Table of Contents

Namespace JCMG.JTween

Classes

ArrayExtensions

Extension methods for arrays

JTweenControl

The global instance for JTween to interact with tweens, tween systems.

Singleton<T>

A basic scene Singleton implementation.

TransformExtensions

Interfaces

ITweenCollection

The base interface collection for ITweenHandles.

ITweenHandle

A user reference to tween data that allows for safe manipulation of its state.

ITweenSequence

A ITweenHandle collection whose contents are operated in succession. A started event is invoked when first played and completed once the last ITweenHandle instance has completed.

ITweenSet

A ITweenHandle collection whose contents are operated on all at once. A started event is invoked when first played and completed once all ITweenHandle instances have completed.

Enums

EaseType

The type of easing to use for a given tween.

LoopType

The type of looping the tween should use while running.

RotateMode

The mode a tween should use when animating rotation.

SpaceType

The coordinate space system the tween should operate with regards to.

Class ArrayExtensions

Extension methods for arrays

Inheritance

System.Object

ArrayExtensions

Namespace: JCMG.JTween

Assembly: JCMG.JTween.dll

Syntax

```
public static class ArrayExtensions
```

Methods

Populate<T>(T[], T)

Assigns the `T` `value` to all elements in this array.

Declaration

```
public static void Populate<T>(this T[] array, T value)
```

Parameters

TYPE	NAME	DESCRIPTION
T[]	array	
T	value	

Type Parameters

NAME	DESCRIPTION
T	

PopulatePositionArray(Vector3[], IList<Transform>, SpaceType)

Populates the UnityEngine.Vector3 `positionArray` with the appropriate position from IList `transformList` based on the passed SpaceType `spaceType`. The `positionArray` and `transformList` must be of equal length.

Declaration

```
public static void PopulatePositionArray(this Vector3[] positionArray, IList<Transform> transformList,
SpaceType spaceType)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector3[]	positionArray	The array of UnityEngine.Vector3s positions will be assigned to.

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IList<UnityEngine.Transform>	transformList	The IList of UnityEngine.Transforms positions will be assigned from.
SpaceType	spaceType	Whether or not the position assigned should be in world or local space.

PopulatePositionArray(Vector3[], Transform[], SpaceType)

Populates the UnityEngine.Vector3 `positionArray` with the appropriate position from UnityEngine.Transform `transformArray` based on the passed SpaceType `spaceType`. The `positionArray` and `transformArray` must be of equal length.

Declaration

```
public static void PopulatePositionArray(this Vector3[] positionArray, Transform[] transformArray, SpaceType spaceType)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector3[]	positionArray	The array of UnityEngine.Vector3s positions will be assigned to.
UnityEngine.Transform[]	transformArray	The array of UnityEngine.Transforms positions will be assigned from.
SpaceType	spaceType	Whether or not the position assigned should be in world or local space.

PopulateRotationArray(Quaternion[], IList<Transform>, SpaceType)

Populates the UnityEngine.Quaternion `rotationArray` with the appropriate rotation from IList `transformList` based on the passed SpaceType `spaceType`. The `rotationArray` and `transformList` must be of equal length.

Declaration

```
public static void PopulateRotationArray(this Quaternion[] rotationArray, IList<Transform> transformList, SpaceType spaceType)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Quaternion[]	rotationArray	The array of UnityEngine.Quaternions rotations will be assigned to.
System.Collections.Generic.IList<UnityEngine.Transform>	transformList	The IList of UnityEngine.Transforms rotations will be assigned from.

TYPE	NAME	DESCRIPTION
SpaceType	spaceType	Whether or not the rotation assigned should be in world or local space.

PopulateRotationArray(Quaternion[], Transform[], SpaceType)

Populates the UnityEngine.Quaternion `rotationArray` with the appropriate rotation from UnityEngine.Transform `transformArray` based on the passed SpaceType `spaceType`. The `rotationArray` and `transformArray` must be of equal length.

Declaration

```
public static void PopulateRotationArray(this Quaternion[] rotationArray, Transform[] transformArray,
SpaceType spaceType)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Quaternion[]	rotationArray	The array of UnityEngine.Quaternions rotations will be assigned to.
UnityEngine.Transform[]	transformArray	The array of UnityEngine.Transforms rotations will be assigned from.
SpaceType	spaceType	Whether or not the rotation assigned should be in world or local space.

PopulateScaleArray(Vector3[], IList<Transform>)

Populates the UnityEngine.Vector3 `scaleArray` with the scale from IList UnityEngine.Transform `transformList`. The `scaleArray` and `transformList` must be of equal length.

Declaration

```
public static void PopulateScaleArray(this Vector3[] scaleArray, IList<Transform> transformList)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector3[]	scaleArray	The array of UnityEngine.Vector3s scale will be assigned to.
System.Collections.Generic.IList<UnityEngine.Transform>	transformList	The IList of UnityEngine.Transforms scale will be assigned from.

PopulateScaleArray(Vector3[], Transform[])

Populates the UnityEngine.Vector3 `scaleArray` with the scale from array UnityEngine.Transform `transformArray`. The `scaleArray` and `transformArray` must be of equal length.

Declaration

```
public static void PopulateScaleArray(this Vector3[] scaleArray, Transform[] transformArray)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector3[]	scaleArray	The array of UnityEngine.Vector3s scale will be assigned to.
UnityEngine.Transform[]	transformArray	The array of UnityEngine.Transforms scale will be assigned from.

Enum EaseType

The type of easing to use for a given tween.

Namespace: JCMG.JTween

Assembly: JCMG.JTween.dll

Syntax

```
[Serializable]
public enum EaseType : byte
```

Fields

NAME	DESCRIPTION
BackIn	
BackInOut	
BackOut	
BounceIn	
BounceInOut	
BounceOut	
CircIn	
CircInOut	
CircOut	
CubicIn	
CubicInOut	
CubicOut	
ElasticIn	
ElasticInOut	
ElasticOut	
ExpoIn	
ExpoInOut	
ExpoOut	
Linear	
Punch	

NAME	DESCRIPTION
QuadIn	
QuadInOut	
QuadOut	
QuartIn	
QuartInOut	
QuartOut	
QuintIn	
QuintInOut	
QuintOut	
SineIn	
SineInOut	
SineOut	

Interface ITweenCollection

The base interface collection for ITweenHandles.

Namespace: JCMG.JTween

Assembly: JCMG.JTween.dll

Syntax

```
public interface ITweenCollection
```

Methods

Add(ITweenHandle)

Adds the ITweenHandle instance to the collection.

Declaration

```
void Add(ITweenHandle tweenHandle)
```

Parameters

TYPE	NAME	DESCRIPTION
ITweenHandle	tweenHandle	

AddOnComplete(Action)

Adds a listener that is invoked when the collection has completed.

Declaration

```
void AddOnComplete(Action onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	onComplete	

AddOnStarted(Action)

Adds a listener that is invoked when the collection has begun playing.

Declaration

```
void AddOnStarted(Action onStart)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	onStart	

Clear()

Recycles all ITweenHandle instances from the collection and clears all local event listeners.

Declaration

```
void Clear()
```

Recycle()

Recycles all ITweenHandle instances in the collection and removes them. All local event listeners will remain.

Declaration

```
void Recycle()
```

Interface ITweenHandle

A user reference to tween data that allows for safe manipulation of its state.

Namespace: JCMG.JTween

Assembly: JCMG.JTween.dll

Syntax

```
public interface ITweenHandle
```

Methods

AddOnCompletedListener(Action)

Adds an event listener that is called when the tween has completed or when Stop() is called while playing/paused.

Declaration

```
void AddOnCompletedListener(Action onCompleted)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	onCompleted	

AddOnStartedListener(Action)

Adds an event listener that is called when the tween is started via Play().

Declaration

```
void AddOnStartedListener(Action onStarted)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	onStarted	

IsCompleted()

Returns true if the tween is completed, otherwise false.

Declaration

```
bool IsCompleted()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

IsPaused()

Returns true if the tween is paused, otherwise false.

Declaration

```
bool IsPaused()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

IsPlaying()

returns true if the tween is playing, otherwise false.

Declaration

```
bool IsPlaying()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

Pause()

Pauses the tween instance if playing.

Declaration

```
void Pause()
```

Play()

Plays the tween instance if paused or not completed. Any listeners added via AddOnStartedListener(Action) will be invoked.

Declaration

```
void Play()
```

Recycle()

Immediately stops the tween and marks the tween as requiring recycling. Any local reference to this ITweenHandle instance should be cleared by setting it to null. If playing or paused, this will not invoke listeners added via AddOnCompletedListener(Action).

Declaration

```
void Recycle()
```

Restart()

Rewinds the tween data back to its original state and immediately plays it.

Declaration

```
void Restart()
```

Rewind()

Rewinds the tween data back to its original state and sets it as paused.

Declaration

```
void Rewind()
```

Stop()

Stops the tween instance if playing or paused and and marks it as completed. Any listeners added via AddOnCompletedListener(Action) will be invoked.

Declaration

```
void Stop()
```

Interface ITweenSequence

A ITweenHandle collection whose contents are operated in succession. A started event is invoked when first played and completed once the last ITweenHandle instance has completed.

Inherited Members

ITweenCollection.Add(ITweenHandle)
ITweenCollection.Clear()
ITweenCollection.Recycle()
ITweenCollection.AddOnComplete(Action)
ITweenCollection.AddOnStarted(Action)

Namespace: JCMG.JTween

Assembly: JCMG.JTween.dll

Syntax

```
public interface ITweenSequence : ITweenCollection
```

Methods

AddOnStep(Action)

Adds a listener that is invoked every time a ITweenHandle in the sequence has completed.

Declaration

```
void AddOnStep(Action onStep)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	onStep	

Pause()

Pauses the current ITweenHandle instance in the sequence if playing.

Declaration

```
void Pause()
```

Play()

If not playing, plays the first ITweenHandle instance in the sequence. Otherwise if the current ITweenHandle instance in the sequence is paused that will be played.

Declaration

```
void Play()
```

Restart()

Rewinds all ITweenHandle instances in the sequence and plays the first one.

Declaration

```
void Restart()
```

Rewind()

Rewinds all ITweenHandle instances in the sequence and initialized the first one as paused.

Declaration

```
void Rewind()
```

Stop()

Stops the currently playing ITweenHandle instance in the sequence if any and marks it as complete.

Declaration

```
void Stop()
```

Interface ITweenSet

A ITweenHandle collection whose contents are operated on all at once. A started event is invoked when first played and completed once all ITweenHandle instances have completed.

Inherited Members

ITweenCollection.Add(ITweenHandle)
ITweenCollection.Clear()
ITweenCollection.Recycle()
ITweenCollection.AddOnComplete(Action)
ITweenCollection.AddOnStarted(Action)

Namespace: JCMG.JTween

Assembly: JCMG.JTween.dll

Syntax

```
public interface ITweenSet : ITweenCollection
```

Methods

Pause()

Pauses all ITweenHandle instances in the ITweenSet.

Declaration

```
void Pause()
```

Play()

Plays all ITweenHandle instances in the ITweenSet.

Declaration

```
void Play()
```

Restart()

Rewinds all ITweenHandle instances in the ITweenSet and plays them.

Declaration

```
void Restart()
```

Rewind()

Rewinds all ITweenHandle instances in the ITweenSet and pauses them.

Declaration

```
void Rewind()
```

Stop()

Stops all ITweenHandle instances in the ITweenSet and marks them as complete.

Declaration

```
void Stop()
```

Class JTweenControl

The global instance for JTween to interact with tweens, tween systems.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
Singleton<JTweenControl>
JTweenControl

Inherited Members

Singleton<JTweenControl>.Instance
Singleton<JTweenControl>.Exists
Singleton<JTweenControl>.OnApplicationQuit()

Namespace: JCMG.JTween

Assembly: JCMG.JTween.dll

Syntax

```
[AddComponentMenu("JCMG/JTween/JTweenControl")]
[RequireComponent(typeof(SingleTransformTweener), typeof(BatchTransformTweener))]
public sealed class JTweenControl : Singleton<JTweenControl>
```

Methods

Awake()

Declaration

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

Overrides

JCMG.JTween.Singleton<JCMG.JTween.JTweenControl>.Awake()

BatchMove(Transform[], Vector3[], Vector3[], Single, out ITweenHandle, SpaceType, EaseType, LoopType, Int32)

Creates a batch of movement tweens for the UnityEngine.Transform[] `targets`.

Declaration

```
public void BatchMove(Transform[] targets, Vector3[] fromArray, Vector3[] toArray, float duration, out
ITweenHandle tweenHandle, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType
loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Vector3[]	fromArray	The UnityEngine.Vector3[] where the move from position should be assigned from.

TYPE	NAME	DESCRIPTION
UnityEngine.Vector3[]	toArray	The UnityEngine.Vector3[] where the move to position should be assigned from.
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

BatchMove(Transform[], Vector3[], Vector3[], Single, SpaceType, EaseType, LoopType, Int32, Action, Action)

Creates a batch of movement tweens for the UnityEngine.Transform[] `targets`.

Declaration

```
public void BatchMove(Transform[] targets, Vector3[] fromArray, Vector3[] toArray, float duration, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Vector3[]	fromArray	The UnityEngine.Vector3[] where the move from position should be assigned from.
UnityEngine.Vector3[]	toArray	The UnityEngine.Vector3[] where the move to position should be assigned from.
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.

TYPE	NAME	DESCRIPTION
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween batch begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween batch completes (default is NULL).

BatchMoveSlice(Transform[], Vector3[], Vector3[], Int32, Int32, Single, out ITweenHandle, SpaceType, EaseType, LoopType, Int32)

Creates a batch of movement tweens for a slice from UnityEngine.Transform[] `targets` starting at `startIndex` to `length`.

Declaration

```
public void BatchMoveSlice(Transform[] targets, Vector3[] fromArray, Vector3[] toArray, int startIndex, int length, float duration, out ITweenHandle tweenHandle, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Vector3[]	fromArray	The UnityEngine.Vector3[] where the move from position should be assigned from.
UnityEngine.Vector3[]	toArray	The UnityEngine.Vector3[] where the move to position should be assigned from.
System.Int32	startIndex	The index where the slice should start from in the parameter arrays.
System.Int32	length	The length from which values should copied in the parameter arrays starting from the `startIndex`.

TYPE	NAME	DESCRIPTION
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

BatchMoveSlice(Transform[], Vector3[], Vector3[], Int32, Int32, Single, SpaceType, EaseType, LoopType, Int32, Action, Action)

Creates a batch of movement tweens for a slice from UnityEngine.Transform[] `targets` starting at `startIndex` to `length`.

Declaration

```
public void BatchMoveSlice(Transform[] targets, Vector3[] fromArray, Vector3[] toArray, int startIndex, int length, float duration, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Vector3[]	fromArray	The UnityEngine.Vector3[] where the move from position should be assigned from.
UnityEngine.Vector3[]	toArray	The UnityEngine.Vector3[] where the move to position should be assigned from.
System.Int32	startIndex	The index where the slice should start from in the parameter arrays.
System.Int32	length	The length from which values should copied in the parameter arrays starting from the `startIndex`.

TYPE	NAME	DESCRIPTION
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween batch begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween batch completes (default is NULL).

BatchRotate(Transform[], Quaternion[], Quaternion[], Single, out ITweenHandle, SpaceType, EaseType, LoopType, Int32)

Creates a batch of rotation tweens for the UnityEngine.Transform[] `targets`.

Declaration

```
public void BatchRotate(Transform[] targets, Quaternion[] fromArray, Quaternion[] toArray, float duration, out ITweenHandle tweenHandle, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Quaternion[]	fromArray	The UnityEngine.Quaternion[] where the from rotation should be assigned from.
UnityEngine.Quaternion[]	toArray	The UnityEngine.Quaternion[] where the to rotation should be assigned from.
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.

TYPE	NAME	DESCRIPTION
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

BatchRotate(Transform[], Quaternion[], Quaternion[], Single, SpaceType, EaseType, LoopType, Int32, Action, Action)

Creates a batch of rotation tweens for the UnityEngine.Transform[] `targets`.

Declaration

```
public void BatchRotate(Transform[] targets, Quaternion[] fromArray, Quaternion[] toArray, float duration,
SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None,
int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Quaternion[]	fromArray	The UnityEngine.Quaternion[] where the from rotation should be assigned from.
UnityEngine.Quaternion[]	toArray	The UnityEngine.Quaternion[] where the to rotation should be assigned from.
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).

TYPE	NAME	DESCRIPTION
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween batch begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween batch completes (default is NULL).

BatchRotateSlice(Transform[], Quaternion[], Quaternion[], Int32, Int32, Single, out ITweenHandle, SpaceType, EaseType, LoopType, Int32)

Creates a batch of rotation tweens for a slice from UnityEngine.Transform[] `targets` starting at `startIndex` to `length`.

Declaration

```
public void BatchRotateSlice(Transform[] targets, Quaternion[] fromArray, Quaternion[] toArray, int startIndex, int length, float duration, out ITweenHandle tweenHandle, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Quaternion[]	fromArray	The UnityEngine.Quaternion[] where the from rotation should be assigned from.
UnityEngine.Quaternion[]	toArray	The UnityEngine.Quaternion[] where the to rotation should be assigned from.
System.Int32	startIndex	The index where the slice should start from in the parameter arrays.
System.Int32	length	The length from which values should copied in the parameter arrays starting from the `startIndex`.
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).

TYPE	NAME	DESCRIPTION
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

BatchRotateSlice(Transform[], Quaternion[], Quaternion[], Int32, Int32, Single, SpaceType, EaseType, LoopType, Int32, Action, Action)

Creates a batch of rotation tweens for a slice from UnityEngine.Transform[] `targets` starting at `startIndex` to `length`.

Declaration

```
public void BatchRotateSlice(Transform[] targets, Quaternion[] fromArray, Quaternion[] toArray, int startIndex, int length, float duration, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Quaternion[]	fromArray	The UnityEngine.Quaternion[] where the from rotation should be assigned from.
UnityEngine.Quaternion[]	toArray	The UnityEngine.Quaternion[] where the to rotation should be assigned from.
System.Int32	startIndex	The index where the slice should start from in the parameter arrays.
System.Int32	length	The length from which values should copied in the parameter arrays starting from the `startIndex`.
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).

TYPE	NAME	DESCRIPTION
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween batch begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween batch completes (default is NULL).

BatchScale(Transform[], Vector3[], Vector3[], Single, out ITweenHandle, SpaceType, EaseType, LoopType, Int32)

Creates a batch of scale tweens for the UnityEngine.Transform[] `targets`.

Declaration

```
public void BatchScale(Transform[] targets, Vector3[] fromArray, Vector3[] toArray, float duration, out ITweenHandle tweenHandle, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Vector3[]	fromArray	The UnityEngine.Vector3[] where the from scale should be assigned from.
UnityEngine.Vector3[]	toArray	The UnityEngine.Vector3[] where the to scale should be assigned from.
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).

TYPE	NAME	DESCRIPTION
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

BatchScale(Transform[], Vector3[], Vector3[], Single, SpaceType, EaseType, LoopType, Int32, Action, Action)

Creates a batch of scale tweens for the UnityEngine.Transform[] `targets`.

Declaration

```
public void BatchScale(Transform[] targets, Vector3[] fromArray, Vector3[] toArray, float duration, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Vector3[]	fromArray	The UnityEngine.Vector3[] where the from scale should be assigned from.
UnityEngine.Vector3[]	toArray	The UnityEngine.Vector3[] where the to scale should be assigned from.
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween batch begins playing (default is NULL).

TYPE	NAME	DESCRIPTION
System.Action	onComplete	The System.Action that should be invoked when the tween batch completes (default is NULL).

BatchScaleSlice(Transform[], Vector3[], Vector3[], Int32, Int32, Single, out ITweenHandle, SpaceType, EaseType, LoopType, Int32)

Creates a batch of scale tweens for a slice from UnityEngine.Transform[] `targets` starting at `startIndex` to `length`.

Declaration

```
public void BatchScaleSlice(Transform[] targets, Vector3[] fromArray, Vector3[] toArray, int startIndex, int length, float duration, out ITweenHandle tweenHandle, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Vector3[]	fromArray	The UnityEngine.Vector3[] where the from scale should be assigned from.
UnityEngine.Vector3[]	toArray	The UnityEngine.Vector3[] where the to scale should be assigned from.
System.Int32	startIndex	The index where the slice should start from in the parameter arrays.
System.Int32	length	The length from which values should copied in the parameter arrays starting from the `startIndex`.
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).

TYPE	NAME	DESCRIPTION
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

BatchScaleSlice(Transform[], Vector3[], Vector3[], Int32, Int32, Single, SpaceType, EaseType, LoopType, Int32, Action, Action)

Creates a batch of scale tweens for a slice from UnityEngine.Transform[] `targets` starting at `startIndex` to `length`.

Declaration

```
public void BatchScaleSlice(Transform[] targets, Vector3[] fromArray, Vector3[] toArray, int startIndex, int length, float duration, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Vector3[]	fromArray	The UnityEngine.Vector3[] where the from scale should be assigned from.
UnityEngine.Vector3[]	toArray	The UnityEngine.Vector3[] where the to scale should be assigned from.
System.Int32	startIndex	The index where the slice should start from in the parameter arrays.
System.Int32	length	The length from which values should copied in the parameter arrays starting from the `startIndex`.
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

TYPE	NAME	DESCRIPTION
System.Action	onStart	The System.Action that should be invoked when the tween batch begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween batch completes (default is NULL).

BatchUpdateTransforms(Transform[], Vector3[], Vector3[], Quaternion[], Quaternion[], Vector3[], Vector3[], Single, out ITweenHandle, SpaceType, EaseType, LoopType, Int32)

Creates a batch of tweens animating movement, rotation, and scaling for the UnityEngine.Transform[] `targets`.

Declaration

```
public void BatchUpdateTransforms(Transform[] targets, Vector3[] fromPosArray, Vector3[] toPosArray,
Quaternion[] fromRotArray, Quaternion[] toRotArray, Vector3[] fromScaleArray, Vector3[] toScaleArray, float
duration, out ITweenHandle tweenHandle, SpaceType spaceType = SpaceType.World, EaseType easeType =
EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Vector3[]	fromPosArray	The UnityEngine.Vector3[] where the from position should be assigned from.
UnityEngine.Vector3[]	toPosArray	The UnityEngine.Vector3[] where the from position should be assigned from.
UnityEngine.Quaternion[]	fromRotArray	The UnityEngine.Quaternion[] where the from rotation should be assigned from.
UnityEngine.Quaternion[]	toRotArray	The UnityEngine.Quaternion[] where the to rotation should be assigned from.
UnityEngine.Vector3[]	fromScaleArray	The UnityEngine.Vector3[] where the from scale should be assigned from.
UnityEngine.Vector3[]	toScaleArray	The UnityEngine.Vector3[] where the to scale should be assigned from.
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.

TYPE	NAME	DESCRIPTION
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

BatchUpdateTransforms(Transform[], Vector3[], Vector3[], Quaternion[], Quaternion[], Vector3[], Vector3[], Single, SpaceType, EaseType, LoopType, Int32, Action, Action)

Creates a batch of tweens animating movement, rotation, and scaling for the UnityEngine.Transform[] `targets`.

Declaration

```
public void BatchUpdateTransforms(Transform[] targets, Vector3[] fromPosArray, Vector3[] toPosArray,
Quaternion[] fromRotArray, Quaternion[] toRotArray, Vector3[] fromScaleArray, Vector3[] toScaleArray, float
duration, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType =
LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Vector3[]	fromPosArray	The UnityEngine.Vector3[] where the from position should be assigned from.
UnityEngine.Vector3[]	toPosArray	The UnityEngine.Vector3[] where the from position should be assigned from.
UnityEngine.Quaternion[]	fromRotArray	The UnityEngine.Quaternion[] where the from rotation should be assigned from.
UnityEngine.Quaternion[]	toRotArray	The UnityEngine.Quaternion[] where the to rotation should be assigned from.
UnityEngine.Vector3[]	fromScaleArray	The UnityEngine.Vector3[] where the from scale should be assigned from.
UnityEngine.Vector3[]	toScaleArray	The UnityEngine.Vector3[] where the to scale should be assigned from.

TYPE	NAME	DESCRIPTION
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween batch begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween batch completes (default is NULL).

BatchUpdateTransformsSlice(Transform[], Vector3[], Vector3[], Quaternion[], Quaternion[], Vector3[], Vector3[], Int32, Int32, Single, out ITweenHandle, SpaceType, EaseType, LoopType, Int32)

Creates a batch of tweens animating movement, rotation, and scaling for a slice from UnityEngine.Transform[] `targets` starting at `startIndex` to `length`.

Declaration

```
public void BatchUpdateTransformsSlice(Transform[] targets, Vector3[] fromPosArray, Vector3[] toPosArray, Quaternion[] fromRotArray, Quaternion[] toRotArray, Vector3[] fromScaleArray, Vector3[] toScaleArray, int startIndex, int length, float duration, out ITweenHandle tweenHandle, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Vector3[]	fromPosArray	The UnityEngine.Vector3[] where the from position should be assigned from.
UnityEngine.Vector3[]	toPosArray	The UnityEngine.Vector3[] where the from position should be assigned from.

TYPE	NAME	DESCRIPTION
UnityEngine.Quaternion[]	fromRotArray	The UnityEngine.Quaternion[] where the from rotation should be assigned from.
UnityEngine.Quaternion[]	toRotArray	The UnityEngine.Quaternion[] where the to rotation should be assigned from.
UnityEngine.Vector3[]	fromScaleArray	The UnityEngine.Vector3[] where the from scale should be assigned from.
UnityEngine.Vector3[]	toScaleArray	The UnityEngine.Vector3[] where the to scale should be assigned from.
System.Int32	startIndex	The index where the slice should start from in the parameter arrays.
System.Int32	length	The length from which values should copied in the parameter arrays starting from the `startIndex`.
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

BatchUpdateTransformsSlice(Transform[], Vector3[], Vector3[], Quaternion[], Quaternion[], Vector3[], Vector3[], Int32, Int32, Single, SpaceType, EaseType, LoopType, Int32, Action, Action)

Creates a batch of tweens animating movement, rotation, and scaling for a slice from UnityEngine.Transform[] `targets` starting at `startIndex` to `length`.

Declaration

```
public void BatchUpdateTransformsSlice(Transform[] targets, Vector3[] fromPosArray, Vector3[] toPosArray, Quaternion[] fromRotArray, Quaternion[] toRotArray, Vector3[] fromScaleArray, Vector3[] toScaleArray, int startIndex, int length, float duration, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform[]	targets	The UnityEngine.Transform[] that are the targets of this tween batch.
UnityEngine.Vector3[]	fromPosArray	The UnityEngine.Vector3[] where the from position should be assigned from.
UnityEngine.Vector3[]	toPosArray	The UnityEngine.Vector3[] where the from position should be assigned from.
UnityEngine.Quaternion[]	fromRotArray	The UnityEngine.Quaternion[] where the from rotation should be assigned from.
UnityEngine.Quaternion[]	toRotArray	The UnityEngine.Quaternion[] where the to rotation should be assigned from.
UnityEngine.Vector3[]	fromScaleArray	The UnityEngine.Vector3[] where the from scale should be assigned from.
UnityEngine.Vector3[]	toScaleArray	The UnityEngine.Vector3[] where the to scale should be assigned from.
System.Int32	startIndex	The index where the slice should start from in the parameter arrays.
System.Int32	length	The length from which values should copied in the parameter arrays starting from the `startIndex`.
System.Single	duration	The length of time in seconds that the tween batch should take to complete or one pass if looping.
SpaceType	spaceType	The coordinate system the tween batch should operate in (default is World).
EaseType	easeType	The type of easing the tween batch should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).

TYPE	NAME	DESCRIPTION
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween batch begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween batch completes (default is NULL).

Move(Transform, Vector3, Vector3, Single, out ITweenHandle, SpaceType, EaseType, LoopType, Int32)

Moves the UnityEngine.Transform `target`.

Declaration

```
public void Move(Transform target, Vector3 from, Vector3 to, float duration, out ITweenHandle tweenHandle,
SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None,
int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	target	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	from	The position the UnityEngine.Transform will be moved from.
UnityEngine.Vector3	to	The position the UnityEngine.Transform will be moved to.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
SpaceType	spaceType	The coordinate system the tween should operate in (default is World).
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).

TYPE	NAME	DESCRIPTION
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

Move(Transform, Vector3, Vector3, Single, SpaceType, EaseType, LoopType, Int32, Action, Action)

Moves the UnityEngine.Transform `target`.

Declaration

```
public void Move(Transform target, Vector3 from, Vector3 to, float duration, SpaceType spaceType =
SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0,
Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	target	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	from	The position the UnityEngine.Transform will be moved from.
UnityEngine.Vector3	to	The position the UnityEngine.Transform will be moved to.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
SpaceType	spaceType	The coordinate system the tween should operate in (default is World).
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

NewSequence()

Creates a new instance of ITweenSequence.

Declaration

```
public ITweenSequence NewSequence()
```

Returns

TYPE	DESCRIPTION
ITweenSequence	

NewSet()

Creates a new instance of ITweenSet.

Declaration

```
public ITweenSet NewSet()
```

Returns

TYPE	DESCRIPTION
ITweenSet	

Rotate(Transform, Quaternion, Quaternion, Single, out ITweenHandle, SpaceType, EaseType, LoopType, Int32)

Rotates the UnityEngine.Transform `target`.

Declaration

```
public void Rotate(Transform target, Quaternion from, Quaternion to, float duration, out ITweenHandle
tweenHandle, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType =
LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	target	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Quaternion	from	The rotation the UnityEngine.Transform will be animated from.
UnityEngine.Quaternion	to	The rotation the UnityEngine.Transform will be animated to.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
SpaceType	spaceType	The coordinate system the tween should operate in (default is World).

TYPE	NAME	DESCRIPTION
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

Rotate(Transform, Quaternion, Quaternion, Single, SpaceType, EaseType, LoopType, Int32, Action, Action)

Rotates the UnityEngine.Transform `target`.

Declaration

```
public void Rotate(Transform target, Quaternion from, Quaternion to, float duration, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	target	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Quaternion	from	The rotation the UnityEngine.Transform will be animated from.
UnityEngine.Quaternion	to	The rotation the UnityEngine.Transform will be animated to.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
SpaceType	spaceType	The coordinate system the tween should operate in (default is World).
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).

TYPE	NAME	DESCRIPTION
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

RotateOnAxis(Transform, Single, Single, RotateMode, out ITweenHandle, SpaceType, EaseType, LoopType, Int32)

Rotates the UnityEngine.Transform `target` around the the specified axis (RotateMode.XYZ is an invalid value and will cause an assertion).

Declaration

```
public void RotateOnAxis(Transform target, float angle, float duration, RotateMode rotateMode, out ITweenHandle tweenHandle, SpaceType spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	target	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
RotateMode	rotateMode	
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
SpaceType	spaceType	The coordinate system the tween should operate in (default is World).
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

RotateOnAxis(Transform, Single, Single, RotateMode, SpaceType, EaseType, LoopType, Int32, Action, Action)

Rotates the UnityEngine.Transform `target` around the the specified axis (RotateMode.XYZ is an invalid value and will cause an assertion).

Declaration

```
public void RotateOnAxis(Transform target, float angle, float duration, RotateMode rotateMode, SpaceType
spaceType = SpaceType.World, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int
loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	target	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
RotateMode	rotateMode	
SpaceType	spaceType	The coordinate system the tween should operate in (default is World).
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

Scale(Transform, Vector3, Vector3, Single, EaseType, LoopType, Int32, Action, Action)

Scales the UnityEngine.Transform `target`.

Declaration

```
public void Scale(Transform target, Vector3 from, Vector3 to, float duration, EaseType easeType =
EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action
onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	target	The UnityEngine.Transform that is the target of this tween.

TYPE	NAME	DESCRIPTION
UnityEngine.Vector3	from	The scale the UnityEngine.Transform will be animated from.
UnityEngine.Vector3	to	The scale the UnityEngine.Transform will be animated to.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

Scale(Transform, Vector3, Vector3, Single, out ITweenHandle, EaseType, LoopType, Int32)

Scales the UnityEngine.Transform `target`.

Declaration

```
public void Scale(Transform target, Vector3 from, Vector3 to, float duration, out ITweenHandle tweenHandle,
EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	target	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	from	The scale the UnityEngine.Transform will be animated from.
UnityEngine.Vector3	to	The scale the UnityEngine.Transform will be animated to.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.

TYPE	NAME	DESCRIPTION
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

Enum LoopType

The type of looping the tween should use while running.

Namespace: JCMG.JTween

Assembly: JCMG.JTween.dll

Syntax

```
[Serializable]
public enum LoopType : byte
```

Fields

NAME	DESCRIPTION
None	No looping.
PingPong	The tween should loop from its target state back to its original state.
Restart	The tween should loop starting back from its original state animating towards its target state.

Enum RotateMode

The mode a tween should use when animating rotation.

Namespace: JCMG.JTween

Assembly: JCMG.JTween.dll

Syntax

```
[Serializable]
public enum RotateMode : byte
```

Fields

NAME	DESCRIPTION
X	Rotates an object around the X axis.
XYZ	Rotates an object from one quaternion to another.
Y	Rotates an object around the Y axis.
Z	Rotates an object around the Z axis.

Class Singleton<T>

A basic scene Singleton implementation.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
Singleton<T>
JTweenControl

Namespace: JCMG.JTween

Assembly: JCMG.JTween.dll

Syntax

```
public abstract class Singleton<T> : MonoBehaviour where T : Component
```

Type Parameters

NAME	DESCRIPTION
T	

Properties

Exists

Returns true if the instance exists, otherwise false.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	

Instance

Returns the global `T` instance.

Declaration

```
public static T Instance { get; }
```

Property Value

TYPE	DESCRIPTION
T	

Methods

Awake()

Declaration

```
protected virtual void Awake()
```

OnApplicationQuit()

Declaration

```
protected virtual void OnApplicationQuit()
```

Enum SpaceType

The coordinate space system the tween should operate with regards to.

Namespace: JCMG.JTween

Assembly: JCMG.JTween.dll

Syntax

```
[Serializable]
public enum SpaceType : byte
```

Fields

NAME	DESCRIPTION
Local	The coordinate system with regards to the parent object.
World	The world space coordinate system.

Class TransformExtensions

Inheritance

System.Object

TransformExtensions

Namespace: JCMG.JTween

Assembly: JCMG.JTween.dll

Syntax

```
public static class TransformExtensions
```

Methods

Move(Transform, Vector3, Single, EaseType, LoopType, Int32, Action, Action)

Moves a transform in world space to position UnityEngine.Vector3 `to`.

Declaration

```
public static void Move(this Transform transform, Vector3 to, float duration, EaseType easeType =
EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action
onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	to	The position the UnityEngine.Transform will be moved to in world space.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

Move(Transform, Vector3, Single, out ITweenHandle, EaseType, LoopType, Int32)

Moves a transform in world space to position UnityEngine.Vector3 `to`.

Declaration

```
public static void Move(this Transform transform, Vector3 to, float duration, out ITweenHandle tweenHandle,
EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	to	The position the UnityEngine.Transform will be moved to in world space.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

Move(Transform, Vector3, Vector3, Single, EaseType, LoopType, Int32, Action, Action)

Moves a transform in world space from position UnityEngine.Vector3 `from` to position UnityEngine.Vector3 `to`.

Declaration

```
public static void Move(this Transform transform, Vector3 from, Vector3 to, float duration, EaseType easeType
= EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action
onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	from	The position the UnityEngine.Transform will be moved from in world space.
UnityEngine.Vector3	to	The position the UnityEngine.Transform will be moved to in world space.

TYPE	NAME	DESCRIPTION
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

Move(Transform, Vector3, Vector3, Single, out ITweenHandle, EaseType, LoopType, Int32)

Moves a transform in world space from position UnityEngine.Vector3 `from` to position UnityEngine.Vector3 `to`.

Declaration

```
public static void Move(this Transform transform, Vector3 from, Vector3 to, float duration, out ITweenHandle tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	from	The position the UnityEngine.Transform will be moved from in world space.
UnityEngine.Vector3	to	The position the UnityEngine.Transform will be moved to in world space.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).

TYPE	NAME	DESCRIPTION
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

MoveLocal(Transform, Vector3, Single, EaseType, LoopType, Int32, Action, Action)

Moves a transform in local space to position UnityEngine.Vector3 `to`.

Declaration

```
public static void MoveLocal(this Transform transform, Vector3 to, float duration, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	to	The position the UnityEngine.Transform will be moved to in local space.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

MoveLocal(Transform, Vector3, Single, out ITweenHandle, EaseType, LoopType, Int32)

Moves a transform in local space to position UnityEngine.Vector3 `to`.

Declaration

```
public static void MoveLocal(this Transform transform, Vector3 to, float duration, out ITweenHandle
tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	to	The position the UnityEngine.Transform will be moved to in local space.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

MoveLocal(Transform, Vector3, Vector3, Single, EaseType, LoopType, Int32, Action, Action)

Moves a transform in local space from position UnityEngine.Vector3 `from` to position UnityEngine.Vector3 `to`.

Declaration

```
public static void MoveLocal(this Transform transform, Vector3 from, Vector3 to, float duration, EaseType
easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null,
Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	from	The position the UnityEngine.Transform will be moved from in local space.
UnityEngine.Vector3	to	The position the UnityEngine.Transform will be moved to in local space.

TYPE	NAME	DESCRIPTION
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

MoveLocal(Transform, Vector3, Vector3, Single, out ITweenHandle, EaseType, LoopType, Int32)

Moves a transform in local space from position UnityEngine.Vector3 `from` to position UnityEngine.Vector3 `to`.

Declaration

```
public static void MoveLocal(this Transform transform, Vector3 from, Vector3 to, float duration, out
ITweenHandle tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int
loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	from	The position the UnityEngine.Transform will be moved from in local space.
UnityEngine.Vector3	to	The position the UnityEngine.Transform will be moved to in local space.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).

TYPE	NAME	DESCRIPTION
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

Rotate(Transform, Quaternion, Single, EaseType, LoopType, Int32, Action, Action)

Rotates in world space this UnityEngine.Transform to rotation UnityEngine.Quaternion `to`.

Declaration

```
public static void Rotate(this Transform transform, Quaternion to, float duration, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Quaternion	to	The rotation the UnityEngine.Transform will be animated to in world space.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

Rotate(Transform, Quaternion, Single, out ITweenHandle, EaseType, LoopType, Int32)

Rotates in world space this UnityEngine.Transform to rotation UnityEngine.Quaternion `to`.

Declaration

```
public static void Rotate(this Transform transform, Quaternion to, float duration, out ITweenHandle
tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Quaternion	to	The rotation the UnityEngine.Transform will be animated to in world space.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

Rotate(Transform, Quaternion, Quaternion, Single, EaseType, LoopType, Int32, Action, Action)

Rotates in world space this UnityEngine.Transform from rotation UnityEngine.Quaternion `from` to rotation UnityEngine.Quaternion `to`.

Declaration

```
public static void Rotate(this Transform transform, Quaternion from, Quaternion to, float duration, EaseType
easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null,
Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Quaternion	from	The rotation the UnityEngine.Transform will be animated from in world space.
UnityEngine.Quaternion	to	The rotation the UnityEngine.Transform will be animated to in world space.

TYPE	NAME	DESCRIPTION
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

Rotate(Transform, Quaternion, Quaternion, Single, out ITweenHandle, EaseType, LoopType, Int32)

Rotates in world space this UnityEngine.Transform from rotation UnityEngine.Quaternion `from` to rotation UnityEngine.Quaternion `to`.

Declaration

```
public static void Rotate(this Transform transform, Quaternion from, Quaternion to, float duration, out ITweenHandle tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Quaternion	from	The rotation the UnityEngine.Transform will be animated from in world space.
UnityEngine.Quaternion	to	The rotation the UnityEngine.Transform will be animated to in world space.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).

TYPE	NAME	DESCRIPTION
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

RotateLocal(Transform, Quaternion, Single, EaseType, LoopType, Int32, Action, Action)

Rotates in local space this UnityEngine.Transform to rotation UnityEngine.Quaternion `to`.

Declaration

```
public static void RotateLocal(this Transform transform, Quaternion to, float duration, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Quaternion	to	The rotation the UnityEngine.Transform will be animated to in local space.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

RotateLocal(Transform, Quaternion, Single, out ITweenHandle, EaseType, LoopType, Int32)

Rotates in local space this UnityEngine.Transform to rotation UnityEngine.Quaternion `to`.

Declaration

```
public static void RotateLocal(this Transform transform, Quaternion to, float duration, out ITweenHandle tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Quaternion	to	The rotation the UnityEngine.Transform will be animated to in local space.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

RotateLocal(Transform, Quaternion, Quaternion, Single, EaseType, LoopType, Int32, Action, Action)

Rotates in local space this UnityEngine.Transform from rotation UnityEngine.Quaternion `from` to rotation UnityEngine.Quaternion `to`.

Declaration

```
public static void RotateLocal(this Transform transform, Quaternion from, Quaternion to, float duration, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Quaternion	from	The rotation the UnityEngine.Transform will be animated from in local space.
UnityEngine.Quaternion	to	The rotation the UnityEngine.Transform will be animated to in local space.

TYPE	NAME	DESCRIPTION
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

RotateLocal(Transform, Quaternion, Quaternion, Single, out ITweenHandle, EaseType, LoopType, Int32)

Rotates in world space this UnityEngine.Transform from rotation UnityEngine.Quaternion `from` to rotation UnityEngine.Quaternion `to`.

Declaration

```
public static void RotateLocal(this Transform transform, Quaternion from, Quaternion to, float duration, out ITweenHandle tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Quaternion	from	The rotation the UnityEngine.Transform will be animated from in local space.
UnityEngine.Quaternion	to	The rotation the UnityEngine.Transform will be animated to in local space.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).

TYPE	NAME	DESCRIPTION
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

RotateX(Transform, Single, Single, EaseType, LoopType, Int32, Action, Action)

Rotates in world space this UnityEngine.Transform around the X axis.

Declaration

```
public static void RotateX(this Transform transform, float angle, float duration, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	The angle in degrees that this UnityEngine.Transform will be rotated in around the world X axis.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

RotateX(Transform, Single, Single, out ITweenHandle, EaseType, LoopType, Int32)

Rotates in world space this UnityEngine.Transform around the X axis.

Declaration

```
public static void RotateX(this Transform transform, float angle, float duration, out ITweenHandle
tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	The angle in degrees that this UnityEngine.Transform will be rotated in around the world X axis.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

RotateXLocal(Transform, Single, Single, EaseType, LoopType, Int32, Action, Action)

Rotates in local space this UnityEngine.Transform around the X axis.

Declaration

```
public static void RotateXLocal(this Transform transform, float angle, float duration, EaseType easeType =
EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action
onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	The angle in degrees that this UnityEngine.Transform will be rotated in around its local X axis.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).

TYPE	NAME	DESCRIPTION
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

RotateXLocal(Transform, Single, Single, out ITweenHandle, EaseType, LoopType, Int32)

Rotates in local space this UnityEngine.Transform around the X axis.

Declaration

```
public static void RotateXLocal(this Transform transform, float angle, float duration, out ITweenHandle
tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	The angle in degrees that this UnityEngine.Transform will be rotated in around its local X axis.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

RotateY(Transform, Single, Single, EaseType, LoopType, Int32, Action, Action)

Rotates in world space this UnityEngine.Transform around the Y axis.

Declaration

```
public static void RotateY(this Transform transform, float angle, float duration, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	The angle in degrees that this UnityEngine.Transform will be rotated in around the world Y axis.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

RotateY(Transform, Single, Single, out ITweenHandle, EaseType, LoopType, Int32)

Rotates in world space this UnityEngine.Transform around the Y axis.

Declaration

```
public static void RotateY(this Transform transform, float angle, float duration, out ITweenHandle tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	The angle in degrees that this UnityEngine.Transform will be rotated in around the world Y axis.

TYPE	NAME	DESCRIPTION
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

RotateYLocal(Transform, Single, Single, EaseType, LoopType, Int32, Action, Action)

Rotates in local space this UnityEngine.Transform around the Y axis.

Declaration

```
public static void RotateYLocal(this Transform transform, float angle, float duration, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	The angle in degrees that this UnityEngine.Transform will be rotated in around its local Y axis.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).

TYPE	NAME	DESCRIPTION
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

RotateYLocal(Transform, Single, Single, out ITweenHandle, EaseType, LoopType, Int32)

Rotates in local space this UnityEngine.Transform around the Y axis.

Declaration

```
public static void RotateYLocal(this Transform transform, float angle, float duration, out ITweenHandle
tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	The angle in degrees that this UnityEngine.Transform will be rotated in around its local Y axis.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

RotateZ(Transform, Single, Single, EaseType, LoopType, Int32, Action, Action)

Rotates in world space this UnityEngine.Transform around the Z axis.

Declaration

```
public static void RotateZ(this Transform transform, float angle, float duration, EaseType easeType =
EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action
onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	The angle in degrees that this UnityEngine.Transform will be rotated in around the world Z axis.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

RotateZ(Transform, Single, Single, out ITweenHandle, EaseType, LoopType, Int32)

Rotates in world space this UnityEngine.Transform around the Z axis.

Declaration

```
public static void RotateZ(this Transform transform, float angle, float duration, out ITweenHandle
tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	The angle in degrees that this UnityEngine.Transform will be rotated in around the world Z axis.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.

TYPE	NAME	DESCRIPTION
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

RotateZLocal(Transform, Single, Single, EaseType, LoopType, Int32, Action, Action)

Rotates in local space this UnityEngine.Transform around the Z axis.

Declaration

```
public static void RotateZLocal(this Transform transform, float angle, float duration, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	The angle in degrees that this UnityEngine.Transform will be rotated in around its local Z axis.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

RotateZLocal(Transform, Single, Single, out ITweenHandle, EaseType, LoopType, Int32)

Rotates in local space this UnityEngine.Transform around the Z axis.

Declaration

```
public static void RotateZLocal(this Transform transform, float angle, float duration, out ITweenHandle tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
System.Single	angle	The angle in degrees that this UnityEngine.Transform will be rotated in around its local Z axis.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

Scale(Transform, Vector3, Single, EaseType, LoopType, Int32, Action, Action)

Scales this UnityEngine.Transform to scale UnityEngine.Vector3 `to`.

Declaration

```
public static void Scale(this Transform transform, Vector3 to, float duration, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	to	The scale the UnityEngine.Transform will be animated to.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.

TYPE	NAME	DESCRIPTION
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

Scale(Transform, Vector3, Single, out ITweenHandle, EaseType, LoopType, Int32)

Scales this UnityEngine.Transform to scale UnityEngine.Vector3 `to`.

Declaration

```
public static void Scale(this Transform transform, Vector3 to, float duration, out ITweenHandle tweenHandle,
EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	to	The scale the UnityEngine.Transform will be animated to.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)

Scale(Transform, Vector3, Vector3, Single, EaseType, LoopType, Int32, Action, Action)

Scales this UnityEngine.Transform from scale UnityEngine.Vector3 `from` to scale UnityEngine.Vector3 `to`.

Declaration

```
public static void Scale(this Transform transform, Vector3 from, Vector3 to, float duration, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0, Action onStart = null, Action onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	from	The scale the UnityEngine.Transform will be animated from.
UnityEngine.Vector3	to	The scale the UnityEngine.Transform will be animated to.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)
System.Action	onStart	The System.Action that should be invoked when the tween begins playing (default is NULL).
System.Action	onComplete	The System.Action that should be invoked when the tween completes (default is NULL).

Scale(Transform, Vector3, Vector3, Single, out ITweenHandle, EaseType, LoopType, Int32)

Scales this UnityEngine.Transform from scale UnityEngine.Vector3 `from` to scale UnityEngine.Vector3 `to`.

Declaration

```
public static void Scale(this Transform transform, Vector3 from, Vector3 to, float duration, out ITweenHandle tweenHandle, EaseType easeType = EaseType.Linear, LoopType loopType = LoopType.None, int loopCount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	The UnityEngine.Transform that is the target of this tween.
UnityEngine.Vector3	from	The scale the UnityEngine.Transform will be animated from.
UnityEngine.Vector3	to	The scale the UnityEngine.Transform will be animated to.
System.Single	duration	The length of time in seconds that the tween should take to complete or one pass if looping.
ITweenHandle	tweenHandle	The ITweenHandle instance that will be initialized for this tween.
EaseType	easeType	The type of easing the tween should use while playing (default is Linear).
LoopType	loopType	The type of looping that should be used (default is None).
System.Int32	loopCount	If looping, the number of loops that should occur before completing. If set to -1, the tween will loop infinitely. (default is zero)