## **UltimateReplay Namespace**

## ▲ Classes

	Class	Description
<b>*</b> \$	ReplayAudio	A replay component that is responsible for recording and replaying audio effects that are played during gameplay. Audio can only be replayed when playback is not being reversed.
***	ReplayBehaviour	This interface can be implemented by mono behaviour scripts in order to receive replay start and end events. It works in a similar way to the 'Start' or 'Update' method however you must explicitly implement the interface as opposed to using magic methods. This allows for slightly improved performance.
<b>9</b> 3	ReplayControls	Default replay controls used for demonstration and testing. Uses legacy GUI for ui rendering.
<b>9</b> Ç	ReplayIgnoreAttribute	Attach this attribute to a class that derives from ReplayBehaviour and the replay system will ignore it. This is

		useful when you want to receive replay events but dont need to record any data.
<b>~</b> \$	ReplayManager	The main interface for Ultimate Replay and allows full control over object recording and playback.
<b>*</b> \$	ReplayObject	Only one instance of ReplayObject can be added to any game object.
<b>~</b> \$	ReplayState	A ReplayState allows replay objects to serialize and deserialize their data. See IReplaySerialize.
<b>*</b> \$	ReplayTime	This class emulates the behaviour of the Time class in Unity and can be used to modify the playback speed of a replay. There are also delta values that can be used to interpolate between frames where a low record frame rate is used. See ReplayTransform for an example.
43	ReplayTransform	Attach this component to a game objects in order to record the objects transform for replays. Only one instance of ReplayTransform can be added to any game object.
<b>~</b> \$	ReplayVarAttribute	Use this attribute on a field to mark it for recording. The type

the field is defined in must inheit from ReplayBehaviour in order for the field to be recorded automatically. Interpolation between field values is also possible where low record rates are used.

### ▲ Enumerations

	Enumeration	Description
19 <sup>10</sup>	PlaybackDirection	The playback direction used during replay plaback.
8	PlaybackOrigin	Represents a playback node that can be used to calcualte playback offsets.
3 <sup>90</sup>	PlaybackState	The state of the active ReplayManager.

# **PlaybackDirection Enumeration**

The playback direction used during replay plaback.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

public enum PlaybackDirection

## Members

Member name	Value	Description
Forward	0	The replay should be played back in normal mode.
Backward	1	The replay should be played back in reverse mode.

## J See Also

Reference UltimateReplay Namespace

# **PlaybackOrigin Enumeration**

Represents a playback node that can be used to calcualte playback offsets.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

public enum PlaybackOrigin

## Members

Member name	Value	Description
Start	0	The start of the playback sequence.
Current	1	The current frame in the playback sequence.
End	2	The end of the playback sequence.

## ⊿ See Also

Reference UltimateReplay Namespace

# PlaybackState Enumeration

The state of the active ReplayManager.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

public enum PlaybackState

## Members

Member name	Value	Description
Idle	0	The manager is doing nothing.
Recording	1	The manager is currently recording the scene using the current record settings.
Recording_Paused	2	The manager is currently paused but is expecting to resume recording.
Playback	3	The manager is performing playback using the current settings.

Playback\_Paused 4

The manager is currently paused but is expecting to resume playback.

### ⊿ See Also

Reference UltimateReplay Namespace

# **ReplayAudio Class**

A replay component that is responsible for recording and replaying audio effects that are played during gameplay. Audio can only be replayed when playback is not being reversed.

## Inheritance Hierarchy

SystemObject Object Component Behaviour MonoBehaviour UltimateReplayReplayBehaviour UltimateReplayReplayAudio

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

C#

Сору \_\_

public class ReplayAudio : ReplayBehaviour

The ReplayAudio type exposes the following members.

### Constructors

		Name	Description
ReplayAudio Initializes a new instance of the ReplayAudio class	<b>≓</b> ©	ReplayAudio	

Тор

## ▲ Methods

	Name	Description
÷	Awake	Called by Unity. (Overrides ReplayBehaviourAwake.)
<b>≞</b>	OnReplayEvent	Called by the replay system when a replay $\epsilon$ has occured. (Overrides ReplayBehaviourOnReplayEvent(ReplayEvent)
≓∳	Play	You should call this method as a replacement AudioSource.Play as it will also record the ti of the audio event so that it can be replayed later.

#### Тор

## ⊿ Fields

	Name	Description
٥	observedAudioSource	The audio source that will be recorded by this ReplayAudio.
Тор		

## ⊿ See Also

Reference UltimateReplay Namespace

# **ReplayAudio Constructor**

Initializes a new instance of the ReplayAudio class

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

**C**#

public ReplayAudio()

### ⊿ See Also

# ReplayAudio Fields

The ReplayAudio type exposes the following members.

## ▲ Fields

	Name	Description		
۵	observedAudioSource	The audio source that will be recorded by this ReplayAudio.		
Тор				
⊿ See Also				
Reference ReplayAudi				

UltimateReplay Namespace

# ReplayAudioobservedAudioSource Field

The audio source that will be recorded by this ReplayAudio.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

Сору

public AudioSource observedAudioSource

Field Value Type: **AudioSource** 

### ⊿ See Also

# **ReplayAudio Methods**

The ReplayAudio type exposes the following members.

## Methods

	Name	Description
<b>≡</b>	Awake	Called by Unity. (Overrides ReplayBehaviourAwake.)
= <b>0</b>	OnReplayEvent	Called by the replay system when a replay e has occured. (Overrides ReplayBehaviourOnReplayEvent(ReplayEvent)
=0	Play	You should call this method as a replacement AudioSource.Play as it will also record the ti of the audio event so that it can be replayed later.

Тор

⊿ See Also

# ReplayAudioAwake Method

Called by Unity.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

**C**#

public override void Awake()

### ⊿ See Also

# ReplayAudioOnReplayEvent Method

Called by the replay system when a replay event has occured.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

Сору

#### )

#### Parameters

replayEvent Type: UltimateReplay.CoreReplayEvent The ReplayEvent that was triggered

#### ⊿ See Also

# ReplayAudioPlay Method

You should call this method as a replacement for AudioSource.Play as it will also record the time of the audio event so that it can be replayed later.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Сору

public void Play()

#### ⊿ See Also

# ReplayBehaviour Class

This interface can be implemented by mono behaviour scripts in order to receive replay start and end events. It works in a similar way to the 'Start' or 'Update' method however you must explicitly implement the interface as opposed to using magic methods. This allows for slightly improved performance.

### ▲ Inheritance Hierarchy

#### SystemObject Object

t

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax



The ReplayBehaviour type exposes the following members.

## Constructors

	Name	Description
ġŴ	ReplayBehaviour	Initializes a new instance of the ReplayBehaviour class
Тор		
⊿ Me	ethods	
	Name	Description
≓ <b>∲</b>	Awake	Called by Unity. Allows to ReplayBehaviour to vali its identity and register in with the replay system.
= <b>0</b>	OnReplayDeserialize	Called by the replay sys when all replay state information should be deserialized.
<b>≡</b>	OnReplayEnd	Called by the replay sys when playback has end You can re-enable game behaviour in this method allow the gameplay to 'ta over'
=♥	OnReplayEvent	Called by the replay sys when an event has beer received during playbac
= <b>0</b>	OnReplayPlayPause	Called by the replay sys when playback is about paused or resumed.
≡ <b>≬</b>	OnReplaySerialize	Called by the replay sys when all replay state

		information should be serialized.
. <b>≕</b>	OnReplayStart	Called by the replay sys when playback is about start. You can disable ga behaviour that should nc during playback in this method, such as player movement.
<b>≡</b> ©	OnReplayUpdate	Called by the replay system every frame while playba active.
ş û	ReplayDeserializeEvents	Deserializes all active ReplayEvent from the st and dispatches any even the OnReplayEvent(ReplayI handler.
ĕ <b>₽</b>	ReplayDeserializeVariables	Deserializes all active ReplayVariable from the state.
ĕ <b>₽</b>	ReplayFindVariables	Attempts to find any vari marked with the ReplayVarAttribute so th they can be serialized la
ĕ <b>₽</b>	ReplayInterpolateVariables	Allows all active ReplayVariable to be interpolated betwen repl frames.
≡Q	ReplayRecordEvent(Byte, ReplayState)	Push an event to the repart of

≓Ŵ	ReplayRecordEvent(ReplayEvents, ReplayState)	Push an event to the rep system for recording.
<u>ş</u>	ReplaySerializeEvents	Serializes all awaiting ReplayEvent to the state
<u>ş</u>	ReplaySerializeVariables	Serializes all active ReplayVariable to the sta
<b>≓</b> ©	Reset	Called by Unity while in mode. Allows the unique be generated when the s is attached to an object.

#### Тор

# ▲ Properties

	Name	Description
<b>*</b>	Identity	Get the ReplayIdentity associated with this ReplayBehaviour.
<b>**</b>	IsRecording	Returns true if the active replay manager is currently recording the scene. Note: If recording is paused this value will still be true.
<b>*</b>	IsReplaying	Returns true if the active replay manager is currently replaying a previous recording. Note: If playback is paused this value will still be true.
	PlaybackDirection	Gets the current PlaybackDirection of replay playback.

Тор

## ⊿ See Also

Reference UltimateReplay Namespace

# **ReplayBehaviour Constructor**

Initializes a new instance of the ReplayBehaviour class

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

protected ReplayBehaviour()

### ⊿ See Also

## **ReplayBehaviour Methods**

The ReplayBehaviour type exposes the following members.

### Methods

	Name	Description
<b>≞</b> ∲	Awake	Called by Unity. Allows t ReplayBehaviour to valid its identity and register it with the replay system.
. <b>≕</b> ∲	OnReplayDeserialize	Called by the replay sys when all replay state information should be deserialized.
≓∳	OnReplayEnd	Called by the replay sys when playback has ende You can re-enable game behaviour in this methoc allow the gameplay to 'ta over'
= <b>\$</b>	OnReplayEvent	Called by the replay sys when an event has beer received during playbac
<b>≡\$</b>	OnReplayPlayPause	Called by the replay sys when playback is about paused or resumed.
=•	OnReplaySerialize	Called by the replay sys when all replay state

		information should be serialized.
. <b>≕</b>	OnReplayStart	Called by the replay sys when playback is about start. You can disable ga behaviour that should no during playback in this method, such as player movement.
≡Ŷ	OnReplayUpdate	Called by the replay system every frame while playba active.
<u>ş</u>	ReplayDeserializeEvents	Deserializes all active ReplayEvent from the st and dispatches any even the OnReplayEvent(ReplayI handler.
ĕ <b>₽</b>	ReplayDeserializeVariables	Deserializes all active ReplayVariable from the state.
<u>ş</u>	ReplayFindVariables	Attempts to find any vari marked with the ReplayVarAttribute so th they can be serialized la
ġ <b>₽</b>	ReplayInterpolateVariables	Allows all active ReplayVariable to be interpolated betwen repl frames.
≡Q	ReplayRecordEvent(Byte, ReplayState)	Push an event to the repart of

-= <b>Q</b>	ReplayRecordEvent(ReplayEvents, ReplayState)	Push an event to the rep system for recording.
<sup>™</sup>	ReplaySerializeEvents	Serializes all awaiting ReplayEvent to the state
<u>ş</u>	ReplaySerializeVariables	Serializes all active ReplayVariable to the sta
<b>≡©</b>	Reset	Called by Unity while in mode. Allows the unique be generated when the s is attached to an object.

#### Тор

## ⊿ See Also

# ReplayBehaviourAwake Method

Called by Unity. Allows the ReplayBehaviour to validate its identity and register its self with the replay system.

Copy

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

public virtual void Awake()

#### ⊿ See Also

# ReplayBehaviourOnReplayDeseriali Method

Called by the replay system when all replay state information should be deserialized.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

)

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to read the data from

Implements IReplaySerializeOnReplayDeserialize(ReplayState)

### ⊿ See Also

# ReplayBehaviourOnReplayEnd Method

Called by the replay system when playback has ended. You can reenable game behaviour in this method to allow the gameplay to 'take over'

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Сору

public virtual void OnReplayEnd()

## ⊿ See Also

# ReplayBehaviourOnReplayEvent Method

Called by the replay system when an event has been received during playback.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

)

#### Parameters

replayEvent

Type: UltimateReplay.CoreReplayEvent The event that was received

#### ▲ See Also

# ReplayBehaviourOnReplayPlayPaus Method

Called by the replay system when playback is about to be paused or resumed.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Copy

)

#### Parameters

paused

Type: SystemBoolean

True if playback is about to be paused or false if plyabck is about to be resumed

### ⊿ See Also

# ReplayBehaviourOnReplaySerialize Method

Called by the replay system when all replay state information should be serialized.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

)

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to write the data to

Implements IReplaySerializeOnReplaySerialize(ReplayState)

### J See Also

# ReplayBehaviourOnReplayStart Method

Called by the replay system when playback is about to start. You can disable game behaviour that should not run during playback in this method, such as player movement.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Сору

public virtual void OnReplayStart()

## ⊿ See Also

# ReplayBehaviourOnReplayUpdate Method

Called by the replay system every frame while playback is active.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

Сору

public virtual void OnReplayUpdate()

### ⊿ See Also

# ReplayBehaviourReplayDeserializeE Method

Deserializes all active ReplayEvent from the state and dispatches any events to the OnReplayEvent(ReplayEvent) handler.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Copy

)

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to read the variable data from

### ⊿ See Also

# ReplayBehaviourReplayDeserialize\ Method

Deserializes all active ReplayVariable from the state.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

#### C#

Copy

 $\mathbf{F}$ 

#### ) •]

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to read the variable data from

#### ⊿ See Also

# ReplayBehaviourReplayFindVariable Method

Attempts to find any variables marked with the ReplayVarAttribute so that they can be serialized later.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

Сору \_

protected virtual void ReplayFindVariables()

#### ⊿ See Also

## ReplayBehaviourReplayInterpolateV Method

Allows all active ReplayVariable to be interpolated betwen replay frames.

Namespace: UltimateReplay

Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

```
protected virtual void ReplayInterpolateVariables(
        float delta
)
4
```

Copy

#### **Parameters**

delta

Type: SystemSingle

The 't' value between frames used for interpolation

### ▲ See Also

## ReplayBehaviourReplayRecordEver Method

## Overload List

	Name	Description
=∳	ReplayRecordEvent(Byte, ReplayState)	Push an event to the replay system for recording.
=♥	ReplayRecordEvent(ReplayEvents, ReplayState)	Push an event to the replay system for recording.

#### Тор

## ⊿ See Also

# ReplayBehaviourReplayRecordEver Method (Byte, ReplayState)

Push an event to the replay system for recording.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

## ▲ Syntax

**C**#

)

#### Parameters

eventID

Type: SystemByte

The event id to uniquley identify the event type

state (Optional)

Type: UltimateReplayReplayState The state data for the event or null if no state data is required

## ⊿ See Also

Reference ReplayBehaviour Class ReplayRecordEvent Overload UltimateReplay Namespace

# ReplayBehaviourReplayRecordEver Method (ReplayEvents, ReplayState

Push an event to the replay system for recording.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

## ▲ Syntax

**C**#

)

#### Parameters

eventID

Type: UltimateReplay.CoreReplayEvents The event id to uniquley identify the event type

state (Optional)

Type: UltimateReplayReplayState

The state data for the event or null if no state data is required

## ⊿ See Also

Reference ReplayBehaviour Class ReplayRecordEvent Overload UltimateReplay Namespace

## ReplayBehaviourReplaySerializeEve Method

Serializes all awaiting ReplayEvent to the state.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

Сору

)

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to write the varaible data to

### ⊿ See Also

## ReplayBehaviourReplaySerializeVar Method

Serializes all active ReplayVariable to the state.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

Copy

)

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to write the varaible data to

### ⊿ See Also

# ReplayBehaviourReset Method

Called by Unity while in editor mode. Allows the unique id to be generated when the script is attached to an object.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

## ▲ Syntax

C#

public virtual void Reset()

## ⊿ See Also

## **ReplayBehaviour Properties**

The ReplayBehaviour type exposes the following members.

## ▲ Properties

	Name	Description
<b>**</b>	Identity	Get the ReplayIdentity associated with this ReplayBehaviour.
<b>**</b>	IsRecording	Returns true if the active replay manager is currently recording the scene. Note: If recording is paused this value will still be true.
<b>*</b>	IsReplaying	Returns true if the active replay manager is currently replaying a previous recording. Note: If playback is paused this value will still be true.
<b>*</b>	PlaybackDirection	Gets the current PlaybackDirection of replay playback.

Тор

## ⊿ See Also

# ReplayBehaviourIdentity Property

Get the ReplayIdentity associated with this ReplayBehaviour.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

C#

Сору

public ReplayIdentity Identity { get; }

Property Value Type: ReplayIdentity

## ⊿ See Also

# ReplayBehaviourIsRecording Property

Returns true if the active replay manager is currently recording the scene. Note: If recording is paused this value will still be true.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

```
public bool IsRecording { get; }
```

Property Value Type: Boolean

### ⊿ See Also

# ReplayBehaviourIsReplaying Property

Returns true if the active replay manager is currently replaying a previous recording. Note: If playback is paused this value will still be true.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)



**C**#

\_\_\_\_\_

Сору

public bool IsReplaying { get; }

Property Value Type: Boolean

### ⊿ See Also

# **ReplayBehaviourPlaybackDirection** Property

Gets the current PlaybackDirection of replay playback.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

public PlaybackDirection PlaybackDirection { get; [▲]

Copy

**Property Value Type: PlaybackDirection** 

## ▲ See Also

# ReplayControls Class

Default replay controls used for demonstration and testing. Uses legacy GUI for ui rendering.

## ▲ Inheritance Hierarchy

SystemObject Object Component Behaviour MonoBehaviour UltimateReplayReplayControls

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

public class ReplayControls : MonoBehaviour

The ReplayControls type exposes the following members.

## Constructors

	Name	Description
<b>≓©</b>	ReplayControls	Initializes a new instance of the ReplayControls class
Tara		

Тор

## Methods

	Name	Description
≡\$	Awake	Called by Unity.
≡ <b>`</b>	OnGUI	Called by unity.
<b>≡∲</b>	Start	Called by Unity.
≡ <b>∲</b>	Update	Called by Unity.

#### Тор

## **▲** Fields

	Name	Description
Ŷ	allowPlaybackFreeCam	Should the free cam mode be enabled during playback.
9	flySpeed	How fast the free cam can move around the scene.
Ŷ	lookSpeed	How fast the free cam can look around the scene.

#### Тор

## ⊿ See Also

Reference UltimateReplay Namespace

## **ReplayControls Constructor**

Initializes a new instance of the ReplayControls class

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

public ReplayControls()

## ⊿ See Also

# ReplayControls Fields

The ReplayControls type exposes the following members.

## ▲ Fields

	Name	Description
Ŷ	allowPlaybackFreeCam	Should the free cam mode be enabled during playback.
٥	flySpeed	How fast the free cam can move around the scene.
Ŷ	lookSpeed	How fast the free cam can look around the scene.

Тор

⊿ See Also

# ReplayControlsallowPlaybackFreeC Field

Should the free cam mode be enabled during playback.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Сору

public bool allowPlaybackFreeCam

Field Value Type: Boolean

### ⊿ See Also

# ReplayControlsflySpeed Field

How fast the free cam can move around the scene.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

C#

Copy \_

public float flySpeed

Field Value Type: Single

### ⊿ See Also

# ReplayControlslookSpeed Field

How fast the free cam can look around the scene.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

public float lookSpeed

Field Value Type: Single

### ⊿ See Also

# **ReplayControls Methods**

The ReplayControls type exposes the following members.

## Methods

	Name	Description
=	Awake	Called by Unity.
= <b>Q</b>	Ongui	Called by unity.
= <b>Q</b>	Start	Called by Unity.
-= <b>Q</b>	Update	Called by Unity.

#### Тор



## ReplayControlsAwake Method

Called by Unity.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

C#

public void Awake()

### ⊿ See Also

## ReplayControlsOnGUI Method

Called by unity.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

public void OnGUI()

### ⊿ See Also

## ReplayControlsStart Method

Called by Unity.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

public void Start()

### ⊿ See Also

# ReplayControlsUpdate Method

Called by Unity.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

public void Update()

### ⊿ See Also

# ReplayIgnoreAttribute Class

Attach this attribute to a class that derives from ReplayBehaviour and the replay system will ignore it. This is useful when you want to receive replay events but dont need to record any data.

## ▲ Inheritance Hierarchy

#### SystemObject SystemAttribute UltimateReplayReplayIgnoreAttribute

Namespace: UltimateReplay

Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax



The ReplayIgnoreAttribute type exposes the following members.

## Constructors

	Name	Description
<b>≓∲</b>	ReplayIgnoreAttribute	Initializes a new instance of the ReplayIgnoreAttribute class
Top		

lob

▲ See Also

#### Reference UltimateReplay Namespace

# ReplayIgnoreAttribute Constructor

Initializes a new instance of the ReplayIgnoreAttribute class

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Copy \_

public ReplayIgnoreAttribute()

## ⊿ See Also

Reference ReplayIgnoreAttribute Class UltimateReplay Namespace

# ReplayManager Class

The main interface for Ultimate Replay and allows full control over object recording and playback.

## ▲ Inheritance Hierarchy

SystemObject Object Component Behaviour MonoBehaviour UltimateReplay.UtilMonoSingletonReplayManager UltimateReplayReplayManager

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax



The ReplayManager type exposes the following members.

## Constructors

	Name	Description
≓Ŵ	ReplayManager	Initializes a new instance of the ReplayManager class

Тор

## ▲ Methods

	Name	Description
<i>≣</i> ₩ S	BeginPlayback	Use this method to begin the playback of the recorded obje SetPlaybackFrame(Single, PlaybackOrigin) or SetPlaybackFrameNormalize PlaybackOrigin) before calling method to specify the exact le at which playback should beg method will run the entire play gathered and then automatics playback on completion if endReplayAfterPlayback is tr
<i>.</i>	BeginPlaybackFrame	Use this method to set the cu playback at a specific replay This will allow the state of a s replay frame to be restored b not continue playback which provide a freeze frame effect. SetPlaybackFrame(Single, PlaybackOrigin) or SetPlaybackFrameNormalize PlaybackOrigin) before calling method to specify the exact le at which the playback frame s be sampled. Use StopPlayba unfreeze the still frame and re normal game mode. This met ignore the value of endReplayAfterPlayback as c single frame is replayed. As a you will need to call StopPlay

		when you want to end the pla frame.
=\$ S	BeginRecording	Use this method to begin san the recorded objects in the sc recordOnStart is true then this method will be called automa when the manager is initialize state information will be recor the assigned ReplayTarget (E ReplayMemoryTarget). When any previous recording data v discarded
=\$ <b>S</b>	DiscardRecording	This method will throw away a recorded data and flush the re- target if necessary. This meth- be called at any time. If the m is currently recording then all data will be discarded and re- will continue. If the manager i currently replaying then all re- will be discarded and playbac stop.
= <b>0 S</b>	FindReplayPrefab	Attempts to find the prefab wi matching name. This is used restore objects that were des during recording.
=\$ <b>S</b>	ForceAwake	Override implementation of ForceAwake. Performs exact same behaviour. Simply inclu that the user does not need to 'UltimateReplay.Util' to access method.
≡Q	OnDestroy	Called by Unity. Allows the ac

		replay manager to cleanup ar recordings.
=♥	OnLevelWasLoaded	Called by Unity. Allows the ac replay manager to cleanup re when a scene change is mad
≓Ŵ	OnValidate	Called by Unity.
= <b>\$</b>	PausePlayback	Use this method to pause rep playback while maintinaing th current replay state. See ResumePlayback to continue playback.
= <b>∲</b>	PauseRecording	Use this method when you way pause recording but may con recording at any point. A good candidate for pausing recordi when the user pauses the ga is shown a pause menu. The manager must already be rec otherwise this method will hav effect.
	RegisterReplayPrefab	Attempts to register a game c a prefab so that the replay sy able to spawn or despawn the as needed. You only need to for objects that are likley to be instantiated or destroyed duri recording. The replay system be able to accuratley restore scene state during playback. specified object must be a pre otherwise an error will be thrc the object will not be registere Prefab instances are not acce

<b>≕∳ S</b>	ReplayDestroy	Attempts to destroy the speci prefab. OnReplayDestroy will called if a listener has been re otherwise default destruction used.
<b>≕∳ S</b>	ReplayInstantiate	Attempts to instantiate the sp prefab. OnReplayInstantiate v called if a listener has been re otherwise default instantiation used.
<b>≕∳ S</b>	ResumePlayback	Use this method to resume pl after a previous call to PausePlayback was called. If PausePlayback was not calle this method then the method no effect.
<b>≈\$</b>	ResumeRecording	Use this method to resume re after a previous call to PauseRecording. The manag already be recording otherwis method will have no effect.
<b>≡0 S</b>	SetPlaybackFrame	Use this method to specify wl the replay sequence the playl should start. If the offset does within the bounds of the repla the value will be clamped to r either the start or end frame.
-: <b>≬ S</b>	SetPlaybackFrameNormalized	Use this method to specify wl the replay sequence the play should start. This method acc normalized offsets values bet and 1 and performs validatior

		using the value.
.≡ <b>∳</b>	Start	Called by Unity. Allows the ac replay manager to initialize.
=\$ S	StopPlayback	Use this method to stop any a playback. This method will or an effect if there is an active p running otherwise it will have effect.
=\$ <b>S</b>	StopRecording	Use this method to stop recon after a previous call to BeginRecording(Boolean). The manager must already be recontherwise this method will have effect. This method must be contherwise the playback otherwise you may get unpre- results.
= <b>Q</b>	Update	Called by Unity. Allows the ac replay manager to update rec playback.
Тор		

## ⊿ Fields

	Name	Description
۶	endReplayAfterPlayback	When true, the replay manager will automatically restore the previous game state after playback has finished. When false, playback will continue to loop forever until a suitable stop

		or pause command is issued manually.
	OnReplayDestroy	Called by the replay system whenever it needs to destroy a game object in order to restore a previous scene state. You can add a listener to override the default behaviour which can be useful if you want to handle the destruction manually for purposes such as object pooling.
Ŷ S	OnReplayInstantiate	Called by the replay system whenever it needs to instantiate a prefab for use during playback. You can add a listener to override the default behaviour which can be useful if you want to handle the instantiation manually for purposes such as object pooling.
	prefabs	A collection of prefabs that may be spawned or destroyed during recording and as a result amy need to be spawned or destroyed in order to accuratley recreate the replay.
٢	recordFPS	The target record framerate of the sampler. The higher this value the higher the memory consumption and

		cpu usage will be. You will need to fine tune this value to tradeoff performance or memory usage for replay accuracy.
۵	recordOnStart	When true, the manager will automatically begin recording the scene one it is initialized.

#### Тор

## ▲ Properties

	Name	Description
<b>≌</b> s	CurrentPlaybackTime	Get the current playback time in seconds. This value will never be greater than duration.
S	CurrentPlaybackTimeNormalized	Get the current playback time as a normalized value between 0-1. 0 represents the starting frame of the recording and 1 represents the very last frame of the recording.
📽 S	IsPaused	Returns true when the active replay manager is in any paused state. Paused states could include

		Playback_Paused or Recording_Paused.
<b>≧</b> S	IsRecording	Returns true if the manager is currently recording the scene. Note: If recording is paused this value will still be true.
in s	IsReplaying	Returns true if the manager is currently playing back previously recorded replay data. Note: if playback is paused this value will still be true.
i≊ s	PlaybackDirection	Gets the current PlaybackDirection of replay playback.
in s	Preparer	Access the current IReplayPreparer that the active replay manager will use to prepare game objects for replay. By default a DefaultReplayPreparer is used.
<b>₽ S</b>	Replay	Get the active replay manager in the scene. If no replay manager could be found then one will be created with default settings.

		This property may be null in a rare case where the active replay manager was destroyed in the same frame as an application quit event was issued. In this situation the replay manager cannot be recreated as it would cause leaked objects.
🖻 S	Scene	Get the ReplayScene associated with the replay system.
in s	Target	The current replay target that is being used to store the replay data. By default, the replay target is ReplayMemoryTarget.

### Тор

## ▲ See Also

Reference UltimateReplay Namespace

# ReplayManager Constructor

Initializes a new instance of the ReplayManager class

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

C#

public ReplayManager()

## ⊿ See Also

# ReplayManager Fields

The ReplayManager type exposes the following members.

## ▲ Fields

	Name	Description
	endReplayAfterPlayback	When true, the replay manager will automatically restore the previous game state after playback has finished. When false, playback will continue to loop forever until a suitable stop or pause command is issued manually.
Ŷ <b>S</b>	OnReplayDestroy	Called by the replay system whenever it needs to destroy a game object in order to restore a previous scene state. You can add a listener to override the default behaviour which can be useful if you want to handle the destruction manually for purposes such as object pooling.
Ŷ S	OnReplayInstantiate	Called by the replay system whenever it needs to instantiate a prefab for use during playback. You can add

		a listener to override the default behaviour which can be useful if you want to handle the instantiation manually for purposes such as object pooling.
۶	prefabs	A collection of prefabs that may be spawned or destroyed during recording and as a result amy need to be spawned or destroyed in order to accuratley recreate the replay.
•	recordFPS	The target record framerate of the sampler. The higher this value the higher the memory consumption and cpu usage will be. You will need to fine tune this value to tradeoff performance or memory usage for replay accuracy.
٢	recordOnStart	When true, the manager will automatically begin recording the scene one it is initialized.

### Тор

## ⊿ See Also

# ReplayManagerendReplayAfterPlay Field

When true, the replay manager will automatically restore the previous game state after playback has finished. When false, playback will continue to loop forever until a suitable stop or pause command is issued manually.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Сору

public bool endReplayAfterPlayback

Field Value Type: Boolean

### ⊿ See Also

# ReplayManagerOnReplayDestroy Field

Called by the replay system whenever it needs to destroy a game object in order to restore a previous scene state. You can add a listener to override the default behaviour which can be useful if you want to handle the destruction manually for purposes such as object pooling.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Сору \_

public static Action<GameObject> OnReplayDestroy

Field Value Type: ActionGameObject

### ⊿ See Also

# ReplayManagerOnReplayInstantiate Field

Called by the replay system whenever it needs to instantiate a prefab for use during playback. You can add a listener to override the default behaviour which can be useful if you want to handle the instantiation manually for purposes such as object pooling.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax



### Field Value

Type: FuncGameObject, Vector3, Quaternion, GameObject

### ⊿ See Also

# ReplayManagerprefabs Field

A collection of prefabs that may be spawned or destroyed during recording and as a result amy need to be spawned or destroyed in order to accuratley recreate the replay.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

Сору

public GameObject[] prefabs

Field Value Type: **GameObject** 

### ⊿ See Also

# ReplayManagerrecordFPS Field

The target record framerate of the sampler. The higher this value the higher the memory consumption and cpu usage will be. You will need to fine tune this value to tradeoff performance or memory usage for replay accuracy.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Сору

public int recordFPS

Field Value Type: Int32

### ⊿ See Also

# ReplayManagerrecordOnStart Field

When true, the manager will automatically begin recording the scene one it is initialized.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Copy \_

public bool recordOnStart

Field Value Type: Boolean

### ⊿ See Also

# ReplayManager Methods

The ReplayManager type exposes the following members.

## Methods

	Name	Description
<b>≣\$</b>	BeginPlayback	Use this method to begin the playback of the recorded obje SetPlaybackFrame(Single, PlaybackOrigin) or SetPlaybackFrameNormalize PlaybackOrigin) before calling method to specify the exact lo at which playback should beg method will run the entire play gathered and then automatica playback on completion if endReplayAfterPlayback is tr
.≕Ŷ S	BeginPlaybackFrame	Use this method to set the cu playback at a specific replay This will allow the state of a s replay frame to be restored b not continue playback which provide a freeze frame effect. SetPlaybackFrame(Single, PlaybackOrigin) or SetPlaybackFrameNormalize PlaybackOrigin) before calling method to specify the exact lo at which the playback frame s be sampled. Use StopPlayba

		unfreeze the still frame and re normal game mode. This met ignore the value of endReplayAfterPlayback as c single frame is replayed. As a you will need to call StopPlay when you want to end the pla frame.
<b>≕∲ S</b>	BeginRecording	Use this method to begin san the recorded objects in the sc recordOnStart is true then thi method will be called automa when the manager is initialize state information will be recor the assigned ReplayTarget (E ReplayMemoryTarget). When any previous recording data v discarded
=\$ <b>S</b>	DiscardRecording	This method will throw away recorded data and flush the re- target if necessary. This meth- be called at any time. If the m- is currently recording then all data will be discarded and re- will continue. If the manager i currently replaying then all re- will be discarded and playbac stop.
=\$ <b>S</b>	FindReplayPrefab	Attempts to find the prefab wi matching name. This is used restore objects that were des during recording.
<b>∉</b> ≬ S	ForceAwake	Override implementation of ForceAwake. Performs exact

		same behaviour. Simply inclu that the user does not need to 'UltimateReplay.Util' to acces method.
= <b>\$</b>	OnDestroy	Called by Unity. Allows the ac replay manager to cleanup ar recordings.
= <b>\$</b>	OnLevelWasLoaded	Called by Unity. Allows the ac replay manager to cleanup re when a scene change is mad
=	OnValidate	Called by Unity.
= <b>\$</b> \$	PausePlayback	Use this method to pause rep playback while maintinaing th current replay state. See ResumePlayback to continue playback.
<b>=∳ S</b>	PauseRecording	Use this method when you wa pause recording but may con recording at any point. A good candidate for pausing recordi when the user pauses the ga is shown a pause menu. The manager must already be rec otherwise this method will hav effect.
<b>≣≬ S</b>	RegisterReplayPrefab	Attempts to register a game c a prefab so that the replay sy able to spawn or despawn the as needed. You only need to for objects that are likley to be instantiated or destroyed duri recording. The replay system

		be able to accuratley restore scene state during playback. specified object must be a pro otherwise an error will be thro the object will not be register Prefab instances are not acce
=\$ <b>S</b>	ReplayDestroy	Attempts to destroy the speci prefab. OnReplayDestroy will called if a listener has been re otherwise default destruction used.
= <b>0</b> S	ReplayInstantiate	Attempts to instantiate the sp prefab. OnReplayInstantiate called if a listener has been re otherwise default instantiatior used.
<b>≕∳ S</b>	ResumePlayback	Use this method to resume pl after a previous call to PausePlayback was called. If PausePlayback was not calle this method then the method no effect.
=\$ S	ResumeRecording	Use this method to resume reafter a previous call to PauseRecording. The manag already be recording otherwis method will have no effect.
=\$ S	SetPlaybackFrame	Use this method to specify we the replay sequence the play should start. If the offset does within the bounds of the repla the value will be clamped to r either the start or end frame.

= <b>\$</b>		
	SetPlaybackFrameNormalized	Use this method to specify wl the replay sequence the play should start. This method acc normalized offsets values bet and 1 and performs validatior using the value.
- <b>=</b>	Start	Called by Unity. Allows the ac replay manager to initialize.
= <b>≬ S</b>	StopPlayback	Use this method to stop any a playback. This method will or an effect if there is an active p running otherwise it will have effect.
<b>≕≬</b> S	StopRecording	Use this method to stop recor after a previous call to BeginRecording(Boolean). The manager must already be record otherwise this method will have effect. This method must be cord before attempting to playback otherwise you may get unpre- results.
=♥	Update	Called by Unity. Allows the ac replay manager to update rec playback.

### Тор

## ⊿ See Also

# ReplayManagerBeginPlayback Method

Use this method to begin the playback of the recorded objects. Use SetPlaybackFrame(Single, PlaybackOrigin) or SetPlaybackFrameNormalized(Single, PlaybackOrigin) before calling this method to specify the exact location at which playback should begin. This method will run the entire playback gathered and then automatically stop playback on completion if endReplayAfterPlayback is true.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

```
C#
public static void BeginPlayback(
            bool fromStart = true,
            PlaybackDirection direction = PlaybackDire(
)
```

### Parameters

fromStart (Optional)

Type: SystemBoolean

When true, the replay will be played from the first frame recorded

### direction (Optional)

Type: UltimateReplayPlaybackDirection

The direction that the replay should be played

## ⊿ See Also

# ReplayManagerBeginPlaybackFram Method

Use this method to set the current playback at a specific replay frame. This will allow the state of a specific replay frame to be restored but will not continue playback which will provide a freeze frame effect. Use SetPlaybackFrame(Single, PlaybackOrigin) or

SetPlaybackFrameNormalized(Single, PlaybackOrigin) before calling this method to specify the exact location at which the playback frame should be sampled. Use StopPlayback to unfreeze the still frame and return to normal game mode. This method will ignore the value of endReplayAfterPlayback as only a single frame is replayed. As a result you will need to call StopPlayback when you want to end the playback frame.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public static void BeginPlaybackFrame()

## ⊿ See Also

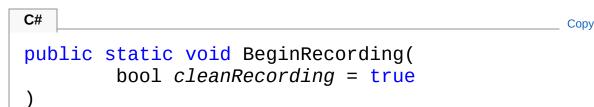
# ReplayManagerBeginRecording Method

Use this method to begin sampling the recorded objects in the scene. If recordOnStart is true then this method will be called automatically when the manager is initialized. Any state information will be recored via the assigned ReplayTarget (Default ReplayMemoryTarget). When true, any previous recording data will be discarded

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax



#### Parameters

cleanRecording (Optional)

Type: SystemBoolean [Missing <param name="cleanRecording"/> documentation for "M:UltimateReplay.ReplayManager.BeginRecording(System.Boolean)"]

### ▲ See Also

# ReplayManagerDiscardRecording Method

This method will throw away any recorded data and flush the replay target if necessary. This method can be called at any time. If the manager is currently recording then all previous data will be discarded and recording will continue. If the manager is currently replaying then all replay data will be discarded and playback will stop.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public static void DiscardRecording()

## ⊿ See Also

# ReplayManagerFindReplayPrefab Method

Attempts to find the prefab with the matching name. This is used to restore objects that were destroyed during recording.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

## ▲ Syntax

**C**#

)

### Parameters

prefabName

Type: SystemString The name of the prefab to locate

Return Value Type: **GameObject** The matching prefab or null if no matching prefab was found

### ⊿ See Also

# ReplayManagerForceAwake Method

Override implementation of ForceAwake. Performs exactly the same behaviour. Simply included so that the user does not need to import 'UltimateReplay.Util' to access the method.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public static void ForceAwake()

## ⊿ See Also

# ReplayManagerOnDestroy Method

Called by Unity. Allows the active replay manager to cleanup any active recordings.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

public void OnDestroy()

### ⊿ See Also

# ReplayManagerOnLevelWasLoaded Method

Called by Unity. Allows the active replay manager to cleanup recordings when a scene change is made.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

```
▲ Syntax
```

**C**#

)

### Parameters

index

Type: SystemInt32 The level id

### ⊿ See Also

# ReplayManagerOnValidate Method

Called by Unity.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Сору

public void OnValidate()

## ⊿ See Also

# ReplayManagerPausePlayback Method

Use this method to pause replay playback while maintinaing the current replay state. See ResumePlayback to continue a playback.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Сору

public static void PausePlayback()

### ⊿ See Also

# ReplayManagerPauseRecording Method

Use this method when you want to pause recording but may continue recording at any point. A good candidate for pausing recording is when the user pauses the game and is shown a pause menu. The manager must already be recording otherwise this method will have no effect.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Сору

public static void PauseRecording()

## ⊿ See Also

# ReplayManagerRegisterReplayPrefa Method

Attempts to register a game object as a prefab so that the replay system is able to spawn or despawn the object as needed. You only need to do this for objects that are likley to be either instantiated or destroyed during recording. The replay system will then be able to accuratley restore the scene state during playback. The specified object must be a prefab otherwise an error will be thrown and the object will not be registered. Prefab instances are not accepted.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

```
C#
        Copy ______
public static void RegisterReplayPrefab(
            GameObject prefab
)
```

### Parameters

### prefab

Type: **GameObject** The prefab object to register with the replay system

## ▲ See Also

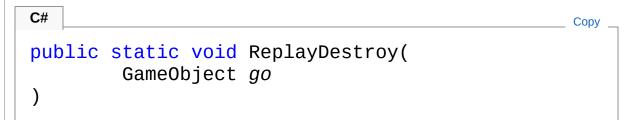
# ReplayManagerReplayDestroy Method

Attempts to destroy the specified prefab. OnReplayDestroy will be called if a listener has been registered otherwise default destruction will be used.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax



### Parameters

go

Type: **GameObject** The game object to destroy

### ⊿ See Also

# ReplayManagerReplayInstantiate Method

Attempts to instantiate the specified prefab. OnReplayInstantiate will be called if a listener has been registered otherwise default instantiation will be used.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

▲ Syntax

```
C#

public static GameObject ReplayInstantiate(

GameObject prefab,

Vector3 position,

Quaternion rotation

)
```

Parameters

prefab

Type: GameObject

The prefab to instantiate

position

Type: Vector3

The position to spawn the prefab at

rotation

Type: **Quaternion** 

The rotation to spawn the prefab with

Return Value Type: GameObject The new instance of the specified prefab or null if an error occurred

## ⊿ See Also

# ReplayManagerResumePlayback Method

Use this method to resume playback after a previous call to PausePlayback was called. If PausePlayback was not called prior to this method then the method will have no effect.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public static void ResumePlayback()

## ⊿ See Also

# ReplayManagerResumeRecording Method

Use this method to resume recording after a previous call to PauseRecording. The manager must already be recording otherwise this method will have no effect.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public static void ResumeRecording()

## ⊿ See Also

# ReplayManagerSetPlaybackFrame Method

Use this method to specify where in the replay sequence the playback should start. If the offset does not lie within the bounds of the replay then the value will be clamped to represent either the start or end frame.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

▲ Syntax

```
C#
Copy
public static void SetPlaybackFrame(
    float offset,
    PlaybackOrigin origin = PlaybackOrigin.Sta
)
```

### Parameters

offset

Type: SystemSingle

The amount of time in seconds to offset the playback

#### origin (Optional)

Type: UltimateReplayPlaybackOrigin

The playback node to take the offset from. If End is specified then the offset value will be used as a negative offset

### ⊿ See Also

Reference

# ReplayManagerSetPlaybackFrameN Method

Use this method to specify where in the replay sequence the playback should start. This method accepts normalized offsets values between 0 and 1 and performs validation before using the value.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

```
C#
        Copy
public static void SetPlaybackFrameNormalized(
        float normalizedOffset,
        PlaybackOrigin origin = PlaybackOrigin.Sta
)
```

#### Parameters

normalizedOffset

Type: SystemSingle

The normalized value representing the offset from the specified origin to start the playback from

#### origin (Optional)

Type: UltimateReplayPlaybackOrigin

The playback node to take the offset from. If End is specified then the offset value will be used as a negative offset

### ⊿ See Also

Reference

# ReplayManagerStart Method

Called by Unity. Allows the active replay manager to initialize.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

**C**#

public void Start()

#### ⊿ See Also

# ReplayManagerStopPlayback Method

Use this method to stop any active playback. This method will only have an effect if there is an active playback running otherwise it will have no effect.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Сору

public static void StopPlayback()

### ⊿ See Also

# ReplayManagerStopRecording Method

Use this method to stop recording after a previous call to BeginRecording(Boolean). The manager must already be recording otherwise this method will have no effect. This method must be called before attempting to playback otherwise you may get unpredicatable results.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Сору

public static void StopRecording()

### ⊿ See Also

# ReplayManagerUpdate Method

Called by Unity. Allows the active replay manager to update recoring or playback.

Copy

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

public void Update()

#### ⊿ See Also

# **ReplayManager Properties**

The ReplayManager type exposes the following members.

### Properties

	Name	Description
📽 s	CurrentPlaybackTime	Get the current playback time in seconds. This value will never be greater than duration.
	CurrentPlaybackTimeNormalized	Get the current playback time as a normalized value between 0-1. 0 represents the starting frame of the recording and 1 represents the very last frame of the recording.
<b>₽ S</b>	IsPaused	Returns true when the active replay manager is in any paused state. Paused states could include Playback_Paused or Recording_Paused.
🖹 S	IsRecording	Returns true if the manager is currently

		recording the scene. Note: If recording is paused this value will still be true.
i≌ s	IsReplaying	Returns true if the manager is currently playing back previously recorded replay data. Note: if playback is paused this value will still be true.
i≊ s	PlaybackDirection	Gets the current PlaybackDirection of replay playback.
i≌ S	Preparer	Access the current IReplayPreparer that the active replay manager will use to prepare game objects for replay. By default a DefaultReplayPreparer is used.
S	Replay	Get the active replay manager in the scene. If no replay manager could be found then one will be created with default settings. This property may be null in a rare case where the active replay manager was destroyed in the same

		frame as an application quit event was issued. In this situation the replay manager cannot be recreated as it would cause leaked objects.
📽 S	Scene	Get the ReplayScene associated with the replay system.
<b>≌</b> S	Target	The current replay target that is being used to store the replay data. By default, the replay target is ReplayMemoryTarget.

#### Тор

### ⊿ See Also

# ReplayManagerCurrentPlaybackTim Property

Get the current playback time in seconds. This value will never be greater than duration.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

public static float CurrentPlaybackTime { get; }

Copy \_

Property Value Type: Single

#### ⊿ See Also

# ReplayManagerCurrentPlaybackTim Property

Get the current playback time as a normalized value between 0-1. 0 represents the starting frame of the recording and 1 represents the very last frame of the recording.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

Property Value Type: Single

#### ⊿ See Also

# ReplayManagerIsPaused Property

Returns true when the active replay manager is in any paused state. Paused states could include Playback\_Paused or Recording\_Paused.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

public static bool IsPaused { get; }

Property Value Type: Boolean

### ⊿ See Also

# ReplayManagerIsRecording Property

Returns true if the manager is currently recording the scene. Note: If recording is paused this value will still be true.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Сору

public static bool IsRecording { get; }

Property Value Type: Boolean

#### ⊿ See Also

# ReplayManagerIsReplaying Property

Returns true if the manager is currently playing back previously recorded replay data. Note: if playback is paused this value will still be true.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Сору

public static bool IsReplaying { get; }

Property Value Type: Boolean

#### ⊿ See Also

# ReplayManagerPlaybackDirection Property

Gets the current PlaybackDirection of replay playback.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ⊿ Syntax

**C**#

public static PlaybackDirection PlaybackDirection

Copy

Property Value Type: PlaybackDirection

### ▲ See Also

# ReplayManagerPreparer Property

Access the current IReplayPreparer that the active replay manager will use to prepare game objects for replay. By default a DefaultReplayPreparer is used.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

C#	Сору
<pre>public static IReplayPreparer Preparer { get; s</pre>	set;
	Þ

Property Value Type: IReplayPreparer

### ⊿ See Also

# ReplayManagerReplay Property

Get the active replay manager in the scene. If no replay manager could be found then one will be created with default settings. This property may be null in a rare case where the active replay manager was destroyed in the same frame as an application quit event was issued. In this situation the replay manager cannot be recreated as it would cause leaked objects.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

\_\_\_\_\_ Сору

public static ReplayManager Replay { get; }

Property Value Type: ReplayManager

⊿ See Also

# ReplayManagerScene Property

Get the ReplayScene associated with the replay system.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Сору

### ▲ Syntax

**C**#

public static ReplayScene Scene { get; }

Property Value Type: ReplayScene

#### ⊿ See Also

# ReplayManagerTarget Property

The current replay target that is being used to store the replay data. By default, the replay target is ReplayMemoryTarget.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

public static ReplayTarget Target { get; set; }

Copy \_

Property Value Type: ReplayTarget

### ⊿ See Also

# **ReplayObject Class**

Only one instance of ReplayObject can be added to any game object.

### Inheritance Hierarchy

```
SystemObject Object
Component
Behaviour
MonoBehaviour
UltimateReplayReplayObject
```

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

public sealed class ReplayObject : MonoBehaviour, IReplaySerialize

Copy

The ReplayObject type exposes the following members.

### Constructors

	Name	Description
.≓ <b>`</b>	ReplayObject	Initializes a new instance of the ReplayObject class
-		

Тор

## ▲ Methods

	Name	Description
≡Ŵ	Awake	Called by Unity.
≓Ŵ	OnDestroy	Called by Unity.
<b>≡</b> ©	OnReplayDeserialize	Called by the replay system when this ReplayObject should deserialize its replay data.
=♥	OnReplaySerialize	Called by the replay system when this ReplayObject should serialize its replay data.
<b>≞</b> \$	RebuildComponentList	Forces the object to refresh its list of observed components. Observed components are components which inherit from ReplayBehaviour and exist on either this game object or a child of this game object.

#### Тор

### **⊿** Fields

	Name	Description
۶	observedComponents	An array of ReplayBehaviour components that this object will serialize during recording. Dynamically adding replay components during recording is not supported.

# Top⊿ Properties

	Name	Description
	IsPrefab	Returns true when this game object is a prefab asset. Returns false when this game object is a scene object or prefab instance.
<b>**</b>	Prefabldentity	Get the prefab associated with this ReplayObject.
<b>**</b>	ReplayIdentity	Get the unique ReplayIdentity for this ReplayObject.
Тор		
▲ See Also		
Reference UltimateReplay Namespace		

# **ReplayObject Constructor**

Initializes a new instance of the ReplayObject class

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

**C**#

public ReplayObject()

### ⊿ See Also

# **ReplayObject Fields**

The ReplayObject type exposes the following members.

### ⊿ Fields

	Name	Description
٩	observedComponents	An array of ReplayBehaviour components that this object will serialize during recording. Dynamically adding replay components during recording is not supported.

Тор



# ReplayObjectobservedComponents Field

An array of ReplayBehaviour components that this object will serialize during recording. Dynamically adding replay components during recording is not supported.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Сору

public ReplayBehaviour[] observedComponents

Field Value Type: ReplayBehaviour

#### ⊿ See Also

# **ReplayObject Methods**

The ReplayObject type exposes the following members.

### ▲ Methods

	Name	Description
≡\$	Awake	Called by Unity.
≡Ŵ	OnDestroy	Called by Unity.
=♥	OnReplayDeserialize	Called by the replay system when this ReplayObject should deserialize its replay data.
=♥	OnReplaySerialize	Called by the replay system when this ReplayObject should serialize its replay data.
=♥	RebuildComponentList	Forces the object to refresh its list of observed components. Observed components are components which inherit from ReplayBehaviour and exist on either this game object or a child of this game object.

Тор

### ⊿ See Also

Reference ReplayObject Class UltimateReplay Namespace

# ReplayObjectAwake Method

Called by Unity.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

C#

public void Awake()

#### ⊿ See Also

# ReplayObjectOnDestroy Method

Called by Unity.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

**C**#

public void OnDestroy()

#### ⊿ See Also

# ReplayObjectOnReplayDeserialize Method

Called by the replay system when this ReplayObject should deserialize its replay data.

Copy

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

)

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to deserialize the data from

Implements IReplaySerializeOnReplayDeserialize(ReplayState)

### ⊿ See Also

# ReplayObjectOnReplaySerialize Method

Called by the replay system when this ReplayObject should serialize its replay data.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

)

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to serialize the data to

Implements IReplaySerializeOnReplaySerialize(ReplayState)

#### ⊿ See Also

# ReplayObjectRebuildComponentList Method

Forces the object to refresh its list of observed components. Observed components are components which inherit from ReplayBehaviour and exist on either this game object or a child of this game object.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Сору

public void RebuildComponentList()

### ⊿ See Also

# **ReplayObject Properties**

The ReplayObject type exposes the following members.

### ▲ Properties

	Name	Description
	IsPrefab	Returns true when this game object is a prefab asset. Returns false when this game object is a scene object or prefab instance.
	Prefabldentity	Get the prefab associated with this ReplayObject.
<b>*</b>	ReplayIdentity	Get the unique ReplayIdentity for this ReplayObject.
Ton		

Тор

⊿ See Also

# ReplayObjectIsPrefab Property

Returns true when this game object is a prefab asset. Returns false when this game object is a scene object or prefab instance.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

public bool IsPrefab { get; }

Property Value Type: Boolean

### ⊿ See Also

# ReplayObjectPrefabIdentity Property

Get the prefab associated with this ReplayObject.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public string PrefabIdentity { get; }

Property Value Type: String

#### ⊿ See Also

# ReplayObjectReplayIdentity Property

Get the unique ReplayIdentity for this ReplayObject.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ⊿ Syntax

C#

Сору

public ReplayIdentity ReplayIdentity { get; }

Property Value Type: ReplayIdentity

#### ⊿ See Also

# ReplayState Class

A ReplayState allows replay objects to serialize and deserialize their data. See IReplaySerialize.

## ▲ Inheritance Hierarchy

SystemObject UltimateReplayReplayState

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0

(1.0.0.0)

#### ▲ Syntax

**C**#

public sealed class ReplayState

The ReplayState type exposes the following members.

#### Constructors

	Name	Description
-a 🏟	ReplayState	Create an empty ReplayState that can be written to.
Тор		
⊿ Meth	ods	
	Name	Description
=	Clear	Clears all buffered data from

Сору

		this ReplayState and resets its state.
÷	Read16	Read a short from the state.
=	Read32	Read an int from the state.
=	ReadBool	Read a bool from the state.
<b>≡</b>	ReadByte	Read a byte from the state.
= <b>Q</b>	ReadBytes(Int32)	Read a byte array from the state.
=0	ReadBytes(Byte, Int32, Int32)	Fill a byte array with data from the state.
÷	ReadColor	Read a colour from the state.
=0	ReadColor32	Read a colour32 from the state.
÷	ReadFloat	Read a float from the state.
= <b>\$</b>	ReadFloatLowPrecision	Attempts to read a low precision float. You should only use this method when the value is relativley small (less than 65000).
≡∳	ReadIdentity	Read a ReplayIdentity from the state.
=∳	ReadQuat	Read a quaternion from the state.
≓Ŵ	ReadState	Read the specified amount of bytes as a new ReplayState.

=	ReadString	Read a string from the state
=0	ReadVec2	Read a vector2 from the state.
=♥	ReadVec3	Read a vector3 from the state.
=♥	ReadVec4	Read a vector4 from the state.
<b>≡©</b>	ToArray	Get the ReplayState data as a byte array.
= <b>(</b> )	TryReadObject	Attempts to read an object state from this ReplayState.
= <b>\$</b>	TryWriteObject	Attempts to write an object to this ReplayState. This method may write extra meta data for deserialization purposes which may cause excessive storage size. Use one of the Write(Byte) methods if the type is known at compile time.
- <b>=</b>	Write(Boolean)	Write a bool to the state.
=∳	Write(Byte)	Write a byte to the state.
<b>≡∲</b>	Write(Byte)	Write a byte array to the state.
≡\$	Write(Int16)	Write a short to the state.
- <b>=</b>	Write(Int32)	Write an int to the state.
= <b>Q</b>	Write(Single)	Write a float to the state.
= <b>Q</b>	Write(String)	Write a string to the state.
=∳	Write(Color)	Write a colour to the state.

.⊒∳	Write(Colo	or32)	Write a colour32 value to the state.
÷	Write(Quat	ternion)	Write a quaternion to the state.
=0	Write(Vect	or2)	Write a vector2 to the state.
=•	Write(Vect	or3)	Write a vector3 to the state.
=	Write(Vect	or4)	Write a vector4 to the state.
÷≓∳	Write(Repl	layIdentity)	Write the specified replay identity to this ReplayState.
≡Ŷ	Write(Repl	layState)	Write the entire contents of a ReplayState to this ReplayState. All bytes will be appended.
÷	Write(Byte	, Int32, Int32)	Write a byte array to the state using an offset position and length.
.≓♥	WriteLowF	Precision	Attempts to write a 32 bit float value as a low precision 16 bit representation. You should only use this method when the value is relativley small (less than 65000). Accuracy may be lost by storing low precision values.
Тор			
⊿ Prope	erties		
	Name	Description	
1			

	CanRead	Returns true if the state contains any more data.
	EndRead	Returns true if the read pointer is at the end of the buffered data or false if there is still data to be read.
	Size	Returns the size of the object state in bytes.
Тор		
▲ See Also		
Reference UltimateReplay Namespace		

## ReplayState Constructor

Create an empty ReplayState that can be written to.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

#### ▲ Syntax

C#

public ReplayState()

#### ⊿ See Also

## ReplayState Methods

The ReplayState type exposes the following members.

#### Methods

	Name	Description
-= 🌑	Clear	Clears all buffered data from this ReplayState and resets its state.
= <b>\</b>	Read16	Read a short from the state.
≡♥	Read32	Read an int from the state.
≡ <b>≬</b>	ReadBool	Read a bool from the state.
≡ <b>∲</b>	ReadByte	Read a byte from the state.
-= <b>Q</b>	ReadBytes(Int32)	Read a byte array from the state.
-= 🚱	ReadBytes(Byte, Int32, Int32)	Fill a byte array with data from the state.
≡ <b>≬</b>	ReadColor	Read a colour from the state.
-=•	ReadColor32	Read a colour32 from the state.
= <b>0</b>	ReadFloat	Read a float from the state.
- <b>=</b>	ReadFloatLowPrecision	Attempts to read a low precision float. You should

		only use this method when the value is relativley small (less than 65000).
æ	ReadIdentity	Read a ReplayIdentity from the state.
- <b>:</b>	ReadQuat	Read a quaternion from the state.
- <b>:</b>	ReadState	Read the specified amount of bytes as a new ReplayState.
≡Ŵ	ReadString	Read a string from the state
æ	ReadVec2	Read a vector2 from the state.
æ	ReadVec3	Read a vector3 from the state.
=	ReadVec4	Read a vector4 from the state.
÷	ToArray	Get the ReplayState data as a byte array.
÷	TryReadObject	Attempts to read an object state from this ReplayState.
<b>≞</b> ∲	TryWriteObject	Attempts to write an object to this ReplayState. This method may write extra meta data for deserialization purposes which may cause excessive storage size. Use one of the Write(Byte) methods if the type is known at compile time.
≡\$	Write(Boolean)	Write a bool to the state.
=		

	Write(Byte)	Write a byte to the state.
≡\$	Write(Byte)	Write a byte array to the state.
≡Ŵ	Write(Int16)	Write a short to the state.
≡Ŵ	Write(Int32)	Write an int to the state.
≡Ŵ	Write(Single)	Write a float to the state.
<b>≡</b>	Write(String)	Write a string to the state.
<b>=</b>	Write(Color)	Write a colour to the state.
<b>≞</b> ©	Write(Color32)	Write a colour32 value to the state.
<b>≞</b> ©	Write(Quaternion)	Write a quaternion to the state.
≡Ŵ	Write(Vector2)	Write a vector2 to the state.
≡\$	Write(Vector3)	Write a vector3 to the state.
≡∳	Write(Vector4)	Write a vector4 to the state.
<b>≡</b> ©	Write(ReplayIdentity)	Write the specified replay identity to this ReplayState.
<b>≡</b> ©	Write(ReplayState)	Write the entire contents of a ReplayState to this ReplayState. All bytes will be appended.
<b>≞</b> ©	Write(Byte, Int32, Int32)	Write a byte array to the state using an offset position and length.
= <b>Q</b>	WriteLowPrecision	Attempts to write a 32 bit float

value as a low precision 16 bit representation. You should only use this method when the value is relativley small (less than 65000). Accuracy may be lost by storing low precision values.

#### Тор

#### ⊿ See Also

## ReplayStateClear Method

Clears all buffered data from this ReplayState and resets its state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

#### ▲ Syntax

C#

public void Clear()

#### ⊿ See Also

## ReplayStateRead16 Method

Read a short from the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

C#

Copy \_

public short Read16()

Return Value Type: Int16 Short value

#### ⊿ See Also

## ReplayStateRead32 Method

Read an int from the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

C#

Copy \_

public int Read32()

Return Value Type: Int32 Int value

#### ⊿ See Also

## ReplayStateReadBool Method

Read a bool from the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

#### ▲ Syntax

**C**#

public bool ReadBool()

Return Value Type: Boolean Bool value

#### ⊿ See Also

## ReplayStateReadByte Method

Read a byte from the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Copy \_

public byte ReadByte()

Return Value Type: Byte Byte value

#### ⊿ See Also

## ReplayStateReadBytes Method

## Overload List

	Name	Description
<b>≡</b>	ReadBytes(Int32)	Read a byte array from the state.
<b>∃</b> û	ReadBytes(Byte, Int32, Int32)	Fill a byte array with data from the state.

Тор

#### ⊿ See Also

# ReplayStateReadBytes Method (Int32)

Read a byte array from the state.

```
Namespace: UltimateReplay
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

#### ▲ Syntax

**C**#

)

#### Parameters

amount Type: SystemInt32 The number of bytes to read

Return Value Type: Byte Byte array value

#### ⊿ See Also

Reference ReplayState Class ReadBytes Overload UltimateReplay Namespace

# ReplayStateReadBytes Method (Byte, Int32, Int32)

Fill a byte array with data from the state.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

#### ⊿ Syntax

**C**#

```
public void ReadBytes(
          byte[] buffer,
          int offset,
          int amount
)
```

/

Parameters

buffer

Type: SystemByte

The byte array to store data in

offset

Type: SystemInt32

The index offset to start filling the buffer at

amount

Type: SystemInt32 The number of bytes to read

#### ⊿ See Also

Reference

ReplayState Class ReadBytes Overload UltimateReplay Namespace

# ReplayStateReadColor Method

Read a colour from the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

#### ▲ Syntax

C#

public Color ReadColor()

Return Value Type: **Color** Colour value

#### ⊿ See Also

# ReplayStateReadColor32 Method

Read a colour32 from the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Copy \_

public Color32 ReadColor32()

Return Value Type: **Color32** Colour32 value

#### ⊿ See Also

## ReplayStateReadFloat Method

Read a float from the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

#### ▲ Syntax

**C**#

public float ReadFloat()

Return Value Type: Single Float value

#### ⊿ See Also

## ReplayStateReadFloatLowPrecision Method

Attempts to read a low precision float. You should only use this method when the value is relativley small (less than 65000).

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Copy \_

public float ReadFloatLowPrecision()

Return Value Type: Single float value

#### ⊿ See Also

# ReplayStateReadIdentity Method

Read a ReplayIdentity from the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

C#

Copy \_

public ReplayIdentity ReadIdentity()

Return Value Type: ReplayIdentity Identity value

#### ⊿ See Also

## ReplayStateReadQuat Method

Read a quaternion from the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Copy \_

public Quaternion ReadQuat()

Return Value Type: **Quaternion** Quaternion value

#### ⊿ See Also

## ReplayStateReadState Method

Read the specified amount of bytes as a new ReplayState.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

#### ▲ Syntax

**C**#

)

#### Parameters

bytes

Type: SystemInt32 The amount of bytes to read into the state

Return Value Type: ReplayState A new ReplayState containing the specified number of bytes

#### J See Also

## ReplayStateReadString Method

Read a string from the state

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

#### ▲ Syntax

**C**#

public string ReadString()

Return Value Type: String string value

#### ⊿ See Also

## ReplayStateReadVec2 Method

Read a vector2 from the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

#### ▲ Syntax

C#

public Vector2 ReadVec2()

Return Value Type: **Vector2** Vector2 value

#### ⊿ See Also

## ReplayStateReadVec3 Method

Read a vector3 from the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

#### ▲ Syntax

C#

public Vector3 ReadVec3()

Return Value Type: **Vector3** Vector3 value

#### ⊿ See Also

## ReplayStateReadVec4 Method

Read a vector4 from the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

#### ▲ Syntax

C#

public Vector4 ReadVec4()

Return Value Type: **Vector4** Vector4 value

#### ⊿ See Also

# ReplayStateToArray Method

Get the ReplayState data as a byte array.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Copy \_

public byte[] ToArray()

Return Value Type: Byte A byte array of data

#### ⊿ See Also

## ReplayStateTryReadObject Method

Attempts to read an object state from this ReplayState.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Сору

```
public Object TryReadObject()
```

Return Value Type: Object The state data for the object

#### ⊿ See Also

# ReplayStateTryWriteObject Method

Attempts to write an object to this ReplayState. This method may write extra meta data for deserialization purposes which may cause excessive storage size. Use one of the Write(Byte) methods if the type is known at compile time.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax



#### Parameters

value

Type: SystemObject The object to write to the state

#### ⊿ See Also

# ReplayStateWrite Method

## Overload List

	Name	Description
=♥	Write(Boolean)	Write a bool to the state.
=\$	Write(Byte)	Write a byte to the state.
=\$	Write(Byte)	Write a byte array to the state.
≡Ŵ	Write(Int16)	Write a short to the state.
=	Write(Int32)	Write an int to the state.
=∳	Write(Single)	Write a float to the state.
=	Write(String)	Write a string to the state.
=	Write(Color)	Write a colour to the state.
=0	Write(Color32)	Write a colour32 value to the state.
≡Q	Write(Quaternion)	Write a quaternion to the state.
≡Q	Write(Vector2)	Write a vector2 to the state.
≡Q	Write(Vector3)	Write a vector3 to the state.
≡Q	Write(Vector4)	Write a vector4 to the state.
=0	Write(ReplayIdentity)	Write the specified replay identity to this ReplayState.

=♥	Write(ReplayState)	Write the entire contents of a ReplayState to this ReplayState. All bytes will be appended.
= <b>Q</b>	Write(Byte, Int32, Int32)	Write a byte array to the state using an offset position and length.

Тор

## ⊿ See Also

# ReplayStateWrite Method (Boolean)

Write a bool to the state.

```
Namespace: UltimateReplay
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

#### ▲ Syntax

**C**#

)

#### Parameters

value Type: SystemBoolean bool value

▲ See Also

# ReplayStateWrite Method (Byte)

Write a byte to the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Сору

## ▲ Syntax

Parameters

value

**C**#

Type: SystemByte Byte value

## ⊿ See Also

# ReplayStateWrite Method (Byte)

Copy

Write a byte array to the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

C#

Parameters

bytes

Type: SystemByte Byte array value

## ⊿ See Also

# ReplayStateWrite Method (Int16)

Write a short to the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Сору

## ▲ Syntax

C#

)

Parameters

value Type: SystemInt16 Short value

## ⊿ See Also

# ReplayStateWrite Method (Int32)

Write an int to the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Сору \_

Parameters

value Type: SystemInt32 Int value

## ⊿ See Also

# ReplayStateWrite Method (Single)

Write a float to the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Сору

## ▲ Syntax

C#

public void Write(
 float value
)

Parameters

value Type: SystemSingle Float value

## ⊿ See Also

# ReplayStateWrite Method (String)

Write a string to the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

## ▲ Syntax

**C**#

public void Write(
 string value
)

,

Parameters

value Type: SystemString string value

## ⊿ See Also

# ReplayStateWrite Method (Color)

Write a colour to the state.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

Сору \_

public void Write(
 Color value
)

)

**C**#

Parameters

value

Type: **Color** Colour value

## ⊿ See Also

# ReplayStateWrite Method (Color32)

Write a colour32 value to the state.

```
Namespace: UltimateReplay
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

#### ⊿ Syntax

**C**#

```
public void Write(
        Color32 value
)
```

)

#### Parameters

value

Type: **Color32** Colour32 value



# ReplayStateWrite Method (Quaternion)

Write a quaternion to the state.

```
Namespace: UltimateReplay
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

#### ⊿ Syntax

**C**#

```
public void Write(
Quaternion value
```

)

#### Parameters

value

Type: **Quaternion** Quaternion value

⊿ See Also

# ReplayStateWrite Method (Vector2)

Write a vector2 to the state.

```
Namespace: UltimateReplay
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

#### ⊿ Syntax

**C**#

)

#### Parameters

value

Type: Vector2 Vector2 value



# ReplayStateWrite Method (Vector3)

Write a vector3 to the state.

```
Namespace: UltimateReplay
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

#### ⊿ Syntax

**C**#

)

#### Parameters

value

Type: Vector3 Vector3 value



# ReplayStateWrite Method (Vector4)

Write a vector4 to the state.

```
Namespace: UltimateReplay
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

#### ⊿ Syntax

**C**#

)

#### Parameters

value

Type: **Vector4** Vector4 value



# ReplayStateWrite Method (ReplayIdentity)

Write the specified replay identity to this ReplayState.

```
Namespace: UltimateReplay
```

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

```
Сору
```

#### Parameters

identity

Type: UltimateReplay.CoreReplayIdentity The identity to write

## ⊿ See Also

# ReplayStateWrite Method (ReplayState)

Write the entire contents of a ReplayState to this ReplayState. All bytes will be appended.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

```
▲ Syntax
```

**C**#

Parameters

other

Type: UltimateReplayReplayState The other state to append

#### ⊿ See Also

# ReplayStateWrite Method (Byte, Int32, Int32)

Write a byte array to the state using an offset position and length.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

#### ⊿ Syntax

**C**#

```
public void Write(
    byte[] bytes,
    int offset,
    int length
)
```

Parameters

bytes

Type: SystemByte Byte array value

offset

Type: SystemInt32

The start index to read data from the array

length

Type: SystemInt32 The amount of data to read

#### ⊿ See Also

Reference

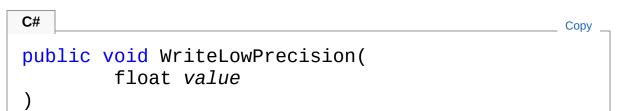
# ReplayStateWriteLowPrecision Method

Attempts to write a 32 bit float value as a low precision 16 bit representation. You should only use this method when the value is relativley small (less than 65000). Accuracy may be lost by storing low precision values.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax



#### Parameters

value

Type: SystemSingle float value

## ⊿ See Also

## **ReplayState Properties**

The ReplayState type exposes the following members.

#### ▲ Properties

	Name	Description
	CanRead	Returns true if the state contains any more data.
<b>*</b>	EndRead	Returns true if the read pointer is at the end of the buffered data or false if there is still data to be read.
	Size	Returns the size of the object state in bytes.

Тор



# ReplayStateCanRead Property

Returns true if the state contains any more data.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

C#

public bool CanRead { get; }

Property Value Type: Boolean

#### ⊿ See Also

# ReplayStateEndRead Property

Returns true if the read pointer is at the end of the buffered data or false if there is still data to be read.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

public bool EndRead { get; }

Property Value Type: Boolean

#### ⊿ See Also

# ReplayStateSize Property

Returns the size of the object state in bytes.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0

Copy \_

(1.0.0.0)

#### ▲ Syntax

**C**#

public int Size { get; }

Property Value Type: Int32

#### ⊿ See Also

# **ReplayTime Class**

This class emulates the behaviour of the Time class in Unity and can be used to modify the playback speed of a replay. There are also delta values that can be used to interpolate between frames where a low record frame rate is used. See ReplayTransform for an example.

## ▲ Inheritance Hierarchy

SystemObject UltimateReplayReplayTime

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Сору

public static class ReplayTime

The ReplayTime type exposes the following members.

## Methods

	Name	Description
<b>≡∳ S</b>	GetCorrectedTimeValueString	Gets the current time as a float and converts it to minutes and seconds formatted as a string.

Тор

## ▲ Properties

	Name	Description
🖻 s	Delta	Represents a delta between current replay frames. This normalized value can be used to interpolate smoothly between replay states where a low record rate is used. Note: this value is not the actual delta time but a value representing the transition progress between replay frames.
🖹 S	Time	Get the current replay playback time.
📽 s	TimeScale	The time scale value used during replay playback. This value is ignored during replay recording.

Тор



Reference UltimateReplay Namespace

# **ReplayTime Methods**

The ReplayTime type exposes the following members.

## ▲ Methods

	Name	Description
= <b>0 S</b>	GetCorrectedTimeValueString	Gets the current time as a float and converts it to minutes and seconds formatted as a string.

Тор

#### ⊿ See Also

## ReplayTimeGetCorrectedTimeValue Method

Gets the current time as a float and converts it to minutes and seconds formatted as a string.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

```
public static string GetCorrectedTimeValueString(
    float timeValue
)
```

Copy

#### Parameters

timeValue

Type: SystemSingle

The time value input, for example: Time.time

Return Value Type: String A formatted time string

#### ▲ See Also

# **ReplayTime Properties**

The ReplayTime type exposes the following members.

## ▲ Properties

	Name	Description
in an	Delta	Represents a delta between current replay frames. This normalized value can be used to interpolate smoothly between replay states where a low record rate is used. Note: this value is not the actual delta time but a value representing the transition progress between replay frames.
🖀 2	Time	Get the current replay playback time.
📽 s	TimeScale	The time scale value used during replay playback. This value is ignored during replay recording.

Тор

#### ⊿ See Also

# ReplayTimeDelta Property

Represents a delta between current replay frames. This normalized value can be used to interpolate smoothly between replay states where a low record rate is used. Note: this value is not the actual delta time but a value representing the transition progress between replay frames.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)



**C**#

Copy

public static float Delta { get; }

Property Value Type: Single

#### ⊿ See Also

# ReplayTimeTime Property

Get the current replay playback time.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

#### ▲ Syntax

**C**#

public static float Time { get; }

Property Value Type: Single

#### ⊿ See Also

# ReplayTimeTimeScale Property

The time scale value used during replay playback. This value is ignored during replay recording.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax



Property Value Type: Single

#### ⊿ See Also

# ReplayTransform Class

Attach this component to a game objects in order to record the objects transform for replays. Only one instance of ReplayTransform can be added to any game object.

#### ▲ Inheritance Hierarchy

SystemObject Object Component Behaviour MonoBehaviour UltimateReplayReplayBehaviour UltimateReplayReplayTransform

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

C#

Сору

public class ReplayTransform : ReplayBehaviour

The ReplayTransform type exposes the following members.

## Constructors

	Name	Description
≓ <b>©</b>	ReplayTransform	Initializes a new instance of the ReplayTransform class

Тор

## ▲ Methods

	Name	Description
≡♥	Awake	Called by Unity. (Overrides ReplayBehaviourAwake.)
=∳	OnReplayDeserialize	Called by the relay system when the ol return to a previous state. (Overrides ReplayBehaviourOnReplayDeserialize
=∳	OnReplaySerialize	Called by the replay system when the be recorded. (Overrides ReplayBehaviourOnReplaySerialize(R
≡♥	OnReplayUpdate	Called during playback and allows the be interpolated to provide a smooth re lower record rates are used. (Overrides ReplayBehaviourOnReplay

#### Тор

## **▲** Fields

	Name	Description
٥	interpolate	Should the transform be interpolated between states. This is recommended when low record rates are used as without interpolation the playback can seem jumpy.
٥	recordPosition	Should the position value of the transform be recorded.

•	recordRotation	Should the rotation value of the transform be recorded.
•	recordScale	Should the scale value of the transform be recorded.
Тор		
⊿ See Also		
Reference UltimateRepl	ay Namespace	

# ReplayTransform Constructor

Initializes a new instance of the ReplayTransform class

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

#### ▲ Syntax

**C**#

public ReplayTransform()

#### ⊿ See Also

# ReplayTransform Fields

The ReplayTransform type exposes the following members.

#### ▲ Fields

	Name	Description
٩	interpolate	Should the transform be interpolated between states. This is recommended when low record rates are used as without interpolation the playback can seem jumpy.
Ŷ	recordPosition	Should the position value of the transform be recorded.
Ŷ	recordRotation	Should the rotation value of the transform be recorded.
Ŷ	recordScale	Should the scale value of the transform be recorded.

Тор

#### ⊿ See Also

# ReplayTransforminterpolate Field

Should the transform be interpolated between states. This is recommended when low record rates are used as without interpolation the playback can seem jumpy.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

Сору

public bool interpolate

Field Value Type: Boolean

#### ⊿ See Also

# ReplayTransformrecordPosition Field

Should the position value of the transform be recorded.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public bool recordPosition

Field Value Type: Boolean

### ⊿ See Also

# ReplayTransformrecordRotation Field

Should the rotation value of the transform be recorded.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public bool recordRotation

Field Value Type: Boolean

### ⊿ See Also

# ReplayTransformrecordScale Field

Should the scale value of the transform be recorded.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

C#

public bool recordScale

Field Value Type: Boolean

### ⊿ See Also

# **ReplayTransform Methods**

The ReplayTransform type exposes the following members.

## Methods

	Name	Description
<b>≡∲</b>	Awake	Called by Unity. (Overrides ReplayBehaviourAwake.)
≓ <b>∲</b>	OnReplayDeserialize	Called by the relay system when the o return to a previous state. (Overrides ReplayBehaviourOnReplayDeserialize
≓ <b>∲</b>	OnReplaySerialize	Called by the replay system when the be recorded. (Overrides ReplayBehaviourOnReplaySerialize(R
<b>≡</b> ©	OnReplayUpdate	Called during playback and allows the be interpolated to provide a smooth rej lower record rates are used. (Overrides ReplayBehaviourOnReplay

Тор

⊿ See Also

# ReplayTransformAwake Method

Called by Unity.

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

**C**#

public override void Awake()

### ⊿ See Also

# ReplayTransformOnReplayDeseriali Method

Called by the relay system when the object should return to a previous state.

Copy

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

)

#### Parameters

state

Type: UltimateReplayReplayState

The state object to deserialize the transform from

Implements

IReplaySerializeOnReplayDeserialize(ReplayState)

### ⊿ See Also

# ReplayTransformOnReplaySerialize Method

Called by the replay system when the object should be recorded.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

Сору

#### )

#### Parameters

state

Type: UltimateReplayReplayState The state object to serialize the transform into

Implements

IReplaySerializeOnReplaySerialize(ReplayState)

### ⊿ See Also

# ReplayTransformOnReplayUpdate Method

Called during playback and allows the transform to be interpolated to provide a smooth replay even if lower record rates are used.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Сору

public override void OnReplayUpdate()

### ⊿ See Also

# ReplayVarAttribute Class

Use this attribute on a field to mark it for recording. The type the field is defined in must inheit from ReplayBehaviour in order for the field to be recorded automatically. Interpolation between field values is also possible where low record rates are used.

## Inheritance Hierarchy

SystemObject SystemAttribute UltimateReplayReplayVarAttribute

Namespace: UltimateReplay Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax



The ReplayVarAttribute type exposes the following members.

## Constructors

	Name	Description
= <b>\$</b>	ReplayVarAttribute	Create a new ReplayVarAttribute for a field.

#### Тор

▲ Fields

	Name	Description
•	interpolate	Should the value of the field be interpolated between frames or should the value snap to the exact frame value. Most built-in types support interpolation such as Byte and Single. Basic Unity types such as <b>Vector2</b> and <b>Color</b> also support interpolation.

Тор

## ⊿ See Also

Reference UltimateReplay Namespace

# ReplayVarAttribute Constructor

Create a new ReplayVarAttribute for a field.

Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

## ▲ Syntax

**C**#

public ReplayVarAttribute(
 bool interpolated = true
)

#### Parameters

interpolated (Optional)

Type: SystemBoolean Should the field value be interpolated between frames

## ⊿ See Also

Reference ReplayVarAttribute Class UltimateReplay Namespace

# ReplayVarAttribute Fields

The ReplayVarAttribute type exposes the following members.

## ▲ Fields

	Name	Description
٩	interpolate	Should the value of the field be interpolated between frames or should the value snap to the exact frame value. Most built-in types support interpolation such as Byte and Single. Basic Unity types such as <b>Vector2</b> and <b>Color</b> also support interpolation.

Тор

### ⊿ See Also

Reference ReplayVarAttribute Class UltimateReplay Namespace

# ReplayVarAttributeinterpolate Field

Should the value of the field be interpolated between frames or should the value snap to the exact frame value. Most built-in types support interpolation such as Byte and Single. Basic Unity types such as **Vector2** and **Color** also support interpolation.

#### Namespace: UltimateReplay

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public bool interpolate

Field Value Type: Boolean

### ⊿ See Also

Reference ReplayVarAttribute Class UltimateReplay Namespace

## UltimateReplay.Core Namespace

## ▲ Classes

	Class	Description
*\$	DefaultReplayPreparer	The default IReplayPreparer used by Ultimate Replay to prepare game objects for gameplay and playback.
*	ReplayIdentity	A replay identity is an essential component in the Ultimate Replay system and is used to identify replay objects between sessions. Replay identities are assigned at edit time where possible and will never change values. Replay identities are also use to identify prefab instances that are spawned during a replay.
<b>*</b> \$	ReplayScene	A ReplayScene contains information about all active replay objects.
*\$	ReplayVariable	Represents a variable that can be recorded using the replay system in order to replay script animations or similar during playback.

## ▲ Structures

Structure	Description
ReplayEvent	Used to mark key recording events that will be triggered during playback. Good candidates would be to trigger audio effects or similar.

## ▲ Interfaces

	Interface	Description
⊷0	IReplayPreparer	A preparer is used by Ultimate Replay to prepare any replay objects for either gameplay mode or playback mode. In order for game systems such as physics and scritps to not affect playback, replay objects must be prepared in some way to disable these systems while playback is enabled. The appropriate prepare method will be called by the replay system when objects need to either enter playback mode or return to gameplay mode.
⊷0	IReplaySerialize	This class should be implemented when you want to serialize custom replay data. This sould really be an interface but it needs to be a class to be assignable in the inspector.

## ▲ Enumerations

ReplayEvents Common events identifiers used to record ReplayEvent with the replay system.		Enumeration	Description
	3 <sup>21</sup>	ReplayEvents	record ReplayEvent with the replay

# DefaultReplayPreparer Class

The default IReplayPreparer used by Ultimate Replay to prepare game objects for gameplay and playback.

## Inheritance Hierarchy

SystemObject UltimateReplay.CoreDefaultReplayPreparer

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax



The DefaultReplayPreparer type exposes the following members.

### Constructors

	Name	Description
.≡∳	DefaultReplayPreparer	Initializes a new instance of the DefaultReplayPreparer class
Тор		
▲ Metho	ds	
	Name	Description
=0		

	PrepareComponentForGameplay	Prepare the specified component for gameplay mode.
= <b>\$</b>	PrepareComponentForPlayback	Prepare the specified component for playback mode.
= <b>Q</b>	PrepareForGameplay	Prepare the specified replay object for gameplay mode.
≓Ŵ	PrepareForPlayback	Prepare the specified replay object for playback mode.

#### Тор

## ⊿ See Also

Reference UltimateReplay.Core Namespace

# DefaultReplayPreparer Constructor

Initializes a new instance of the DefaultReplayPreparer class

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

Сору

public DefaultReplayPreparer()

### ⊿ See Also

# DefaultReplayPreparer Methods

The DefaultReplayPreparer type exposes the following members.

## ▲ Methods

	Name	Description
≓ <b>©</b>	PrepareComponentForGameplay	Prepare the specified component for gameplay mode.
≓Q	PrepareComponentForPlayback	Prepare the specified component for playback mode.
≓ <b>©</b>	PrepareForGameplay	Prepare the specified replay object for gameplay mode.
<b>≞</b> ©	PrepareForPlayback	Prepare the specified replay object for playback mode.

#### Тор

⊿ See Also

# DefaultReplayPreparerPrepareCom Method

Prepare the specified component for gameplay mode.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

```
public virtual void PrepareComponentForGameplay(
        Component component
```

Copy

)

#### Parameters

*component* Type: **Component** The component to prepare

### ▲ See Also

# DefaultReplayPreparerPrepareCom Method

Prepare the specified component for playback mode.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

Copy

)

#### Parameters

*component* Type: **Component** The component to prepare

### ▲ See Also

# DefaultReplayPreparerPrepareForG Method

Prepare the specified replay object for gameplay mode.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

Сору

#### )

#### Parameters

replayObject

Type: UltimateReplayReplayObject The replay object to prepare

Implements

IReplayPreparerPrepareForGameplay(ReplayObject)

## ⊿ See Also

# DefaultReplayPreparerPrepareForP Method

Prepare the specified replay object for playback mode.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

**C**#

Сору

#### )

#### Parameters

replayObject

Type: UltimateReplayReplayObject The replay object to prepare

Implements

IReplayPreparerPrepareForPlayback(ReplayObject)

## ⊿ See Also

## IReplayPreparer Interface

A preparer is used by Ultimate Replay to prepare any replay objects for either gameplay mode or playback mode. In order for game systems such as physics and scritps to not affect playback, replay objects must be prepared in some way to disable these systems while playback is enabled. The appropriate prepare method will be called by the replay system when objects need to either enter playback mode or return to gameplay mode.

#### Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public interface IReplayPreparer

The IReplayPreparer type exposes the following members.

## Methods

	Name	Description
=♥	PrepareForGameplay	Prepares the specified replay object for gameplay. The implementing type should restore all game systems that affect the replay object so that the object is in its original state. This method will be called for each replay object that must be prepared.

PrepareForPlayback	Prepares the specified replay object for playback. The implementing type should ensure that all game systems likley to affect the replay object during playback are suitable disabled in order to avoid glitching or unpredicted behaviour. This method will be called for each replay object that must be prepared.
--------------------	--

### Тор

## ⊿ See Also

Reference UltimateReplay.Core Namespace

# **IReplayPreparer Methods**

The IReplayPreparer type exposes the following members.

## ▲ Methods

	Name	Description
	PrepareForGameplay	Prepares the specified replay object for gameplay. The implementing type should restore all game systems that affect the replay object so that the object is in its original state. This method will be called for each replay object that must be prepared.
≓∳	PrepareForPlayback	Prepares the specified replay object for playback. The implementing type should ensure that all game systems likley to affect the replay object during playback are suitable disabled in order to avoid glitching or unpredicted behaviour. This method will be called for each replay object that must be prepared.

Тор

⊿ See Also

#### Reference IReplayPreparer Interface UltimateReplay.Core Namespace

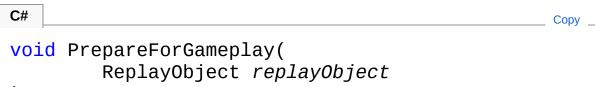
## IReplayPreparerPrepareForGamepla Method

Prepares the specified replay object for gameplay. The implementing type should restore all game systems that affect the replay object so that the object is in its original state. This method will be called for each replay object that must be prepared.

#### Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax



)

#### Parameters

replayObject

Type: UltimateReplayReplayObject The replay object to prepare

## ⊿ See Also

Reference IReplayPreparer Interface UltimateReplay.Core Namespace

# **IReplayPreparerPrepareForPlaybac** Method

Prepares the specified replay object for playback. The implementing type should ensure that all game systems likley to affect the replay object during playback are suitable disabled in order to avoid glitching or unpredicted behaviour. This method will be called for each replay object that must be prepared.

#### Namespace: UltimateReplay.Core

Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

)

Copy void PrepareForPlayback(

ReplayObject replayObject

### Parameters

replayObject Type: UltimateReplayReplayObject The replay object that should be prepared

## ▲ See Also

Reference **IReplayPreparer Interface** UltimateReplay.Core Namespace

# IReplaySerialize Interface

This class should be implemented when you want to serialize custom replay data. This sould really be an interface but it needs to be a class to be assignable in the inspector.

#### Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

public interface IReplaySerialize

The IReplaySerialize type exposes the following members.

### Methods

	Name	Description
≓ <b>©</b>	OnReplayDeserialize	Called by the replay system when all replay state data should be deserialized.
≓Q	OnReplaySerialize	Called by the replay system when all replay state data should be serialized.

Тор

⊿ See Also

Reference

UltimateReplay.Core Namespace

# **IReplaySerialize Methods**

The IReplaySerialize type exposes the following members.

## Methods

	Name	Description
=∳	OnReplayDeserialize	Called by the replay system when all replay state data should be deserialized.
=♥	OnReplaySerialize	Called by the replay system when all replay state data should be serialized.

Тор

## ⊿ See Also

Reference IReplaySerialize Interface UltimateReplay.Core Namespace

# IReplaySerializeOnReplayDeserializ Method

Called by the replay system when all replay state data should be deserialized.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

)

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to read the data from

### ⊿ See Also

Reference IReplaySerialize Interface UltimateReplay.Core Namespace

# IReplaySerializeOnReplaySerialize Method

Called by the replay system when all replay state data should be serialized.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

)

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to write the data to

### ⊿ See Also

Reference IReplaySerialize Interface UltimateReplay.Core Namespace

# **ReplayEvent Structure**

Used to mark key recording events that will be triggered during playback. Good candidates would be to trigger audio effects or similar.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

C# \_\_\_\_\_\_\_\_\_Copy \_\_\_\_\_\_Copy \_\_\_\_\_Copy \_\_\_\_COPY \_\_\_\_\_COPY \_\_\_\_COPY \_\_\_\_\_COPY \_\_\_\_COPY \_\_\_\_\_COPY \_\_\_\_COPY \_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_COPY \_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_\_COPY \_\_\_COPY \_\_COPY \_\_\_COPY \_\_COPY \_\_COPY \_\_COPY \_\_COPY \_\_COPY \_\_COPY \_\_COPY \_\_COPY \_COPY \_CO

The ReplayEvent type exposes the following members.

## Methods

	Name	Description
=♥	OnReplayDeserialize	Called by the replay system when all replay state information should be deserialized.
<b>≓</b> ©	OnReplaySerialize	Called by the replay system when all replay state information should be serialized.
Тор		
<b>▲</b> Fields		

Name	Description	

•	eventData	The replay state data associated with this ReplayEvent. The event data should contain no more than 255 bytes to ensure that the data is serialized correctly.
۵	eventID	A unique event identifier used to distinguish between different replay events.
Тор		
⊿ See	e Also	

Reference UltimateReplay.Core Namespace

# **ReplayEvent Fields**

The ReplayEvent type exposes the following members.

## ▲ Fields

	Name	Description
	eventData	The replay state data associated with this ReplayEvent. The event data should contain no more than 255 bytes to ensure that the data is serialized correctly.
Ŷ	eventID	A unique event identifier used to distinguish between different replay events.

Тор

### ⊿ See Also

# ReplayEventeventData Field

The replay state data associated with this ReplayEvent. The event data should contain no more than 255 bytes to ensure that the data is serialized correctly.

#### Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public ReplayState eventData

Field Value Type: ReplayState

### ⊿ See Also

## ReplayEventeventID Field

A unique event identifier used to distinguish between different replay events.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

public byte eventID

Field Value Type: Byte

### ⊿ See Also

# **ReplayEvent Methods**

The ReplayEvent type exposes the following members.

### Methods

	Name	Description
=∳	OnReplayDeserialize	Called by the replay system when all replay state information should be deserialized.
=♥	OnReplaySerialize	Called by the replay system when all replay state information should be serialized.

Тор

### ⊿ See Also

## ReplayEventOnReplayDeserialize Method

Called by the replay system when all replay state information should be deserialized.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

)

### Parameters

state

Type: UltimateReplayReplayState The ReplayState to read the data from

Implements IReplaySerializeOnReplayDeserialize(ReplayState)

### ⊿ See Also

## ReplayEventOnReplaySerialize Method

Called by the replay system when all replay state information should be serialized.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

)

### Parameters

state

Type: UltimateReplayReplayState The ReplayState to write the data to

Implements IReplaySerializeOnReplaySerialize(ReplayState)

### ⊿ See Also

## **ReplayEvents Enumeration**

Common events identifiers used to record ReplayEvent with the replay system.

Copy

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ⊿ Syntax

**C**#

public enum ReplayEvents

### Members

Member name	Value	Description
ObjectSpawn	1	Object instantiated event.
ObjectDespawn	2	Object destroyed event.
PlaySound	3	Play audio clip event.
PlayMusic	4	Play audio music event.
ParticleStart	5	Start particle system event.
ParticleEnd	6	Stop particle system event.

### ⊿ See Also

Reference UltimateReplay.Core Namespace

# **ReplayIdentity Class**

A replay identity is an essential component in the Ultimate Replay system and is used to identify replay objects between sessions. Replay identities are assigned at edit time where possible and will never change values. Replay identities are also use to identify prefab instances that are spawned during a replay.

### ▲ Inheritance Hierarchy

SystemObject UltimateReplay.CoreReplayIdentity

Namespace: UltimateReplay.Core

```
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)
```

### ▲ Syntax

**C**#

```
[SerializableAttribute]
public sealed class ReplayIdentity : IEquatable<Re
</pre>
```

Copy

The ReplayIdentity type exposes the following members.

### Constructors

	Name	Description
. <b>⊒</b>	ReplayIdentity	Create a new ReplayIdentity.

Тор

Methods

	Name	Description
=♥	Equals(Object)	Override implementation. (Overrides ObjectEquals(Object).)
≡Ŵ	Equals(ReplayIdentity)	IEquateable implementation.
= <b>\$</b>	Generate	Generates a unique ReplayIdentity.
=0	GetHashCode	Override implementation. (Overrides ObjectGetHashCode.)
= <b>\$</b>	IsUnique(Int32)	Returns true if the specified id is unique or false if not.
= <b>≬ S</b>	IsUnique(ReplayIdentity)	Returns true if the specified ReplayIdentity is unique or false if not.
≡♥	ToString	Override implementation. (Overrides ObjectToString.)

### Тор

## ▲ Operators

	Name	Description
(∕_ =+) <b>S</b>	Equality	Override equals operator.
( <u>/-</u> =+) <b>S</b>	(Int16 to ReplayIdentity)	Implicit short conversion (16 bit identity only).
⊻ <u>-</u> =+) <b>S</b>	(ReplayIdentity to Int16)	Implicit int conversion (32 bit identity only).

S S	Inequality		Override not-equals operator.
Тор			
▲ Fields			
	Name		Description
Ŷ S	byteSize		Get the size in bytes of a ReplayIdentity representation.
<b>∛ S</b>	unassigned	Identity	Get the default value for a ReplayIdentity id which equates to an unassigned identity.
Тор			
▲ Prope	rties		
	Name	Descr	iption
	IsAssigned		s true if the identity has been ated or false if it has not.
Тор			
▲ See Also			
Reference UltimateReplay.Core Namespace			

# **ReplayIdentity Constructor**

Create a new ReplayIdentity.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

**C**#

public ReplayIdentity()

### ⊿ See Also

# **ReplayIdentity Fields**

The ReplayIdentity type exposes the following members.

## ▲ Fields

	Name	Description
Ŷ S	byteSize	Get the size in bytes of a ReplayIdentity representation.
<b>∛ S</b>	unassignedIdentity	Get the default value for a ReplayIdentity id which equates to an unassigned identity.

Тор

### ⊿ See Also

# ReplayIdentitybyteSize Field

Get the size in bytes of a ReplayIdentity representation.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Copy \_

public static readonly int byteSize

Field Value Type: Int32

### ⊿ See Also

## ReplayIdentityunassignedIdentity Field

Get the default value for a ReplayIdentity id which equates to an unassigned identity.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору \_

public const int unassignedIdentity

Field Value Type: Int32

### ⊿ See Also

# **ReplayIdentity Methods**

The ReplayIdentity type exposes the following members.

### Methods

	Name	Description
=∲	Equals(Object)	Override implementation. (Overrides ObjectEquals(Object).)
≡ <b>Q</b>	Equals(ReplayIdentity)	IEquateable implementation.
=≬	Generate	Generates a unique ReplayIdentity.
=♥	GetHashCode	Override implementation. (Overrides ObjectGetHashCode.)
= <b>0 S</b>	IsUnique(Int32)	Returns true if the specified id is unique or false if not.
= <b>≬ S</b>	IsUnique(ReplayIdentity)	Returns true if the specified ReplayIdentity is unique or false if not.
=♥	ToString	Override implementation. (Overrides ObjectToString.)
Тор		

⊿ See Also

# ReplayIdentityEquals Method

## Overload List

	Name	Description
-=•	Equals(Object)	Override implementation. (Overrides ObjectEquals(Object).)
=	Equals(ReplayIdentity)	IEquateable implementation.
Тор		
⊿ See	Also	
	ce entity Class eplay.Core Namespace	

# ReplayIdentityEquals Method (Object)

Override implementation.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

### ⊿ Syntax

**C**#

)

### Parameters

obj

Type: SystemObject The object to compare against

**Return Value** 

Type: Boolean [Missing <returns> documentation for "M:UltimateReplay.Core.ReplayIdentity.Equals(System.Object)"]

### ⊿ See Also

Reference ReplayIdentity Class Equals Overload UltimateReplay.Core Namespace

# ReplayIdentityEquals Method (ReplayIdentity)

IEquateable implementation.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Сору

### ▲ Syntax

**C**#

)

### Parameters

obj

Type: UltimateReplay.CoreReplayIdentity The ReplayIdentity to compare against

**Return Value** 

Type: Boolean [Missing <returns> documentation for "M:UltimateReplay.Core.ReplayIdentity.Equals(UltimateReplay.Core.ReplayIdentity)"]

Implements IEquatableTEquals(T)

## ▲ See Also

Reference ReplayIdentity Class Equals Overload UltimateReplay.Core Namespace

# ReplayIdentityGenerate Method

Generates a unique ReplayIdentity.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

C#

public void Generate()

### ⊿ See Also

## ReplayIdentityGetHashCode Method

Override implementation.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public override int GetHashCode()

#### **Return Value**

Type: Int32 [Missing <returns> documentation for "M:UltimateReplay.Core.ReplayIdentity.GetHashCode"]

### ⊿ See Also

# ReplayIdentityIsUnique Method

## Overload List

	Name	Description
= <b>0 S</b>	IsUnique(Int32)	Returns true if the specified id is unique or false if not.
= <b>≬ S</b>	IsUnique(ReplayIdentity)	Returns true if the specified ReplayIdentity is unique or false if not.

Тор

### ⊿ See Also

# ReplayIdentityIsUnique Method (Int32)

Returns true if the specified id is unique or false if not.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ⊿ Syntax

**C**#

```
public static bool IsUnique(
```

```
int id
```

)

### Parameters

id

Type: SystemInt32 The int id to check

**Return Value** 

Type: Boolean [Missing <returns> documentation for "M:UltimateReplay.Core.ReplayIdentity.IsUnique(System.Int32)"]

### ⊿ See Also

Reference ReplayIdentity Class IsUnique Overload UltimateReplay.Core Namespace

# ReplayIdentityIsUnique Method (ReplayIdentity)

Returns true if the specified ReplayIdentity is unique or false if not.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

## ⊿ Syntax

**C**#

)

### Parameters

id

Type: UltimateReplay.CoreReplayIdentity The identity to check

**Return Value** 

Type: Boolean [Missing <returns> documentation for "M:UltimateReplay.Core.ReplayIdentity.IsUnique(UltimateReplay.Core.ReplayIdentity)"]

### ⊿ See Also

Reference ReplayIdentity Class IsUnique Overload UltimateReplay.Core Namespace

# ReplayIdentityToString Method

Override implementation.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ⊿ Syntax

C#

Copy \_

public override string ToString()

**Return Value** 

Type: String [Missing <returns> documentation for "M:UltimateReplay.Core.ReplayIdentity.ToString"]

## ▲ See Also

# ReplayIdentity Operators and Type Conversions

The ReplayIdentity type exposes the following members.

## ▲ Operators

	Name	Description
<u>(∕-</u> =+) <b>S</b>	Equality	Override equals operator.
<sup>(∠</sup> = =+) <b>S</b>	(Int16 to ReplayIdentity)	Implicit short conversion (16 bit identity only).
( <u>/-</u> =+) <b>S</b>	(ReplayIdentity to Int16)	Implicit int conversion (32 bit identity only).
( <u>/-</u> =+) <b>S</b>	Inequality	Override not-equals operator.

Тор

### ⊿ See Also

# ReplayIdentityEquality Operator

Override equals operator.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

## ▲ Syntax

**C**#

### Parameters

а

Type: UltimateReplay.CoreReplayIdentity First ReplayIdentity

b

Type: UltimateReplay.CoreReplayIdentity Second ReplayIdentity

### **Return Value**

Type: Boolean [Missing <returns> documentation for "M:UltimateReplay.Core.ReplayIdentity.op\_Equality(UltimateReplay.Core.ReplayIdentity,UI

## ⊿ See Also

# ReplayIdentity Conversion Operators

### Overload List

	Name	Description
(/ <u>-</u> =+) <b>S</b>	(Int16 to ReplayIdentity)	Implicit short conversion (16 bit identity only).
(/- =+) <b>S</b>	(ReplayIdentity to Int16)	Implicit int conversion (32 bit identity only).

Тор

### ⊿ See Also

# ReplayIdentity Conversion (Int16 to ReplayIdentity)

Implicit short conversion (16 bit identity only).

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

## ⊿ Syntax

```
C#
```

Copy

)

### Parameters

*identity* Type: SystemInt16 The identity to convert

Return Value

Type: ReplayIdentity [Missing <returns> documentation for "M:UltimateReplay.Core.ReplayIdentity.op\_Implicit(System.Int16)~UltimateReplay.Core.Re

### ⊿ See Also

# ReplayIdentity Conversion (ReplayIdentity to Int16)

Implicit int conversion (32 bit identity only).

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

```
⊿ Syntax
```

**C**#

)

#### Parameters

identity

Type: UltimateReplay.CoreReplayIdentity The identify to convert

**Return Value** 

Type: Int16 [Missing <returns> documentation for "M:UltimateReplay.Core.ReplayIdentity.op\_Implicit(UltimateReplay.Core.ReplayIdentity)~S

### ⊿ See Also

Reference ReplayIdentity Class Overload UltimateReplay.Core Namespace

# ReplayIdentityInequality Operator

Override not-equals operator.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

#### Parameters

а

Type: UltimateReplay.CoreReplayIdentity First ReplayIdentity

b

Type: UltimateReplay.CoreReplayIdentity Second ReplayIdentity

#### **Return Value**

Type: Boolean [Missing <returns> documentation for "M:UltimateReplay.Core.ReplayIdentity.op\_Inequality(UltimateReplay.Core.ReplayIdentity,I

### ⊿ See Also

# **ReplayIdentity Properties**

The ReplayIdentity type exposes the following members.

### ▲ Properties

	Name	Description
	IsAssigned	Returns true if the identity has been generated or false if it has not.
Тор		
⊿ See Also		
Reference ReplayIdentity Class UltimateReplay.Core Namespace		

# ReplayIdentityIsAssigned Property

Returns true if the identity has been generated or false if it has not.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ⊿ Syntax

**C**#

public bool IsAssigned { get; }

Property Value Type: Boolean

### ⊿ See Also

# ReplayScene Class

A ReplayScene contains information about all active replay objects.

### ▲ Inheritance Hierarchy

SystemObject UltimateReplay.CoreReplayScene

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ⊿ Syntax

C# \_\_\_\_\_\_ Copy \_\_\_\_\_ Copy

The ReplayScene type exposes the following members.

### Constructors

	Name	Description
.≓ <b>©</b>	ReplayScene	Initializes a new instance of the ReplayScene class

#### Тор

### Methods

	Name	Description
≡Ŵ	RecordSnapshot	Take a snapshot of the current replay scene using the

		specified timestamp.
=	RegisterReplayObject	Registers a replay object with the replay system so that it can be recorded for playback. Typically all ReplayObject will auto register when they 'Awake' meaning that you will not need to manually register objects.
-=•	RestoreSnapshot	Restore the scene to the state described by the specified snapshot.
=♥	UnregisterReplayObject	Unregisters a replay object from the replay system so that it will no longer be recorded for playback. Typically all ReplayObject will auto un- register when they are destroyed so you will normally not need to un-register a replay object.

#### Тор

## ▲ Properties

	Name	Description
<b>*</b>	ActiveReplayObjects	Get a collection of all game objects that are registered with the replay system.
<b>*</b>	ReplayEnabled	Enable or disable the replay scene in preparation for playback or live mode. When true, all

replay objects will be prepared for playback causing certain components or scripts to be disabled to prevent interference from game systems. A prime candidate would be the RigidBody component which could cause a replay object to be affected by gravity and as a result deviate from its intended position. When false, all replay objects will be returned to their 'Live' state when all game systems will be reactivated.

#### Тор

### ⊿ See Also

Reference UltimateReplay.Core Namespace

# ReplayScene Constructor

Initializes a new instance of the ReplayScene class

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

C#

public ReplayScene()

### ⊿ See Also

# ReplayScene Methods

The ReplayScene type exposes the following members.

### Methods

	Name	Description
	INALLIC	Description
æ	RecordSnapshot	Take a snapshot of the current replay scene using the specified timestamp.
.≡♥	RegisterReplayObject	Registers a replay object with the replay system so that it can be recorded for playback. Typically all ReplayObject will auto register when they 'Awake' meaning that you will not need to manually register objects.
≡ <b>∲</b>	RestoreSnapshot	Restore the scene to the state described by the specified snapshot.
.≓♥	UnregisterReplayObject	Unregisters a replay object from the replay system so that it will no longer be recorded for playback. Typically all ReplayObject will auto un- register when they are destroyed so you will normally not need to un-register a replay object.

#### Тор

### ⊿ See Also

# ReplaySceneRecordSnapshot Method

Take a snapshot of the current replay scene using the specified timestamp.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public ReplaySnapshot RecordSnapshot(
 float timeStamp

)

#### Parameters

timeStamp

Type: SystemSingle

The timestamp for the frame indicating its position in the playback sequence

Return Value Type: ReplaySnapshot A new snapshot of the current replay scene

### ⊿ See Also

# ReplaySceneRegisterReplayObject Method

Registers a replay object with the replay system so that it can be recorded for playback. Typically all ReplayObject will auto register when they 'Awake' meaning that you will not need to manually register objects.

#### Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax



#### Parameters

replayObject

Type: UltimateReplayReplayObject The ReplayObject to register

### ⊿ See Also

# ReplaySceneRestoreSnapshot Method

Restore the scene to the state described by the specified snapshot.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ⊿ Syntax

**C**#

Сору

#### )

#### Parameters

snapshot

Type: UltimateReplay.StorageReplaySnapshot The snapshot to restore

### ▲ See Also

# ReplaySceneUnregisterReplayObjec Method

Unregisters a replay object from the replay system so that it will no longer be recorded for playback. Typically all ReplayObject will auto un-register when they are destroyed so you will normally not need to un-register a replay object.

#### Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax



#### Parameters

replayObject

Type: UltimateReplayReplayObject

### ⊿ See Also

# **ReplayScene Properties**

The ReplayScene type exposes the following members.

### ▲ Properties

	Name	Description
<b>*</b>	ActiveReplayObjects	Get a collection of all game objects that are registered with the replay system.
	ReplayEnabled	Enable or disable the replay scene in preparation for playback or live mode. When true, all replay objects will be prepared for playback causing certain components or scripts to be disabled to prevent interference from game systems. A prime candidate would be the RigidBody component which could cause a replay object to be affected by gravity and as a result deviate from its intended position. When false, all replay objects will be returned to their 'Live' state when all game systems will be reactivated.

Тор

### ⊿ See Also

# ReplaySceneActiveReplayObjects Property

Get a collection of all game objects that are registered with the replay system.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Copy \_

Property Value Type: HashSetReplayObject

### ⊿ See Also

# ReplaySceneReplayEnabled Property

Enable or disable the replay scene in preparation for playback or live mode. When true, all replay objects will be prepared for playback causing certain components or scripts to be disabled to prevent interference from game systems. A prime candidate would be the RigidBody component which could cause a replay object to be affected by gravity and as a result deviate from its intended position. When false, all replay objects will be returned to their 'Live' state when all game systems will be reactivated.

#### Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

public bool ReplayEnabled { get; set; }

Property Value Type: Boolean

### ⊿ See Also

# **ReplayVariable Class**

Represents a variable that can be recorded using the replay system in order to replay script animations or similar during playback.

### Inheritance Hierarchy

```
SystemObject UltimateReplay.CoreReplayVariable
```

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax



The ReplayVariable type exposes the following members.

### Constructors

	Name	Description
=	ReplayVariable	Create a new ReplayVariable.
Тор		
⊿ Meth	ods	
	Name	Description
= <b>≬ S</b>	CanInterpolate	Returns true if the

		specified type can be interpolated by the replay system.
≡ <b>⊘</b>	Interpolate	Attempts to interpolate the ReplayVariable value using the values from the last and next frame.
<b>≓</b> ≬ S	InterpolateByte	Default interpolator for byte.
<b>≡</b> ∳ S	InterpolateColor	Default interpolator for Color.
≓ <b>≬ S</b>	InterpolateColor32	Default interpolator for Color32.
≓ <b>≬ S</b>	InterpolateDouble	Default interpolator for double.
<b>≓</b> ≬ S	InterpolateFloat	Default interpolator for float.
≓ <b>≬ S</b>	InterpolateInt	Default interpolator for int.
≓ <b>≬ S</b>	InterpolateLong	Default interpolator for long.
≓ <b>≬ S</b>	InterpolateQuat	Default interpolator for Quaternion.
= <b>≬ S</b>	InterpolateShort	Default interpolator for short.
= <b>≬ S</b>	InterpolateValue	Attempts to interpolate the ReplayVariable value

		using the values from the last and next frame. In order for interpolation to succeed, the last and next values must be of the same type.
= <b>\$</b> S	InterpolateVec2	Default interpolator for Vector2.
= <b>\$</b>	InterpolateVec3	Default interpolator for Vector3.
= <b>\$</b>	InterpolateVec4	Default interpolator for Vector4.
= <b>0</b>	OnReplayDeserialize	Called by the replay system when the variable should be deserialized.
= <b>0</b>	OnReplaySerialize	Called by the replay system when the variable should be serialized.
≓ <b>≬ S</b>	RegisterCustomInterpolatorT	Allows a custom interpolation method to be registered so that unsupported variable types can be interpolated automatically.
<b>≓</b> ∳	UpdateValueRange	Sets the current interpolation range for the ReplayVariable value.

### <sup>Top</sup> ▲ Properties

	Name	Description
	Attribute	Get the ReplayVarAttribute associated with this ReplayVariable.
<b>*</b>	gameObject	Get the game object that this ReplayVariable is attached to.
<b>*</b>	IsInterpolated	Returns true if this ReplayVariable should be interpolated between frames.
	IsInterpolationSupported	Returns true if this ReplayVariable supports interpolation. Interpolation can only be supported if the variable type has a registered interpolator.
<b>*</b>	Name	Get the name of this ReplayVariable.
<b>*</b>	Value	The current value for this ReplayVariable.

Тор

### ⊿ See Also

Reference UltimateReplay.Core Namespace

# **ReplayVariable Constructor**

Create a new ReplayVariable.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

Сору

#### )

**C**#

#### Parameters

owner

Type: UltimateReplayReplayBehaviour

The ReplayBehaviour that this ReplayVariable is defined in

field

Type: System.ReflectionFieldInfo The field info for the variable field

attribute

Type: UltimateReplayReplayVarAttribute The ReplayVarAttribute for the field

### ⊿ See Also

# **ReplayVariable Methods**

The ReplayVariable type exposes the following members.

### Methods

	Name	Description
≓ <b>ù S</b>	CanInterpolate	Returns true if the specified type can be interpolated by the replay system.
<b>.</b> ≡ <b>\$</b>	Interpolate	Attempts to interpolate the ReplayVariable value using the values from the last and next frame.
≡ŷ S	InterpolateByte	Default interpolator for byte.
≡ <b>≬</b> S	InterpolateColor	Default interpolator for Color.
<b>≋</b> ≬ S	InterpolateColor32	Default interpolator for Color32.
≡¥ S	InterpolateDouble	Default interpolator for double.
≡Ŵ S	InterpolateFloat	Default interpolator for float.
≡ŷ S	InterpolateInt	Default interpolator for int.

≓ <b>≬ S</b>	InterpolateLong	Default interpolator for long.
≓ <b>≬ S</b>	InterpolateQuat	Default interpolator for Quaternion.
≓ <b>≬ S</b>	InterpolateShort	Default interpolator for short.
=\$ S	InterpolateValue	Attempts to interpolate the ReplayVariable value using the values from the last and next frame. In order for interpolation to succeed, the last and next values must be of the same type.
= <b>0 S</b>	InterpolateVec2	Default interpolator for Vector2.
≓ <b>≬ S</b>	InterpolateVec3	Default interpolator for Vector3.
= <b>0 S</b>	InterpolateVec4	Default interpolator for Vector4.
=∳	OnReplayDeserialize	Called by the replay system when the variable should be deserialized.
=♥	OnReplaySerialize	Called by the replay system when the variable should be serialized.

= <b>0 S</b>	RegisterCustomInterpolatorT	Allows a custom interpolation method to be registered so that unsupported variable types can be interpolated automatically.
<b>=∲</b>	UpdateValueRange	Sets the current interpolation range for the ReplayVariable value.

#### Тор

⊿ See Also

# ReplayVariableCanInterpolate Method

Returns true if the specified type can be interpolated by the replay system.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

```
public static bool CanInterpolate(
    Type type
```

)

#### Parameters

type

Type: SystemType

The system type to check for interpolation support

Return Value Type: Boolean

True if interpolation is supported or faluse if it is not

### ⊿ See Also

# ReplayVariableInterpolate Method

Attempts to interpolate the ReplayVariable value using the values from the last and next frame.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax



#### Parameters

delta

Type: SystemSingle

The normalized delta representing the progression from the last frame to the next frame

### ⊿ See Also

# ReplayVariableInterpolateByte Method

Default interpolator for byte.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

### ⊿ Syntax

Parameters

last

Type: SystemObject Last value

next

Type: SystemObject Next value

delta

Type: SystemSingle Interpolation delta

Return Value Type: Object The interpolated byte value

### ⊿ See Also

#### Reference

# ReplayVariableInterpolateColor Method

Default interpolator for Color.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

### ▲ Syntax

```
c#
public static Object InterpolateColor(
        Object last,
        Object next,
        float delta
)
```

Parameters

last

Type: SystemObject Last value

next

Type: SystemObject Next value

delta

Type: SystemSingle Interpolation delta

Return Value Type: Object The interpolated Color value

#### Reference

## ReplayVariableInterpolateColor32 Method

Default interpolator for Color32.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

### ⊿ Syntax

Parameters

last

Type: SystemObject Last value

next

Type: SystemObject Next value

delta

Type: SystemSingle Interpolation delta

Return Value Type: Object The interpolated Color32 value

#### Reference

# ReplayVariableInterpolateDouble Method

Default interpolator for double.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

### ⊿ Syntax

Parameters

last

Type: SystemObject Last value

next

Type: SystemObject Next value

delta

Type: SystemSingle Interpolation delta

Return Value Type: Object The interpolated double value

#### Reference

# ReplayVariableInterpolateFloat Method

Default interpolator for float.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

### ▲ Syntax

**C**#

Parameters

last

Type: SystemObject Last value

next

Type: SystemObject Next value

delta

Type: SystemSingle Interpolation delta

Return Value Type: Object

The interpolated float value

#### Reference

# **ReplayVariableInterpolateInt** Method

Default interpolator for int.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

### ▲ Syntax

**C**#

```
public static Object InterpolateInt(
        Object last,
        Object next,
        float delta
```

**Parameters** 

last

)

Type: SystemObject Last value

next

Type: SystemObject Next value

delta

Type: SystemSingle Interpolation delta

**Return Value** 

Type: Object The interpolated int value

#### Reference

# ReplayVariableInterpolateLong Method

Default interpolator for long.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

### ▲ Syntax

Parameters

last

Type: SystemObject Last value

next

Type: SystemObject Next value

delta

Type: SystemSingle Interpolation delta

Return Value Type: Object

The interpolated long value

#### Reference

# ReplayVariableInterpolateQuat Method

Default interpolator for Quaternion.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

### ▲ Syntax

Parameters

last

Type: SystemObject Last value

next

Type: SystemObject Next value

delta

Type: SystemSingle Interpolation delta

**Return Value** 

Type: Object

The interpolated Quaternion value

#### Reference

# ReplayVariableInterpolateShort Method

Default interpolator for short.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

### ▲ Syntax

**C**#

Parameters

last

Type: SystemObject Last value

next

Type: SystemObject Next value

delta

Type: SystemSingle Interpolation delta

Return Value Type: Object The interpolated short value

#### Reference

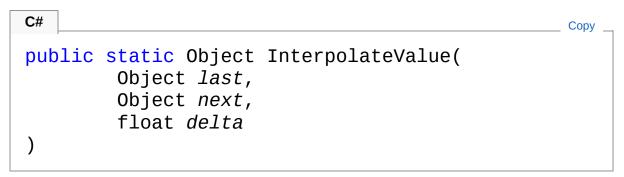
# ReplayVariableInterpolateValue Method

Attempts to interpolate the ReplayVariable value using the values from the last and next frame. In order for interpolation to succeed, the last and next values must be of the same type.

#### Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax



Parameters

last

Type: SystemObject

The value of the variable in the last frame

next

Type: SystemObject

The value of the variable in the next frame

delta

Type: SystemSingle

The normalized delta representing the progression from the last frame to the next frame

**Return Value** 

#### Type: Object

The interpolated value result or null if interpolation is not supported for the type

### ⊿ See Also

# ReplayVariableInterpolateVec2 Method

Default interpolator for Vector2.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

### ▲ Syntax

c#
public static Object InterpolateVec2(
 Object last,
 Object next,
 float delta
)

Parameters

last

Type: SystemObject Last value

next

Type: SystemObject Next value

delta

Type: SystemSingle Interpolation delta

Return Value Type: Object The interpolated Vector2 value

#### Reference

# ReplayVariableInterpolateVec3 Method

Default interpolator for Vector3.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

### ▲ Syntax

Parameters

last

Type: SystemObject Last value

next

Type: SystemObject Next value

delta

Type: SystemSingle Interpolation delta

Return Value Type: Object The interpolated Vector3 value

#### Reference

# ReplayVariableInterpolateVec4 Method

Default interpolator for Vector4.

```
Namespace: UltimateReplay.Core
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

### ▲ Syntax

Parameters

last

Type: SystemObject Last value

next

Type: SystemObject Next value

delta

Type: SystemSingle Interpolation delta

Return Value Type: Object The interpolated Vector4 value

#### Reference

# ReplayVariableOnReplayDeserialize Method

Called by the replay system when the variable should be deserialized.

Сору

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ⊿ Syntax

C#

)

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to deserialize the data from

Implements IReplaySerializeOnReplayDeserialize(ReplayState)

### ⊿ See Also

# ReplayVariableOnReplaySerialize Method

Called by the replay system when the variable should be serialized.

Сору

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ⊿ Syntax

**C**#

)

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to serialize the data into

Implements IReplaySerializeOnReplaySerialize(ReplayState)

### ⊿ See Also

# ReplayVariableRegisterCustomInter Method

Allows a custom interpolation method to be registered so that unsupported variable types can be interpolated automatically.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

```
C#

public static void RegisterCustomInterpolator<T>(

Func<Object, Object, float, Object> interpolator

)
```

#### Parameters

*interpolatorFunc* 

Type: SystemFuncObject, Object, Single, Object

The interpolation method to invoke when interpolation of the custom type is required

#### Type Parameters

Т

The type of varaible that the custom interpolation should be used for

### ⊿ See Also

# ReplayVariableUpdateValueRange Method

Sets the current interpolation range for the ReplayVariable value.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Сору

### ⊿ Syntax

**C**#

#### Parameters

last

Type: SystemObject

The value of the variable in the last frame

next

Type: SystemObject The value of the variable in the next frame

### ▲ See Also

## **ReplayVariable Properties**

The ReplayVariable type exposes the following members.

### ▲ Properties

	Name	Description
	Attribute	Get the ReplayVarAttribute associated with this ReplayVariable.
	gameObject	Get the game object that this ReplayVariable is attached to.
<b>*</b>	IsInterpolated	Returns true if this ReplayVariable should be interpolated between frames.
	IsInterpolationSupported	Returns true if this ReplayVariable supports interpolation. Interpolation can only be supported if the variable type has a registered interpolator.
	Name	Get the name of this ReplayVariable.
<b>*</b>	Value	The current value for this ReplayVariable.
Тор		

#### Reference

# ReplayVariableAttribute Property

Get the ReplayVarAttribute associated with this ReplayVariable.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

public ReplayVarAttribute Attribute { get; }

Сору

Property Value Type: ReplayVarAttribute

### ⊿ See Also

# ReplayVariablegameObject Property

Get the game object that this ReplayVariable is attached to.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ⊿ Syntax

**C**#

Сору

```
public GameObject gameObject { get; }
```

Property Value Type: GameObject

### ⊿ See Also

# ReplayVariableIsInterpolated Property

Returns true if this ReplayVariable should be interpolated between frames.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Copy \_

public bool IsInterpolated { get; }

Property Value Type: Boolean

#### ⊿ See Also

# ReplayVariableIsInterpolationSuppor Property

Returns true if this ReplayVariable supports interpolation. Interpolation can only be supported if the variable type has a registered interpolator.

Namespace: UltimateReplay.Core

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

public bool IsInterpolationSupported { get; }

Copy \_

Property Value Type: Boolean

### ⊿ See Also

# ReplayVariableName Property

Get the name of this ReplayVariable.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

**C**#

public string Name { get; }

Property Value Type: String

### ⊿ See Also

# ReplayVariableValue Property

The current value for this ReplayVariable.

Namespace: UltimateReplay.Core Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

C#

public Object Value { get; set; }

Property Value Type: Object

#### ⊿ See Also

## UltimateReplay.Demo Namespace

## ▲ Classes

	Class	Description
<b>*</b> \$	AudioDemo	A script using in the audio demo to show the ReplayAudio component working.
*\$	CubeSpawner	A demo script used in the stress test scene which spawns a large number of cubes.

# AudioDemo Class

A script using in the audio demo to show the ReplayAudio component working.

### ⊿ Inheritance Hierarchy

SystemObject Object Component Behaviour MonoBehaviour UltimateReplay.DemoAudioDemo

Namespace: UltimateReplay.Demo Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

public class AudioDemo : MonoBehaviour

The AudioDemo type exposes the following members.

### Constructors

	Name	Description
. <b>≕</b> ∲	AudioDemo	Initializes a new instance of the AudioDemo class
Тор		

## ▲ Methods

	Name	Description
= <b>Q</b>	OnGUI	Called by Unity.
= <b>Q</b>	Update	Called by Unity.
Тор		
▲ Fields		
	Name	Description
0	replayAudio	The ReplayAudio script.
Тор		
⊿ See Also		
Reference UltimateReplay.Demo Namespace		

## AudioDemo Constructor

Initializes a new instance of the AudioDemo class

Namespace: UltimateReplay.Demo Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

**C**#

public AudioDemo()

### ⊿ See Also

# AudioDemo Fields

The AudioDemo type exposes the following members.

⊿ Fields		
	Name	Description
۵	replayAudio	The ReplayAudio script.
Тор		
⊿ See A	lso	
Reference AudioDemo UltimateRep		ace

# AudioDemoreplayAudio Field

The ReplayAudio script.

Namespace: UltimateReplay.Demo Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

C#

Copy \_

public ReplayAudio replayAudio

Field Value Type: ReplayAudio

### ⊿ See Also

# AudioDemo Methods

The AudioDemo type exposes the following members.

## ▲ Methods

	Name	Description
.≡ <b>Q</b>	OnGUI	Called by Unity.
≡ <b>≬</b>	Update	Called by Unity.

Тор

### ⊿ See Also

## AudioDemoOnGUI Method

Called by Unity.

Namespace: UltimateReplay.Demo Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

**C**#

public void OnGUI()

### ⊿ See Also

## AudioDemoUpdate Method

Called by Unity.

Namespace: UltimateReplay.Demo Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

C#

public void Update()

### ⊿ See Also

# CubeSpawner Class

A demo script used in the stress test scene which spawns a large number of cubes.

### ▲ Inheritance Hierarchy

SystemObject Object Component Behaviour MonoBehaviour UltimateReplay.DemoCubeSpawner

Namespace: UltimateReplay.Demo Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

Copy \_\_\_\_\_\_\_\_ Copy \_\_\_\_\_\_\_\_ Copy \_\_\_\_\_\_\_

The CubeSpawner type exposes the following members.

### Constructors

	Name	Description
.≓ <b>©</b>	CubeSpawner	Initializes a new instance of the CubeSpawner class
Top		

Тор

## ▲ Methods

	Name	Description
<b>≓∲</b>	Start	Called by Unity.
Тор		
▲ Fields		
	Name	Description
Ŷ	explosiveForce	The amount of force that is initially given to the spawning cubes.
Ŷ	spawnAmount	The amount of objects to spawn into the scene.
Ŷ	spawnCubes	An array of prefabs used to randomly spawn objects.
٥	spawnHeight	The maximum height that an object can be spawned from the center.
٥	spawnRange	The maximum distance that an object can be spawned from the center.
•	spawnRange	The maximum distance that an object

Тор

⊿ See Also

Reference UltimateReplay.Demo Namespace

# **CubeSpawner Constructor**

Initializes a new instance of the CubeSpawner class

Namespace: UltimateReplay.Demo Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

C#

public CubeSpawner()

### ⊿ See Also

# CubeSpawner Fields

The CubeSpawner type exposes the following members.

### ▲ Fields

	Name	Description
٥	explosiveForce	The amount of force that is initially given to the spawning cubes.
٥	spawnAmount	The amount of objects to spawn into the scene.
Ŷ	spawnCubes	An array of prefabs used to randomly spawn objects.
Ŷ	spawnHeight	The maximum height that an object can be spawned from the center.
٥	spawnRange	The maximum distance that an object can be spawned from the center.

Тор

### ⊿ See Also

# CubeSpawnerexplosiveForce Field

The amount of force that is initially given to the spawning cubes.

Namespace: UltimateReplay.Demo Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

C#

public float explosiveForce

Field Value Type: Single

### ⊿ See Also

# CubeSpawnerspawnAmount Field

The amount of objects to spawn into the scene.

Namespace: UltimateReplay.Demo Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

**C**#

public int spawnAmount

Field Value Type: Int32

### ⊿ See Also

# CubeSpawnerspawnCubes Field

An array of prefabs used to randomly spawn objects.

Namespace: UltimateReplay.Demo Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

**C**#

public GameObject[] spawnCubes

Field Value Type: **GameObject** 

### ⊿ See Also

# CubeSpawnerspawnHeight Field

The maximum height that an object can be spawned from the center.

Namespace: UltimateReplay.Demo Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ⊿ Syntax

**C**#

public float spawnHeight

Field Value Type: Single

### ⊿ See Also

# CubeSpawnerspawnRange Field

The maximum distance that an object can be spawned from the center.

Copy \_

Namespace: UltimateReplay.Demo Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

public float spawnRange

Field Value Type: Single

### ⊿ See Also

# CubeSpawner Methods

The CubeSpawner type exposes the following members.

▲ Methods			
	Name	Description	
÷	Start	Called by Unity.	
Тор			
⊿ See	Also		
	<b>ice</b> awner Class Replay.Demo Nam	espace	

## CubeSpawnerStart Method

Called by Unity.

Namespace: UltimateReplay.Demo Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

public IEnumerator Start()

**Return Value** 

Type: IEnumerator [Missing <returns> documentation for "M:UltimateReplay.Demo.CubeSpawner.Start"]

Copy \_

## ⊿ See Also

# UltimateReplay.Storage Namespace

### ▲ Classes

	Class	Description
<b>*</b> \$	Compression	Compression utility using the GZip compression algorithm.
*\$	ReplayMemoryTarget	Represents a memory storage buffer where replay data can be stored for game sessions. The buffer can be used as a continuous rolling buffer of a fixed size where a fixed amount of playback footage is recorded and then overwritten by new data as it is received.
*\$	ReplaySnapshot	A frame state is a snapshot of a replay frame that is indexed based on its time stamp. By sequencing multiple frame states you can create the replay effect.
<b>~</b> ţ	ReplayTarget	Represents and abstract storage device capable of holding recorded state data for playback at a later date. Depending upon implementation, a ReplayTarget

## ▲ Structures

Structure	Description
\$ ReplayInitialData	Represents the intial settings of a newly spawned replay object. When a game object is instantiated it must be given an initial position and rotation.

## Enumerations

	Enumeration	Description
3 <sup>20</sup>	CompressionLevel	The amount of compression to apply to a data stream.
3 <sup>9</sup>	ReplayInitialDataFlags	Represents initial data that may be stored by an object.
38	ReplayTargetTask	Represents a task that can be issued to a ReplayTarget.

# **Compression Class**

Compression utility using the GZip compression algorithm.

## Inheritance Hierarchy

SystemObject UltimateReplay.StorageCompression

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

C# \_\_\_\_\_\_\_ Copy \_\_\_\_\_\_ public static class Compression

The Compression type exposes the following members.

### ▲ Methods

	Name	Description
= <b>≬ S</b>	CompressData	Compress a data stream using the GZip compression algorithm.
≓ <b>≬ S</b>	DecompressData	Decompress a data stream using the GZip compression algorithm.

Тор

⊿ See Also

Reference

UltimateReplay.Storage Namespace

# **Compression Methods**

The Compression type exposes the following members.

### ▲ Methods

	Name	Description
≓ <b>≬ S</b>	CompressData	Compress a data stream using the GZip compression algorithm.
<b>≓≬ S</b>	DecompressData	Decompress a data stream using the GZip compression algorithm.

Тор

### ⊿ See Also

Reference Compression Class UltimateReplay.Storage Namespace

# CompressionCompressData Method

Compress a data stream using the GZip compression algorithm.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ⊿ Syntax

### Parameters

data Type: SystemByte The input data to compress

### level (Optional)

Type: UltimateReplay.StorageCompressionLevel The target compression level to use

Return Value Type: Byte The compressed data

### ⊿ See Also

Reference

Compression Class UltimateReplay.Storage Namespace

# CompressionDecompressData Method

Decompress a data stream using the GZip compression algorithm.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

### Parameters

data Type: SystemByte

The input data to decompress

### level (Optional)

Type: UltimateReplay.StorageCompressionLevel The target compression level to use

Return Value Type: Byte The decompressed data

### ⊿ See Also

Reference

Compression Class UltimateReplay.Storage Namespace

## **CompressionLevel Enumeration**

The amount of compression to apply to a data stream.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

C#

public enum CompressionLevel

### Members

	lember ame	Value	Description
N	one	0	No compression is applied and all data is left unchanged.
0	ptimal	1	All data is compressed to the optimal level.

## ⊿ See Also

Reference UltimateReplay.Storage Namespace

# **ReplayInitialData Structure**

Represents the intial settings of a newly spawned replay object. When a game object is instantiated it must be given an initial position and rotation.

Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ⊿ Syntax

**C**#

public struct ReplayInitialData

The ReplayInitialData type exposes the following members.

### ▲ Fields

	Name	Description
٩	position	Initial position data.
٩	rotation	Initial rotation data.
٩	scale	Initial scale data.

Тор

### ⊿ See Also

Reference UltimateReplay.Storage Namespace

# ReplayInitialData Fields

The ReplayInitialData type exposes the following members.

### ▲ Fields

	Name	Description
٢	position	Initial position data.
۵	rotation	Initial rotation data.
۵	scale	Initial scale data.

Тор

### ⊿ See Also

# **ReplayInitialDataposition Field**

Initial position data.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

C#

public Vector3 position

Field Value Type: Vector3

### ⊿ See Also

# **ReplayInitialDatarotation Field**

Initial rotation data.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

C#

Copy \_

public Quaternion rotation

Field Value Type: **Quaternion** 

### ⊿ See Also

## ReplayInitialDatascale Field

Initial scale data.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Copy \_

public Vector3 scale

Field Value Type: Vector3

### ⊿ See Also

## ReplayInitialDataFlags Enumeration

Represents initial data that may be stored by an object.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ⊿ Syntax

**C**#

\_\_\_ Сору

[FlagsAttribute]
public enum ReplayInitialDataFlags

### Members

Member name	e Value	Description
None	0	No initial data is stored.
Position	1	Initial position is recorded.
Rotation	2	Initial rotation is recorded.
Scale	4	Initial scale is recorded.

## ▲ See Also

Reference UltimateReplay.Storage Namespace

# ReplayMemoryTarget Class

Represents a memory storage buffer where replay data can be stored for game sessions. The buffer can be used as a continuous rolling buffer of a fixed size where a fixed amount of playback footage is recorded and then overwritten by new data as it is received.

### ▲ Inheritance Hierarchy

#### SystemObject Object

Component Behaviour MonoBehaviour UltimateReplayReplayBehaviour UltimateReplay.StorageReplayTarget UltimateReplay.StorageReplayMemoryTarget

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

=Q

```
[SerializableAttribute]
public class ReplayMemoryTarget : ReplayTarget
```

The ReplayMemoryTarget type exposes the following members.

### Constructors

Name

Description

Copy

ReplayMemoryTarget Initializes a new instance of the

#### Тор

### Methods

	Name	Description
≓	PrepareTarget	Clears all state information for the current recording essentially restoring the memory to its initial state. (Overrides ReplayTargetPrepareTarget(ReplayTarget <sup>-</sup>
. <b>≕</b>	RecordSnapshot	Store a replay snapshot in the replay targe new snapshot causes the internal buffer to 'overflow' then the recoding clip will be wra so that the recording duration is no more the recordSeconds. (Overrides ReplayTargetRecordSnapshot(ReplaySna
<b>≡©</b>	RestoreSnapshot	Recall a snapshot from the replay target b on the specified replay offset. (Overrides ReplayTargetRestoreSnapshot(Single).)

Тор

## **▲** Fields

	Name	Description
۵	recordSeconds	The amount of time in seconds of recording that should be kept in memory before it is discarded. The time is measured backwards from the current time to give a rolling buffer of

the last 'n' seconds. This is useful in situations where you are only need the previous few secnonds of gameplay to be recorded for example: A kill cam. If this value is set to 0 then the internal buffer will not be wrapped at all. You should take extra care when using an unconstrained buffer as there is potential to run into an OutOfMemoryException, especially on mobile platforms where memory is at a premium.

#### Тор

### ▲ Properties

Name	Description
Duration	The amount of time in seconds that the recording lasts. Usually this value will be equal to recordSeconds however it will take atleast the amount of recordSeconds to initially fill the buffer before it wraps around. (Overrides ReplayTargetDuration.)
MemorySize	Get the amount of size in bytes that this memory target requires for all state data. This size does not include internal structures used to store the data but exclusivley contains game state sizes. (Overrides ReplayTargetMemorySize.)

Тор

⊿ See Also

#### Reference UltimateReplay.Storage Namespace

# ReplayMemoryTarget Constructor

Initializes a new instance of the ReplayMemoryTarget class

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

### ▲ Syntax

**C**#

public ReplayMemoryTarget()

#### ⊿ See Also

## ReplayMemoryTarget Fields

The ReplayMemoryTarget type exposes the following members.

### ▲ Fields

Name	Description
recordSeconds	The amount of time in seconds of recording that should be kept in memory before it is discarded. The time is measured backwards from the current time to give a rolling buffer of the last 'n' seconds. This is useful in situations where you are only need the previous few secnonds of gameplay to be recorded for example: A kill cam. If this value is set to 0 then the internal buffer will not be wrapped at all. You should take extra care when using an unconstrained buffer as there is potential to run into an OutOfMemoryException, especially on mobile platforms where memory is at a premium.

Тор

#### ⊿ See Also

## ReplayMemoryTargetrecordSeconds Field

The amount of time in seconds of recording that should be kept in memory before it is discarded. The time is measured backwards from the current time to give a rolling buffer of the last 'n' seconds. This is useful in situations where you are only need the previous few secnonds of gameplay to be recorded for example: A kill cam. If this value is set to 0 then the internal buffer will not be wrapped at all. You should take extra care when using an unconstrained buffer as there is potential to run into an OutOfMemoryException, especially on mobile platforms where memory is at a premium.

#### Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public float recordSeconds

Field Value Type: Single

#### ⊿ See Also

## ReplayMemoryTarget Methods

The ReplayMemoryTarget type exposes the following members.

### Methods

	Name	Description
=∳	PrepareTarget	Clears all state information for the current recording essentially restoring the memory to its initial state. (Overrides ReplayTargetPrepareTarget(ReplayTarget
= <b>\$</b>	RecordSnapshot	Store a replay snapshot in the replay targe new snapshot causes the internal buffer to 'overflow' then the recoding clip will be wra so that the recording duration is no more the recordSeconds. (Overrides ReplayTargetRecordSnapshot(ReplaySna
<b>≓</b> \$	RestoreSnapshot	Recall a snapshot from the replay target b on the specified replay offset. (Overrides ReplayTargetRestoreSnapshot(Single).)

#### Тор

### ⊿ See Also

## ReplayMemoryTargetPrepareTarget Method

Clears all state information for the current recording essentially restoring the memory target to its initial state.

Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

)

#### Parameters

mode

Type: UltimateReplay.StorageReplayTargetTask [Missing <param name="mode"/> documentation for "M:UltimateReplay.Storage.ReplayMemoryTarget.PrepareTarget(UltimateReplay.Storage

Copy

#### ⊿ See Also

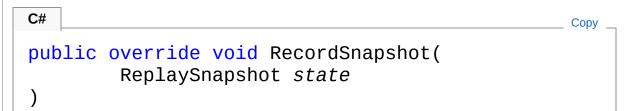
## ReplayMemoryTargetRecordSnapsh Method

Store a replay snapshot in the replay target. If the new snapshot causes the internal buffer to 'overflow' then the recoding clip will be wrapped so that the recording duration is no more than recordSeconds.

#### Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax



#### Parameters

state

Type: UltimateReplay.StorageReplaySnapshot The snapshot to store

### ⊿ See Also

## ReplayMemoryTargetRestoreSnapsl Method

Recall a snapshot from the replay target based on the specified replay offset.

Namespace: UltimateReplay.Storage

```
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)
```

### ▲ Syntax

**C**#

```
public override ReplaySnapshot RestoreSnapshot(
    float offset
```

Copy

)

#### Parameters

offset

Type: SystemSingle

The offset pointing to the individual snapshot to recall

Return Value Type: ReplaySnapshot The replay snapshot at the specified offset

### ⊿ See Also

# **ReplayMemoryTarget Properties**

The ReplayMemoryTarget type exposes the following members.

## ▲ Properties

Name	Description
Duration	The amount of time in seconds that the recording lasts. Usually this value will be equal to recordSeconds however it will take atleast the amount of recordSeconds to initially fill the buffer before it wraps around. (Overrides ReplayTargetDuration.)
MemorySize	Get the amount of size in bytes that this memory target requires for all state data. This size does not include internal structures used to store the data but exclusivley contains game state sizes. (Overrides ReplayTargetMemorySize.)

Тор

### ⊿ See Also

## ReplayMemoryTargetDuration Property

The amount of time in seconds that the recording lasts. Usually this value will be equal to recordSeconds however it will take atleast the amount of recordSeconds to initially fill the buffer before it wraps around.

#### Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

**C**#

\_

Copy

public override float Duration { get; }

Property Value Type: Single

#### ⊿ See Also

## ReplayMemoryTargetMemorySize Property

Get the amount of size in bytes that this memory target requires for all state data. This size does not include internal structures used to store the data but exclusivley contains game state sizes.

#### Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)



**C**#

Сору

public override int MemorySize { get; }

Property Value Type: Int32

#### ⊿ See Also

## ReplaySnapshot Class

A frame state is a snapshot of a replay frame that is indexed based on its time stamp. By sequencing multiple frame states you can create the replay effect.

### Inheritance Hierarchy

SystemObject UltimateReplay.StorageReplaySnapshot

Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

▲ Syntax

**C**#

```
C#
[SerializableAttribute]
public sealed class ReplaySnapshot : IReplaySerial.
        IReplayDataSerialize
```

The ReplaySnapshot type exposes the following members.

### Constructors

	Name	Description
=♥	ReplaySnapshot	Create a new snapshot with the specified time stamp.

Тор

Methods

	Name	Description
≡♥	OnReplayDataDeserialize	Called by the replay system when this ReplaySnapshot should be deserialized from binary.
<b>≡\$</b>	OnReplayDataSerialize	Called by the replay system when this ReplaySnapshot should be serialized to binary.
<b>≡</b> ©	OnReplayDeserialize	Called by the replay system when this ReplaySnapshot should be deserialized.
<b>∃</b> ∲	OnReplaySerialize	Called by the replay system when this ReplaySnapshot should be serialized.
≓\$	RecordInitialReplayObjectData	Attempts to record the initial information about a newly created replay object. The initial information will only be sroted if the data is not equal to the default value.
<b>≞</b> ©	RecordSnapshot	Registers the specified replay state with this snapshot. The specified identity

		is used during playback to ensure that the replay objects receives the correct state to deserialize.
≓Ŵ	Reset	Clears all state information from the snapshot but keeps the time stamp.
	RestoreInitialReplayObjectData	Attempts to get the initial data for the replay object with the specified ReplayIdentity. A ReplayInitialData will be returned containing the initial state for the replay object.
<b>≓</b> ©	RestoreReplayObjects	Attempts to restore any replay objects that were spawned or despawned during this snapshot.
. <b>≕</b>	RestoreSnapshot	Attempts to recall the state information for the specified replay object identity. If the identity does not exist in the scene then the return value will be null.

Тор

## ▲ Properties

	Name	Description
<b>≧</b>	Size	Get the size in bytes of the snapshot data.
<b>*</b>	TimeStamp	The time stamp for this snapshot. The time stamp is used to identify the snapshot location in the sequence.

Тор

#### ⊿ See Also

Reference UltimateReplay.Storage Namespace

## ReplaySnapshot Constructor

Create a new snapshot with the specified time stamp.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

public ReplaySnapshot(
 float timeStamp

)

#### Parameters

*timeStamp* Type: SystemSingle The time stamp to give to this snapshot

### ⊿ See Also

# **ReplaySnapshot Methods**

The ReplaySnapshot type exposes the following members.

### Methods

	Name	Description
. <b>≕</b>	OnReplayDataDeserialize	Called by the replay system when this ReplaySnapshot should be deserialized from binary.
.≓	OnReplayDataSerialize	Called by the replay system when this ReplaySnapshot should be serialized to binary.
<b>≓</b>	OnReplayDeserialize	Called by the replay system when this ReplaySnapshot should be deserialized.
≡ <b>⊘</b>	OnReplaySerialize	Called by the replay system when this ReplaySnapshot should be serialized.
<b>≞</b> ©	RecordInitialReplayObjectData	Attempts to record the initial information about a newly created

		replay object. The initial information will only be sroted if the data is not equal to the default value.
.≓	RecordSnapshot	Registers the specified replay state with this snapshot. The specified identity is used during playback to ensure that the replay objects receives the correct state to deserialize.
. <b>≕</b>	Reset	Clears all state information from the snapshot but keeps the time stamp.
<b>.</b> ≡ <b>`</b>	RestoreInitialReplayObjectData	Attempts to get the initial data for the replay object with the specified ReplayIdentity. A ReplayInitialData will be returned containing the initial state for the replay object.
<b>≓</b> \$	RestoreReplayObjects	Attempts to restore any replay objects that were spawned or despawned during this snapshot.

RestoreSnapshot	Attempts to recall the state information for the specified replay object identity. If the identity does not exist in the scene then the return value will be null.
-----------------	---

#### Тор

### ⊿ See Also

Reference

## ReplaySnapshotOnReplayDataDese Method

Called by the replay system when this ReplaySnapshot should be deserialized from binary.

Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

public void OnReplayDataDeserialize(
 BinaryReader reader

)

#### Parameters

reader

Type: System.IOBinaryReader The binary stream to read the data from

#### ⊿ See Also

## ReplaySnapshotOnReplayDataSeria Method

Called by the replay system when this ReplaySnapshot should be serialized to binary.

Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

#### ▲ Syntax

**C**#

```
public void OnReplayDataSerialize(
    BinaryWriter writer
```

)

#### Parameters

writer

Type: System.IOBinaryWriter The binary stream to write te data to

#### ⊿ See Also

## ReplaySnapshotOnReplayDeserializ Method

Called by the replay system when this ReplaySnapshot should be deserialized.

Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

)

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to read the data from

Implements

IReplaySerializeOnReplayDeserialize(ReplayState)

#### ⊿ See Also

## ReplaySnapshotOnReplaySerialize Method

Called by the replay system when this ReplaySnapshot should be serialized.

Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

)

#### Parameters

state

Type: UltimateReplayReplayState The ReplayState to write the data to

Implements IReplaySerializeOnReplaySerialize(ReplayState)

#### ⊿ See Also

## ReplaySnapshotRecordInitialReplay Method

Attempts to record the initial information about a newly created replay object. The initial information will only be sroted if the data is not equal to the default value.

#### Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

C#	Сору
<pre>public void RecordInitialReplayObjectData(</pre>	
ReplayIdentity <i>identity</i> ,	
Vector3 <i>position</i> ,	
Quaternion <i>rotation</i> ,	
Vector3 <i>scale</i>	
)	

#### Parameters

identity

Type: UltimateReplay.CoreReplayIdentity

The object to store the initial data for

position

Type: **Vector3** The initial position of the object

rotation

Type: **Quaternion** The initial rotation of the object

scale

Type: **Vector3** The initial scale of the object

### ⊿ See Also

## ReplaySnapshotRecordSnapshot Method

Registers the specified replay state with this snapshot. The specified identity is used during playback to ensure that the replay objects receives the correct state to deserialize.

#### Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

```
C#

public void RecordSnapshot(

ReplayIdentity identity,

ReplayState state
```

)

#### Parameters

identity

Type: UltimateReplay.CoreReplayIdentity The identity of the object that was serialized

state

Type: UltimateReplayReplayState The state data for the object

### ▲ See Also

## ReplaySnapshotReset Method

Clears all state information from the snapshot but keeps the time stamp.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

## ⊿ Syntax

C#

public void Reset()

#### ⊿ See Also

## ReplaySnapshotRestoreInitialReplay Method

Attempts to get the initial data for the replay object with the specified ReplayIdentity. A ReplayInitialData will be returned containing the initial state for the replay object.

#### Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

#### ▲ Syntax

```
C# ______Copy _____Copy ______Copy ______Copy ______Copy _______Copy ______Copy ______Copy ______Copy ______Copy ______Copy ______Copy _____Copy ____Copy _____Copy ____COPY _____COPY ____COPY ____COPY ____COPY ____COPY ____COPY ___COPY ____C
```

#### Parameters

identity

Type: UltimateReplay.CoreReplayIdentity The identity of the object to get the initial data for

#### Return Value Type: ReplayInitialData

A ReplayInitialData contianing initial values

### ⊿ See Also

## ReplaySnapshotRestoreReplayObje Method

Attempts to restore any replay objects that were spawned or despawned during this snapshot.

Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

)

#### Parameters

scene

Type: UltimateReplay.CoreReplayScene [Missing <param name="scene"/> documentation for "M:UltimateReplay.Storage.ReplaySnapshot.RestoreReplayObjects(UltimateReplay.Cor

Copy

### ⊿ See Also

## ReplaySnapshotRestoreSnapshot Method

Attempts to recall the state information for the specified replay object identity. If the identity does not exist in the scene then the return value will be null.

#### Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

C#	Сору
<pre>public ReplayState RestoreSnapshot(</pre>	

#### Parameters

identity

Type: UltimateReplay.CoreReplayIdentity

The identity of the object to deserialize

**Return Value** 

Type: ReplayState

The state information for the specified identity or null if the identity does not exist

### ⊿ See Also

## **ReplaySnapshot Properties**

The ReplaySnapshot type exposes the following members.

## ▲ Properties

	Name	Description
	Size	Get the size in bytes of the snapshot data.
<b>*</b>	TimeStamp	The time stamp for this snapshot. The time stamp is used to identify the snapshot location in the sequence.

Тор

### ⊿ See Also

# ReplaySnapshotSize Property

Get the size in bytes of the snapshot data.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

C#

public int Size { get; }

Property Value Type: Int32

### ⊿ See Also

# ReplaySnapshotTimeStamp Property

The time stamp for this snapshot. The time stamp is used to identify the snapshot location in the sequence.

Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Copy \_

```
public float TimeStamp { get; }
```

Property Value Type: Single

### ⊿ See Also

# ReplayTarget Class

Represents and abstract storage device capable of holding recorded state data for playback at a later date. Depending upon implementation, a ReplayTarget may be volatile or non-volatile.

## Inheritance Hierarchy

#### SystemObject Object

#### Component Behaviour MonoBehaviour

UltimateReplayReplayBehaviour UltimateReplay.StorageReplayTarget UltimateReplay.StorageReplayMemoryTarget

#### Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0

(1.0.0.0)

### ▲ Syntax

**C**# [SerializableAttribute]

```
public abstract class ReplayTarget : ReplayBehavio
```

Copy

The ReplayTarget type exposes the following members.

## Constructors

j,

Name

Description

ReplayTarget Initializes a new instance of the

#### Тор

## Methods

	Name	Description
≓ <b>©</b>	PrepareTarget	Called by the recording system to notify the active ReplayTarget of an upcoming event.
<b>≓</b> ©	RecordSnapshot	Store a replay snapshot in the replay target.
<b>≡</b> ©	RestoreSnapshot	Recall a snapshot from the replay target based on the specified replay offset.

Тор

## ▲ Fields

	Name	Description
<b>°</b>	duration	The amount of time in seconds that the current recording data lasts. If no data exists then the duration will dfault to a length of 0.
Тор		
▲ Prope	erties	
	Name	Description
<b>*</b>	Duration	The amount of time in seconds that this recording lasts.

MemorySize Get the total amount of bytes that this replay uses.

Тор

## ⊿ See Also

Reference UltimateReplay.Storage Namespace

## ReplayTarget Constructor

Initializes a new instance of the ReplayTarget class

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

C#

protected ReplayTarget()

### ⊿ See Also

# ReplayTarget Fields

The ReplayTarget type exposes the following members.

## ▲ Fields

	Name	Description			
9 <sup>.0</sup>	duration	The amount of time in seconds that the current recording data lasts. If no data exists then the duration will dfault to a length of 0.			
Тор	Тор				
▲ See Also					
Reference ReplayTarget Class UltimateReplay.Storage Namespace					

# ReplayTargetduration Field

The amount of time in seconds that the current recording data lasts. If no data exists then the duration will dfault to a length of 0.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ⊿ Syntax

**C**#

protected float duration

Field Value Type: Single

### ⊿ See Also

## **ReplayTarget Methods**

The ReplayTarget type exposes the following members.

## Methods

	Name	Description
≓ <b>∲</b>	PrepareTarget	Called by the recording system to notify the active ReplayTarget of an upcoming event.
<b>≡</b> ©	RecordSnapshot	Store a replay snapshot in the replay target.
<b>≡</b> ©	RestoreSnapshot	Recall a snapshot from the replay target based on the specified replay offset.

Тор

### ⊿ See Also

## ReplayTargetPrepareTarget Method

Called by the recording system to notify the active ReplayTarget of an upcoming event.

Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

**C**#

)

#### Parameters

mode

Type: UltimateReplay.StorageReplayTargetTask The ReplayTargetTask that the target should prepare for

### ⊿ See Also

## ReplayTargetRecordSnapshot Method

Store a replay snapshot in the replay target.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ⊿ Syntax

C#

Сору

)

#### Parameters

state

Type: UltimateReplay.StorageReplaySnapshot The snapshot to store

### ⊿ See Also

## ReplayTargetRestoreSnapshot Method

Recall a snapshot from the replay target based on the specified replay offset.

Namespace: UltimateReplay.Storage

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

public abstract ReplaySnapshot RestoreSnapshot(
 float offset

Copy

)

#### Parameters

offset

Type: SystemSingle

The offset pointing to the individual snapshot to recall

Return Value Type: ReplaySnapshot The replay snapshot at the specified offset

### ⊿ See Also

## **ReplayTarget Properties**

The ReplayTarget type exposes the following members.

## ▲ Properties

	Name	Description
<b>*</b>	Duration	The amount of time in seconds that this recording lasts.
<b>*</b>	MemorySize	Get the total amount of bytes that this replay uses.
Tere		

Тор

### ⊿ See Also

# ReplayTargetDuration Property

The amount of time in seconds that this recording lasts.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

C#

Copy \_

public abstract float Duration { get; }

Property Value Type: Single

### ⊿ See Also

# ReplayTargetMemorySize Property

Get the total amount of bytes that this replay uses.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Copy \_

public abstract int MemorySize { get; }

Property Value Type: Int32

### ⊿ See Also

## ReplayTargetTask Enumeration

Represents a task that can be issued to a ReplayTarget.

Namespace: UltimateReplay.Storage Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

public enum ReplayTargetTask

### Members

Member name	Value	Description
Commit	0	The replay target should commit all data currently in memeory to its end destination. Similar to a flush method.
Discard	1	The replay target should discard any recorded data.
PrepareWrite	2	The replay target should prepare for subsequent write requests.
PrepareRead	3	The replay target should prepare for subsequent read requests.

## ⊿ See Also

#### Reference

UltimateReplay.Storage Namespace

## UltimateReplay.Util Namespace

## ▲ Classes

	Class	Description
<b>*;</b>	BitConverterNonAlloc	Custom implmenetation of the BitConverter class that does not make any allocations. This is important as the methods may be called thousands of times per second.
<b>*</b> \$	MonoSingletonT	Singleton imeplentation using <b>MonoBehaviour</b> as a base class. A dedicated owner object is created so that the singleton instance can be attached to it.

## BitConverterNonAlloc Class

Custom implmenetation of the BitConverter class that does not make any allocations. This is important as the methods may be called thousands of times per second.

## ▲ Inheritance Hierarchy

SystemObject UltimateReplay.UtilBitConverterNonAlloc

#### Namespace: UltimateReplay.Util

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

### ▲ Syntax

**C**#

Сору

public static class BitConverterNonAlloc

The BitConverterNonAlloc type exposes the following members.

## Methods

	Name	Description
= <b>\$</b>	GetBool	Retreive a 8-bit bool from the specified byte array.
= <b>≬ S</b>	GetBytes(Byte, Boolean)	Store an 8-bit bool into the specified byte array.
= <b>≬ S</b>	GetBytes(Byte, Int16)	Store a 16 bit int into the specified byte array. The buffer to store the int which must have a size of 2 or greaterThe

		short value to store
= <b>≬ S</b>	GetBytes(Byte, Int32)	Store a 32-bit int into the specified byte array.
= <b>≬ S</b>	GetBytes(Byte, Single)	Store a 32-bit float into the specified byte array.
= <b>0 S</b>	GetFloat	Retreive a 32-bit float from the specified byte array.
≓ <b>≬ S</b>	GetInt	Retreive a 32-bit int from the specified byte array.
= <b>≬ S</b>	GetShort	Retreive a 16-bit int from the specified byte array.

Тор

## ⊿ See Also

Reference

UltimateReplay.Util Namespace

## BitConverterNonAlloc Methods

The BitConverterNonAlloc type exposes the following members.

## Methods

	Name	Description
<b>⊧≬ S</b>	GetBool	Retreive a 8-bit bool from the specified byte array.
≓Ŷ S	GetBytes(Byte, Boolean)	Store an 8-bit bool into the specified byte array.
<b>≕∳ S</b>	GetBytes(Byte, Int16)	Store a 16 bit int into the specified byte array. The buffer to store the int which must have a size of 2 or greaterThe short value to store
≓ŷ S	GetBytes(Byte, Int32)	Store a 32-bit int into the specified byte array.
≓Ŷ S	GetBytes(Byte, Single)	Store a 32-bit float into the specified byte array.
≓ŷ S	GetFloat	Retreive a 32-bit float from the specified byte array.
⊧ŷ S	GetInt	Retreive a 32-bit int from the specified byte array.
= <b>0 S</b>	GetShort	Retreive a 16-bit int from the specified byte array.

Тор

## ⊿ See Also

## BitConverterNonAllocGetBool Method

Retreive a 8-bit bool from the specified byte array.

```
Namespace: UltimateReplay.Util
```

```
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)
```

Copy

### ▲ Syntax

**C**#

)

#### Parameters

buffer

Type: SystemByte

The buffer to retreive the bool from which must have a size of 1 or greater

Return Value Type: Boolean The unpacked bool value

## ⊿ See Also

## BitConverterNonAllocGetBytes Method

## Overload List

	Name	Description
≓ <b>≬ S</b>	GetBytes(Byte, Boolean)	Store an 8-bit bool into the specified byte array.
<b>≡</b> ∳ <b>S</b>	GetBytes(Byte, Int16)	Store a 16 bit int into the specified byte array. The buffer to store the int which must have a size of 2 or greaterThe short value to store
≓ <b>≬ S</b>	GetBytes(Byte, Int32)	Store a 32-bit int into the specified byte array.
≓ <b>≬ S</b>	GetBytes(Byte, Single)	Store a 32-bit float into the specified byte array.

Тор

## ⊿ See Also

## BitConverterNonAllocGetBytes Method (Byte, Boolean)

Store an 8-bit bool into the specified byte array.

Namespace: UltimateReplay.Util

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy

### ▲ Syntax

C#

```
public static void GetBytes(
          byte[] buffer,
          bool value
```

)

#### Parameters

buffer

Type: SystemByte

The buffer to store the bool which must have a size of 1 or greater

value

Type: SystemBoolean The bool value to store

## ⊿ See Also

Reference BitConverterNonAlloc Class GetBytes Overload UltimateReplay.Util Namespace

## BitConverterNonAllocGetBytes Method (Byte, Int16)

Store a 16 bit int into the specified byte array. The buffer to store the int which must have a size of 2 or greaterThe short value to store

Namespace: UltimateReplay.Util

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

)

#### Parameters

#### buffer

Type: SystemByte [Missing <param name="buffer"/> documentation for "M:UltimateReplay.Util.BitConverterNonAlloc.GetBytes(System.Byte[],System.Int16)"]

Copy

#### value

Type: SystemInt16 [Missing <param name="value"/> documentation for "M:UltimateReplay.Util.BitConverterNonAlloc.GetBytes(System.Byte[],System.Int16)"]

### ⊿ See Also

Reference BitConverterNonAlloc Class GetBytes Overload UltimateReplay.Util Namespace

## BitConverterNonAllocGetBytes Method (Byte, Int32)

Store a 32-bit int into the specified byte array.

```
Namespace: UltimateReplay.Util
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

## ▲ Syntax

**C**#

```
public static void GetBytes(
            byte[] buffer,
            int value
```

)

#### Parameters

buffer

Type: SystemByte

The buffer to store the int which must have a size of 4 or greater

value

Type: SystemInt32 The int value to store

## ▲ See Also

Reference BitConverterNonAlloc Class GetBytes Overload UltimateReplay.Util Namespace

## BitConverterNonAllocGetBytes Method (Byte, Single)

Store a 32-bit float into the specified byte array.

```
Namespace: UltimateReplay.Util
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

## ⊿ Syntax

C#

```
public static void GetBytes(
          byte[] buffer,
          float value
```

)

#### Parameters

buffer

Type: SystemByte

The buffer to store the float which must have a size of 4 or greated

Copy

value

Type: SystemSingle The float value to store

## ▲ See Also

Reference BitConverterNonAlloc Class GetBytes Overload UltimateReplay.Util Namespace

## BitConverterNonAllocGetFloat Method

Retreive a 32-bit float from the specified byte array.

```
Namespace: UltimateReplay.Util
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

## ⊿ Syntax

**C**#

)

#### Parameters

buffer

Type: SystemByte

The buffer to retreive the float from which must have a size of 4 or greater

Copy

Return Value Type: Single The unpacked float value

### ⊿ See Also

## BitConverterNonAllocGetInt Method

Retreive a 32-bit int from the specified byte array.

```
Namespace: UltimateReplay.Util
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

Copy

## ▲ Syntax

**C**#

)

#### Parameters

buffer

Type: SystemByte

The buffer to retreive the int from which must have a size of 4 or greater

Return Value Type: Int32 The unpacked int value

### ⊿ See Also

## BitConverterNonAllocGetShort Method

Retreive a 16-bit int from the specified byte array.

```
Namespace: UltimateReplay.Util
Assembly: UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0
(1.0.0.0)
```

## ⊿ Syntax

**C**#

)

#### Parameters

buffer

Type: SystemByte

The buffer to retreive the short from which must have a size of 2 or greater

Copy

Return Value Type: Int16 The unpacked short value

### ⊿ See Also

# MonoSingletonT Class

Singleton imeplentation using **MonoBehaviour** as a base class. A dedicated owner object is created so that the singleton instance can be attached to it.

## Inheritance Hierarchy

SystemObject Object

Component Behaviour MonoBehaviour UltimateReplay.UtilMonoSingletonT UltimateReplayReplayManager

Namespace: UltimateReplay.Util

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

#### **Type Parameters**

Т

The generic type that the singleton is wrapping

The MonoSingletonT type exposes the following members.

## Constructors

	Name		Description
ġ.	MonoSir	•	Initializes a new instance of the MonoSingletonT class
Тор			
▲ Metho	ds		
	Name		Description
≓∳	Awake		Called by Unity. Allows the singleton to find and claim any scene instances.
≓ <b>≬ S</b>	ForceAwake		Attempts to forcefully create the singleton instance if one does not already exist.
≡ <b>≬</b>	OnApplicationQuit		Called by Unity. Allows the singleton to prevent recreation of the instance when the game is about to quit.
Тор			
⊿ Fields			
	Name	Descrip	tion
9 <sup>9</sup> S	active		ve singleton instance or null if no has been created yet. Use Active to

guarentee that the instance is non-null.

#### Тор

## ▲ Properties

	Name	Description
🖹 S	Active	Get the active singleton instance. If not instance exists then one is created.
i	IsDisposing	Returns true if the singleton is currently disposing. This will only occur when the game is about to quit.
Тор		

## ⊿ See Also

Reference UltimateReplay.Util Namespace

## MonoSingletonT Constructor

Initializes a new instance of the MonoSingletonT class

Namespace: UltimateReplay.Util

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

C#

protected MonoSingleton()

## ⊿ See Also

# MonoSingletonT Fields

The MonoSingletonT generic type exposes the following members.

## ▲ Fields

	Name	Description		
° <sup>₽</sup> S	active	The active singleton instance or null if no instance has been created yet. Use Active to guarentee that the instance is non-null.		
Тор				
⊿ See Also				
Referen	се			

# MonoSingletonTactive Field

The active singleton instance or null if no instance has been created yet. Use Active to guarentee that the instance is non-null.

Namespace: UltimateReplay.Util

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ⊿ Syntax

**C**#

protected static T active

Field Value Type: T

## ⊿ See Also

# MonoSingletonT Methods

The MonoSingletonT generic type exposes the following members.

## ▲ Methods

	Name	Description
. <b>≡©</b>	Awake	Called by Unity. Allows the singleton to find and claim any scene instances.
≡ŷ S	ForceAwake	Attempts to forcefully create the singleton instance if one does not already exist.
<b>≡</b> ©	OnApplicationQuit	Called by Unity. Allows the singleton to prevent recreation of the instance when the game is about to quit.

Тор

## ⊿ See Also

# MonoSingletonTAwake Method

Called by Unity. Allows the singleton to find and claim any scene instances.

Namespace: UltimateReplay.Util

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

public virtual void Awake()

## ⊿ See Also

# MonoSingleton*T*ForceAwake Method

Attempts to forcefully create the singleton instance if one does not already exist.

Namespace: UltimateReplay.Util

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Copy \_

```
public static T ForceAwake()
```

Return Value Type: T The active singleton instance

## ⊿ See Also

# MonoSingletonTOnApplicationQuit Method

Called by Unity. Allows the singleton to prevent recreation of the instance when the game is about to quit.

Namespace: UltimateReplay.Util

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Copy \_

public void OnApplicationQuit()

## ⊿ See Also

# MonoSingletonT Properties

The MonoSingletonT generic type exposes the following members.

## ▲ Properties

	Name	Description
📽 s	Active	Get the active singleton instance. If not instance exists then one is created.
🚰 S	IsDisposing	Returns true if the singleton is currently disposing. This will only occur when the game is about to quit.

Тор

## ▲ See Also

# MonoSingletonTActive Property

Get the active singleton instance. If not instance exists then one is created.

Namespace: UltimateReplay.Util

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

Copy \_

## ▲ Syntax

**C**#

public static T Active { get; }

Property Value Type: T

## ⊿ See Also

# MonoSingleton*T*IsDisposing Property

Returns true if the singleton is currently disposing. This will only occur when the game is about to quit.

Namespace: UltimateReplay.Util

**Assembly:** UltimateReplay (in UltimateReplay.dll) Version: 1.0.0.0 (1.0.0.0)

## ▲ Syntax

**C**#

Сору

public static bool IsDisposing { get; }

Property Value Type: Boolean

## ⊿ See Also