComputeBuffer

## \_yComputeBuffer

Pointer to the buffer containing Y (luminance) data of the frame being processed.

```
protected ComputeBuffer _yComputeBuffer
```

Field Value

ComputeBuffer

## Properties

### CameraFrameForwarder

The native camera frame forwarder.

```
public CameraFrameForwarder CameraFrameForwarder { get; }
```

Property Value

[CameraFrameForwarder](#)

## FrameRenderTexture

The RenderTexture which will contain the RGBA camera frames.

```
public RenderTexture FrameRenderTexture { get; }
```

Property Value

RenderTexture

## Methods

### Awake()

```
protected void Awake()
```

### CopyNativeDataToComputeBuffer(ref ComputeBuffer, nint, int)

Copies native (unmanaged) byte data to a compute buffer.

```
protected static void CopyNativeDataToComputeBuffer(ref ComputeBuffer computeBuffer, nint  
nativeBufferPtr, int nativeBufferSize)
```

Parameters

**computeBuffer** ComputeBuffer

The buffer to copy to.



**nativeBufferPtr** nint

The memory to copy from.

**nativeBufferSize** int

The number of bytes to copy.

## OnDestroy()

```
protected void OnDestroy()
```

## OnFrameReady(nint, nint, nint, int, int, int, int, int, int)

Callback for [CameraFrameForwarder](#).

```
protected virtual void OnFrameReady(nint yBuffer, nint uBuffer, nint vBuffer, int ySize, int uSize, int vSize, int yRowStride, int uvRowStride, int uvPixelStride)
```

### Parameters

**yBuffer** nint

Pointer to the buffer containing Y (luminance) data of the frame.

**uBuffer** nint

Pointer to the buffer containing U (color) data of the frame.

**vBuffer** nint

Pointer to the buffer containing V (color) data of the frame.

**ySize** int

The size of **yBuffer**.

**uSize** int

The size of **uBuffer**.

**vSize** int

The size of **vBuffer**.

**yRowStride** int

The size of each row of the image in **yBuffer** in bytes.

**uvRowStride** int

The size of each row of the image in **uBuffer** and **vBuffer** in bytes.

**uvPixelStride** int

The size of a pixel in a row of the image in **uBuffer** and **vBuffer** in bytes.

## Release()

Releases the ComputeBuffers and RenderTextures associated with this converter.

```
public void Release()
```

## SendFrameToComputeBuffer(int, int, int)

Sends the camera frame stored in the compute buffers to the compute shader and dispatches it.

```
protected virtual void SendFrameToComputeBuffer(int yRowStride, int uvRowStride,  
int uvPixelStride)
```

### Parameters

**yRowStride** int

The size of each row of the image in [\\_yComputeBuffer](#) in bytes.

**uvRowStride** int

The size of each row of the image in [\\_uComputeBuffer](#) and [\\_vComputeBuffer](#) in bytes.

**uvPixelStride** int

The size of a pixel in a row of the image in [\\_uComputeBuffer](#) and [\\_vComputeBuffer](#) in bytes.

## SetupCameraFrameForwarder(CameraFrameForwarder, Resolution)

Sets the camera frame forwarder.

```
public virtual void SetupCameraFrameForwarder(CameraFrameForwarder cameraFrameForwarder,  
Resolution textureResolution)
```

### Parameters

**cameraFrameForwarder** [CameraFrameForwarder](#)

**textureResolution** Resolution