

# Cross Platform Screen Recorder Kit

[Cross Platform Screen Recorder Kit](#) : Easy Gif/Video Recording on iOS & Android allows to easily screen record your games on Unity. It supports both iOS and Android platforms with unified and simple API.

## Plugin records what ever is visible on screen

Plugin supports mobile platforms Android(Demo) (API 21 and above), iOS(11 and above - iPhone 5S or later, iPad mini 2 or later, iPod Touch 6th generation, iPad Air or later.)

## Highlights

- Unified API design.
- Auto-generates Android Manifest file.
- Easy API for Recording Gif/Video
- Full source code is included.

## Feature set

- Screen Record complete screen with microphone(includes UI)
- Preview Recorded Video
- Single API for both iOS and Android Platforms
- Get recorded file path
- Save recorded video to gallery
- Share recorded video (Share for Gif coming soon)

## Important links

[Product](#) | [Tutorial](#) | [Support](#)

# Namespace **VoxelBusters.ScreenRecorderKit**

## Classes

### **GifRecorderSettings**

### **ScreenRecorderBuilder**

Builder for creating an instance of IScreenRecorder for Video or Gif.

### **ScreenRecorderErrorCode**

### **VideoRecorderRuntimeSettings**

Settings for configuring a video recorder.

## Interfaces

### **IScreenRecorder**

Interface for holding an instance of VideoRecorder or GidRecorder used for recording.

# Class GifRecorderSettings

## Inheritance

System.Object

GifRecorderSettings

Namespace: [VoxelBusters.ScreenRecorderKit](#)

Assembly: cs.temp.dll.dll

## Syntax

```
[Serializable]
public class GifRecorderSettings : SettingsPropertyGroup
```

## Constructors

GifRecorderSettings( Boolean, Int32, Int32, Boolean, Int32, Int32, Int32, Single, ThreadPriority )

## Declaration

```
public GifRecorderSettings( bool isEnabled = true, int width = 320, int height = 200, bool autoAspect = true, int fps = 15, int repeat = 0, int quality = 15, float bufferSize = 3F, ThreadPriority threadPriority = null )
```

## Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	isEnabled	
System.Int32	width	
System.Int32	height	
System.Boolean	autoAspect	
System.Int32	fps	
System.Int32	repeat	
System.Int32	quality	
System.Single	bufferSize	
ThreadPriority	threadPriority	

## Properties

### AutoAspect

Enabling this will adjust the aspect automatically.

## Declaration

```
public bool AutoAspect { get; }
```

## Property Value

TYPE	DESCRIPTION
System.Boolean	

**BufferSize**

Size of the buffer used.

**Declaration**

```
public float BufferSize { get; }
```

**Property Value**

TYPE	DESCRIPTION
System.Single	

**FramePerSecond**

Number of frames to capture per second

**Declaration**

```
public int FramePerSecond { get; }
```

**Property Value**

TYPE	DESCRIPTION
System.Int32	

**Height**

Height of the recording.

**Declaration**

```
public int Height { get; }
```

**Property Value**

TYPE	DESCRIPTION
System.Int32	

**MaxFrameCount**

Max frames allowed per recording.

**Declaration**

```
public int MaxFrameCount { get; }
```

**Property Value**

TYPE	DESCRIPTION
System.Int32	

Quality

Quality of the recording.

Declaration

```
public int Quality { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Repeat

Enable to loop the gif

Declaration

```
public int Repeat { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

ThreadPriority

Priority of the recording thread.

Declaration

```
public ThreadPriority ThreadPriority { get; }
```

Property Value

TYPE	DESCRIPTION
ThreadPriority	

TimePerFrame

Time per each frame.

Declaration

```
public float TimePerFrame { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

Width

Width of the recording.

Declaration

```
public int Width { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

# Interface IScreenRecorder

Interface for holding an instance of VideoRecorder or GidRecorder used for recording.

Namespace: [VoxelBusters.ScreenRecorderKit](#)

Assembly: cs.temp.dll.dll

## Syntax

```
public interface IScreenRecorder
```

## Methods

### CanRecord()

Check if recording is allowed or not.

#### Declaration

```
bool CanRecord()
```

#### Returns

TYPE	DESCRIPTION
System.Boolean	Returns false if recording api is not available or if resources are not available for recording.

### DiscardRecording(CompletionCallback)

Discard an active recording. This won't trigger SetOnRecordingAvailable callback as the current recording is disposed.

#### Declaration

```
void DiscardRecording(CompletionCallback callback = null)
```

#### Parameters

TYPE	NAME	DESCRIPTION
CompletionCallback	callback	Callback to get triggered once discard action is complete.

### Flush()

Flush any resources created for recording.

#### Declaration

```
void Flush()
```

### IsPausedOrRecording()

Check if an active recording exists

#### Declaration

```
bool IsPausedOrRecording()
```

#### Returns

TYPE	DESCRIPTION
System.Boolean	Returns true if a recording is active or recording is paused. False if no active recording exists.

**IsRecording()**

Check if recording is happening or not.

**Declaration**

```
bool IsRecording()
```

**Returns**

TYPE	DESCRIPTION
System.Boolean	Returns true if recording is active and not paused. False if no active recording exists or a recording is paused

**OpenRecording(CompletionCallback)**

Open an active recording which just got recorded.

**Declaration**

```
void OpenRecording(CompletionCallback callback = null)
```

**Parameters**

TYPE	NAME	DESCRIPTION
CompletionCallback	callback	Callback to get triggered once the action is complete.

**PauseRecording(CompletionCallback)**

Pause an active recording

**Declaration**

```
void PauseRecording(CompletionCallback callback = null)
```

**Parameters**

TYPE	NAME	DESCRIPTION
CompletionCallback	callback	Callback to get triggered once after pause action is complete.

**PrepareRecording(CompletionCallback)**

Prepare for recording

**Declaration**

```
void PrepareRecording(CompletionCallback callback = null)
```

**Parameters**



TYPE	NAME	DESCRIPTION
CompletionCallback	callback	Callback to get triggered once prepare action is complete

### SaveRecording(CompletionCallback<ScreenRecorderSaveRecordingResult>)

Save an active recording

#### Declaration

```
void SaveRecording(CompletionCallback<ScreenRecorderSaveRecordingResult> callback = null)
```

#### Parameters

TYPE	NAME	DESCRIPTION
CompletionCallback<VoxelBusters.ScreenRecorderKit.ScreenRecorderSaveRecordingResult>	callback	Callback to get triggered once save action is complete. This returns a result where you can fetch path.

### SaveRecording(String, CompletionCallback<ScreenRecorderSaveRecordingResult>)

Save an active recording with a provided filename.

#### Declaration

```
void SaveRecording(string fileName, CompletionCallback<ScreenRecorderSaveRecordingResult> callback = null)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	fileName	Filename to be set on saving a recording.
CompletionCallback<VoxelBusters.ScreenRecorderKit.ScreenRecorderSaveRecordingResult>	callback	Callback to get triggered once save action is complete. This returns a result where you can fetch path.

### SetOnRecordingAvailable(SuccessCallback<ScreenRecorderRecordingAvailableResult>)

Set a callback to get triggered when a recording result is available. This will be GifTexture in-case of gif recording or video file path in-case of video recording.

#### Declaration

```
void SetOnRecordingAvailable(SuccessCallback<ScreenRecorderRecordingAvailableResult> callback = null)
```

#### Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
SuccessCallback<VoxelBusters.ScreenRecorderKit.ScreenRecorderRecordingAvailableResult>	callback	Callback to get triggered once a recording is available.

**ShareRecording(String, String, CompletionCallback)**

Share an active recording which just got recorded.

**Declaration**

```
void ShareRecording(string text = null, string subject = null, CompletionCallback callback = null)
```

**Parameters**

TYPE	NAME	DESCRIPTION
System.String	text	Share text used for sharing.
System.String	subject	Subject text that needs to be used as subject.
CompletionCallback	callback	Callback that gets triggered once share action is complete.

**StartRecording(CompletionCallback)**

Start a recording

**Declaration**

```
void StartRecording(CompletionCallback callback = null)
```

**Parameters**

TYPE	NAME	DESCRIPTION
CompletionCallback	callback	Callback to get triggered once start action is complete

**StopRecording(CompletionCallback)**

Stop an active recording

**Declaration**

```
void StopRecording(CompletionCallback callback = null)
```

**Parameters**

TYPE	NAME	DESCRIPTION
CompletionCallback	callback	Callback to get triggered once after stop action is complete.

**StopRecording(Boolean, CompletionCallback)**

Stop an active recording

Declaration

```
void StopRecording(bool flushMemory, CompletionCallback callback = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	flushMemory	If true, clears all the memory allocated for this recording.
CompletionCallback	callback	Callback to get triggered once after stop action is complete.

# Class ScreenRecorderBuilder

Builder for creating an instance of IScreenRecorder for Video or Gif.

Inheritance

System.Object

ScreenRecorderBuilder

Namespace: [VoxelBusters.ScreenRecorderKit](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ScreenRecorderBuilder
```

Methods

Build()

Declaration

```
public IScreenRecorder Build()
```

Returns

TYPE	DESCRIPTION
<a href="#">IScreenRecorder</a>	

CreateGifRecorder(Camera, GifRecorderSettings)

Creates an instance of GIF recorder.

Declaration

```
public static ScreenRecorderBuilder CreateGifRecorder(Camera camera = null, GifRecorderSettings settings = null)
```

Parameters

TYPE	NAME	DESCRIPTION
Camera	camera	Camera which needs to be recorded. If null, whole screen will be recorded.
<a href="#">GifRecorderSettings</a>	settings	Settings for configuring this gif recorder.

Returns

TYPE	DESCRIPTION
<a href="#">ScreenRecorderBuilder</a>	

CreateVideoRecorder(VideoRecorderRuntimeSettings)

Creates an instance of the video recorder.

Declaration

```
public static ScreenRecorderBuilder CreateVideoRecorder(VideoRecorderRuntimeSettings runtimeSettings = null)
```

Parameters

TYPE	NAME	DESCRIPTION
<a href="#">VideoRecorderRuntimeSettings</a>	runtimeSettings	Runtime configurable properties.

Returns

TYPE	DESCRIPTION
<a href="#">ScreenRecorderBuilder</a>	

CreateVideoRecorder([VideoRecorderRuntimeSettingsOption](#)[])

Creates an instance of the video recorder.

Declaration

```
public static ScreenRecorderBuilder CreateVideoRecorder(params VideoRecorderRuntimeSettingsOption[] runtimeSettingsOptions)
```

Parameters

TYPE	NAME	DESCRIPTION
VoxelBusters.ScreenRecorderKit.VideoRecorderRuntimeSettingsOption[]	runtimeSettingsOptions	Runtime configurable properties.

Returns

TYPE	DESCRIPTION
<a href="#">ScreenRecorderBuilder</a>	

CreateVideoRecorder([VideoRecorderSettings](#), [VideoRecorderRuntimeSettings](#))

Creates an instance of the video recorder.

Declaration

```
public static ScreenRecorderBuilder CreateVideoRecorder(VideoRecorderSettings settings = null, VideoRecorderRuntimeSettings runtimeSettings = null)
```

Parameters

TYPE	NAME	DESCRIPTION
VoxelBusters.ScreenRecorderKit.VideoRecorderSettings	settings	Basic settings properties.
<a href="#">VideoRecorderRuntimeSettings</a>	runtimeSettings	Runtime configurable properties.

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
<a href="#">ScreenRecorderBuilder</a>	

**CreateVideoRecorder(VideoRecorderSettings, VideoRecorderRuntimeSettingsOption[])**

Creates an instance of the video recorder.

**Declaration**

```
public static ScreenRecorderBuilder CreateVideoRecorder(VideoRecorderSettings settings = null, params
VideoRecorderRuntimeSettingsOption[] runtimeSettingsOptions)
```

**Parameters**

TYPE	NAME	DESCRIPTION
VoxelBusters.ScreenRecorderKit.VideoRecorderSettings	settings	Basic settings properties.
VoxelBusters.ScreenRecorderKit.VideoRecorderRuntimeSettingsOption[]	runtimeSettingsOptions	Runtime configurable properties.

**Returns**

TYPE	DESCRIPTION
<a href="#">ScreenRecorderBuilder</a>	

**SetOnRecordingAvailable(SuccessCallback<ScreenRecorderRecordingAvailableResult>)**

**Declaration**

```
public ScreenRecorderBuilder SetOnRecordingAvailable(SuccessCallback<ScreenRecorderRecordingAvailableResult>
callback = null)
```

**Parameters**

TYPE	NAME	DESCRIPTION
SuccessCallback<VoxelBusters.ScreenRecorderKit.ScreenRecorderRecordingAvailableResult>	callback	

**Returns**

TYPE	DESCRIPTION
<a href="#">ScreenRecorderBuilder</a>	

# Class ScreenRecorderErrorCode

## Inheritance

System.Object

ScreenRecorderErrorCode

Namespace: [VoxelBusters.ScreenRecorderKit](#)

Assembly: cs.temp.dll.dll

## Syntax

```
public static class ScreenRecorderErrorCode
```

## Fields

### kActiveRecordingUnavailable

Error code indicating that there is no active recording to operate on.

#### Declaration

```
public const int kActiveRecordingUnavailable = 4
```

#### Field Value

TYPE	DESCRIPTION
System.Int32	

### kApiUnavailable

Error code indicating that api is unavailable.

#### Declaration

```
public const int kApiUnavailable = 1
```

#### Field Value

TYPE	DESCRIPTION
System.Int32	

### kFeatureUnsupported

Error code indicating that feature is not supported.

#### Declaration

```
public const int kFeatureUnsupported = 5
```

#### Field Value

TYPE	DESCRIPTION
System.Int32	

### kPermissionUnavailable

Error code indicating that required permission is unavailable.

#### Declaration

```
public const int kPermissionUnavailable = 3
```

#### Field Value

TYPE	DESCRIPTION
System.Int32	

### kRecordingInProgress

Error code indicating that recorder is currently busy recording as recording is in progress.

#### Declaration

```
public const int kRecordingInProgress = 2
```

#### Field Value

TYPE	DESCRIPTION
System.Int32	

### kShareServiceUnavailable

Error code indicating that there is no service available for sharing.

#### Declaration

```
public const int kShareServiceUnavailable = 6
```

#### Field Value

TYPE	DESCRIPTION
System.Int32	

### kUnknown

Error code indicating that an unknown or unexpected error occurred.

#### Declaration

```
public const int kUnknown = 0
```

#### Field Value

TYPE	DESCRIPTION
System.Int32	



# Class VideoRecorderRuntimeSettings

Settings for configuring a video recorder.

### Inheritance

System.Object

VideoRecorderRuntimeSettings

Namespace: [VoxelBusters.ScreenRecorderKit](#)

Assembly: cs.temp.dll.dll

### Syntax

```
public class VideoRecorderRuntimeSettings
```

### Constructors

#### VideoRecorderRuntimeSettings(Nullable<Boolean>)

Create an instance of VideoRecorderRuntimeSettings by passing microphone status.

### Declaration

```
public VideoRecorderRuntimeSettings(bool? enableMicrophone = default(bool? ))
```

### Parameters

TYPE	NAME	DESCRIPTION
System.Nullable<System.Boolean>	enableMicrophone	

### Properties

#### EnableMicrophone

Set this to true if you want to use microphone.

### Declaration

```
public bool? EnableMicrophone { get; }
```

### Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	

### Operators

#### Implicit(VideoRecorderRuntimeSettingsOption[] to VideoRecorderRuntimeSettings)

### Declaration

```
public static implicit operator VideoRecorderRuntimeSettings(VideoRecorderRuntimeSettingsOption[] options)
```

### Parameters

TYPE	NAME	DESCRIPTION
VoxelBusters.ScreenRecorderKit.VideoRecorderRuntimeSettingsOption[]	options	

### Returns

TYPE	DESCRIPTION
<a href="#">VideoRecorderRuntimeSettings</a>	