

Runtime Serialization

Generated by Doxygen 1.8.9.1

Mon Nov 2 2015 12:19:32

Contents

1	Namespace Index	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Namespace Documentation	7
4.1	Package UnityEngine	7
4.2	Package VoxelBusters	7
4.3	Package VoxelBusters.RuntimeSerialization	7
4.3.1	Detailed Description	8
4.3.2	Enumeration Type Documentation	8
4.3.2.1	eSaveTarget	8
5	Class Documentation	9
5.1	IRuntimeSerializable	9
5.1.1	Detailed Description	9
5.1.2	Member Function Documentation	9
5.1.2.1	WriteSerializationData	9
5.1.2.2	ReadSerializationData	9
5.2	IRuntimeSerializableActivator	10
5.2.1	Detailed Description	10
5.3	IRuntimeSerializableExtension	10
5.3.1	Detailed Description	10
5.3.2	Member Function Documentation	10
5.3.2.1	CreateInstance	10
5.3.2.2	WriteSerializationData	11
5.3.2.3	ReadSerializationData	11
5.4	IRuntimeSerializationCallback	11
5.4.1	Detailed Description	11
5.4.2	Member Function Documentation	11

5.4.2.1	OnAfterRuntimeSerialize	11
5.4.2.2	OnAfterRuntimeDeserialize	11
5.5	IRuntimeSerializationEventListener	12
5.5.1	Detailed Description	12
5.5.2	Member Function Documentation	12
5.5.2.1	OnAfterRuntimeSerialize	12
5.5.2.2	OnAfterRuntimeDeserialize	12
5.6	NonRuntimeSerializedFieldAttribute	12
5.6.1	Detailed Description	12
5.7	RSExtensionManager	12
5.7.1	Detailed Description	13
5.7.2	Member Function Documentation	13
5.7.2.1	AddNewExtension	13
5.7.2.2	RemoveExtension	13
5.7.2.3	GetExtension	13
5.8	RManager	13
5.8.1	Detailed Description	14
5.8.2	Member Function Documentation	14
5.8.2.1	Serialize< T >	14
5.8.2.2	DeserializeData< T >	15
5.8.2.3	Serialize< T >	15
5.8.2.4	Deserialize< T >	15
5.8.2.5	ContainsKey	16
5.8.2.6	GetSerializationData	16
5.8.2.7	RestoreSerializationData	16
5.8.2.8	RegisterEventListener	16
5.8.2.9	UnRegisterEventListener	16
5.8.2.10	Save	17
5.8.2.11	Remove	17
5.8.2.12	RemoveAll	17
5.8.2.13	Purge	17
5.8.2.14	GetSerializationFormatVersion	17
5.9	RSUtility	17
5.9.1	Detailed Description	18
5.9.2	Member Function Documentation	18
5.9.2.1	Instantiate	18
5.9.2.2	Instantiate	18
5.9.2.3	CreateGameObject	18
5.9.2.4	AddComponent< T >	18
5.10	RuntimeSerializableAttribute	19

5.10.1 Detailed Description	19
5.10.2 Constructor & Destructor Documentation	19
5.10.2.1 RuntimeSerializableAttribute	19
5.10.2.2 RuntimeSerializableAttribute	19
5.11 RuntimeSerializationInfo	20
5.11.1 Detailed Description	20
5.11.2 Member Function Documentation	20
5.11.2.1 AddValue< T >	20
5.11.2.2 AddValue	21
5.11.2.3 GetValue< T >	21
5.11.2.4 GetValue	21
5.11.2.5 TryGetValue< T >	21
5.11.2.6 TryGetValue	22
5.11.2.7 ContainsValue	22
5.12 RuntimeSerializeFieldAttribute	22
5.12.1 Detailed Description	22
5.13 UIDSystem	23
5.13.1 Detailed Description	23
5.13.2 Member Function Documentation	23
5.13.2.1 UpdateUIDs	23
5.13.2.2 ReassignUIDs	23
5.13.2.3 AssignUIDToNewComponent	23
5.13.2.4 UpdateCanSerializeComponentStatus	24
Index	25

Chapter 1

Namespace Index

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

IRuntimeSerializable	9
IRuntimeSerializableActivator	10
IRuntimeSerializableExtension	10
IRuntimeSerializationCallback	11
IRuntimeSerializationEventListener	12
NonRuntimeSerializedFieldAttribute	12
RSExtensionManager	12
RManager	13
RSUtility	17
RuntimeSerializableAttribute	19
RuntimeSerializationInfo	20
RuntimeSerializeFieldAttribute	22
UIDSystem	23

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

IRuntimeSerializable	Implement this interface to control serialization and deserialization of RuntimeSerializableAttribute object.	9
IRuntimeSerializableActivator	Implement this interface to control instance creation of serialized type. Static method with signature: object CreateInstance (RuntimeSerializationInfo _info) is invoked when creating serialized object instance. RuntimeSerializationInfo contains all the initializer values that were formerly serialized. So get all the initializer values from RuntimeSerializationInfo and create object instance.	10
IRuntimeSerializableExtension	Implement this abstract class to support runtime serialization for classes which belong to external assembly.	10
IRuntimeSerializationCallback	Implement this interface to receive implicit callbacks on RuntimeSerializableAttribute object. . .	11
IRuntimeSerializationEventListener	Implement this interface to observe serialization process of RuntimeSerializableAttribute object.	12
NonRuntimeSerializedFieldAttribute	Indicates that a field of a RuntimeSerializableAttribute class should not be serialized at runtime.	12
RSExtensionManager	Manages extension selection to delegate serialization and deserialization process. You can Add, Remove and Get extensions of a specific type.	12
RSManager	RSManager class is responsible for serializing and deserializing objects at runtime.	13
RSUtility	Utility class which has methods to support serialization for objects created at runtime.	17
RuntimeSerializableAttribute	Indicates that a class can be serialized at runtime.	19
RuntimeSerializationInfo	Stores all the data required to serialize or deserialize an object.	20
RuntimeSerializeFieldAttribute	Indicates that a field of a RuntimeSerializableAttribute class should be serialized at runtime. . .	22

UIDSystem

Unique Identifier System is used for uniquely identify a GameObject and Component attached to it. Please attach this component to all the GameObject that will undergo serialization and deserialization. When this component is attached to a GameObject, it recursively creates [UID↔System](#) component in all descendants and assigns UID. All the assigned UID's are cached and used for RS system for identifying an object in Scene hierarchy. Note that UID for each object, is assigned only once until and unless user forcefully reset's component. Please dont forget to enable IsPrefab flag, which is used for differentiating between normal GameObject and Prefab. UID's for Prefab are generated and assigned at runtime only.

23

Chapter 4

Namespace Documentation

4.1 Package UnityEngine

4.2 Package VoxelBusters

Namespaces

- package [RuntimeSerialization](#)

The [VoxelBusters.RuntimeSerialization](#) namespace contains classes that can be used for serializing and deserializing objects at runtime. Serialization is a process of converting an object into binary stream of data and storing it on disk. Deserialization is a process of reading stored data and recreating objects from it.

4.3 Package VoxelBusters.RuntimeSerialization

The [VoxelBusters.RuntimeSerialization](#) namespace contains classes that can be used for serializing and deserializing objects at runtime. Serialization is a process of converting an object into binary stream of data and storing it on disk. Deserialization is a process of reading stored data and recreating objects from it.

Classes

- interface [IRuntimeSerializable](#)
Implement this interface to control serialization and deserialization of [RuntimeSerializableAttribute](#) object.
- interface [IRuntimeSerializableActivator](#)
Implement this interface to control instance creation of serialized type. Static method with signature: object Create↔ Instance ([RuntimeSerializationInfo](#) _info) is invoked when creating serialized object instance. [RuntimeSerialization↔ Info](#) contains all the initializer values that were formerly serialized. So get all the initializer values from [Runtime↔ SerializationInfo](#) and create object instance.
- class [IRuntimeSerializableExtension](#)
Implement this abstract class to support runtime serialization for classes which belong to external assembly.
- interface [IRuntimeSerializationCallback](#)
Implement this interface to receive implicit callbacks on [RuntimeSerializableAttribute](#) object.
- interface [IRuntimeSerializationEventListener](#)
Implement this interface to observe serialization process of [RuntimeSerializableAttribute](#) object.
- class [NonRuntimeSerializedFieldAttribute](#)
Indicates that a field of a [RuntimeSerializableAttribute](#) class should not be serialized at runtime.
- class [RuntimeSerializableAttribute](#)
Indicates that a class can be serialized at runtime.

- class [RuntimeSerializationInfo](#)

Stores all the data required to serialize or deserialize an object.

- class [RuntimeSerializeFieldAttribute](#)

Indicates that a field of a [RuntimeSerializableAttribute](#) class should be serialized at runtime.

- class [UIDSystem](#)

Unique Identifier System is used for uniquely identify a [GameObject](#) and [Component](#) attached to it. Please attach this component to all the [GameObject](#) that will undergo serialization and deserialization. When this component is attached to a [GameObject](#), it recursively creates [UIDSystem](#) component in all descendants and assigns [UID](#). All the assigned [UID](#)'s are cached and used for RS system for identifying an object in Scene hierarchy. Note that [UID](#) for each object, is assigned only once until and unless user forcefully reset's component. Please dont forget to enable [IsPrefab](#) flag, which is used for differentiating between normal [GameObject](#) and [Prefab](#). [UID](#)'s for [Prefab](#) are generated and assigned at runtime only.

Enumerations

- enum [eSaveTarget](#) : byte {
[PLAYER_PREFS](#),
[FILE_SYSTEM](#) }

Saves serialization data to specified save target.

4.3.1 Detailed Description

The [VoxelBusters.RuntimeSerialization](#) namespace contains classes that can be used for serializing and deserializing objects at runtime. Serialization is a process of converting an object into binary stream of data and storing it on disk. Deserialization is a process of reading stored data and recreating objects from it.

The [IRuntimeSerializable](#) interface provides a way to control their own serialization behaviour. The [IRuntimeSerializableExtension](#) interface provides a way to support runtime serialization for classes which belong to external assembly.

4.3.2 Enumeration Type Documentation

4.3.2.1 enum [eSaveTarget](#) : byte

Saves serialization data to specified save target.

Enumerator

[PLAYER_PREFS](#) Saves serialization data to PlayerPrefs.

[FILE_SYSTEM](#) Saves serialization data to files. This option is not available for Unity WebPlayer.

Chapter 5

Class Documentation

5.1 IRuntimeSerializable

Implement this interface to control serialization and deserialization of [RuntimeSerializableAttribute](#) object.

Public Member Functions

- void [WriteSerializationData](#) ([RuntimeSerializationInfo](#) _info)
Populate [RuntimeSerializationInfo](#) with the properties required to serialize [RuntimeSerializableAttribute](#) object.
- object [ReadSerializationData](#) ([RuntimeSerializationInfo](#) _info)
Retrieve properties of [RuntimeSerializableAttribute](#) object from [RuntimeSerializationInfo](#).

5.1.1 Detailed Description

Implement this interface to control serialization and deserialization of [RuntimeSerializableAttribute](#) object.

5.1.2 Member Function Documentation

5.1.2.1 void WriteSerializationData (RuntimeSerializationInfo _info)

Populate [RuntimeSerializationInfo](#) with the properties required to serialize [RuntimeSerializableAttribute](#) object.

Parameters

_info	The RuntimeSerializationInfo provides interface to store properties of RuntimeSerializableAttribute object to be serialized.
-----------------------	--

5.1.2.2 object ReadSerializationData (RuntimeSerializationInfo _info)

Retrieve properties of [RuntimeSerializableAttribute](#) object from [RuntimeSerializationInfo](#).

Returns

Object retrieved from deserializing serialization data.

Parameters

<code>_info</code>	The RuntimeSerializationInfo provides interface to retrieve RuntimeSerializableAttribute object property values.
--------------------	--

5.2 IRuntimeSerializableActivator

Implement this interface to control instance creation of serialized type. Static method with signature: object CreateInstance ([RuntimeSerializationInfo](#) _info) is invoked when creating serialized object instance. [RuntimeSerializationInfo](#) contains all the initializer values that were formerly serialized. So get all the initializer values from [RuntimeSerializationInfo](#) and create object instance.

5.2.1 Detailed Description

Implement this interface to control instance creation of serialized type. Static method with signature: object CreateInstance ([RuntimeSerializationInfo](#) _info) is invoked when creating serialized object instance. [RuntimeSerializationInfo](#) contains all the initializer values that were formerly serialized. So get all the initializer values from [RuntimeSerializationInfo](#) and create object instance.

5.3 IRuntimeSerializableExtension

Implement this abstract class to support runtime serialization for classes which belong to external assembly.

Inherited by BoneWeightRSExtension, BoundsRSExtension, GuidRSExtension, HashSetRSExtension, ICollectionRSExtension, IDictionaryRSExtension, JointAngleLimits2DRSExtension, JointDriveRSExtension, JointLimitsRSExtension, JointMotor2DRSExtension, JointMotorRSExtension, JointSpringRSExtension, JointSuspension2DRSExtension, JointTranslationLimits2DRSExtension, LayerMaskRSExtension, NavMeshHitRSExtension, RectRSExtension, SoftJointLimitRSExtension, TimeSpanRSExtension, and ObjectRSExtension.

Public Member Functions

- virtual object [CreateInstance](#) ([RuntimeSerializationInfo](#) _info)
Creates the instance of formerly serialized object type.
- abstract void [WriteSerializationData](#) (object _object, [RuntimeSerializationInfo](#) _info)
Populate [RuntimeSerializationInfo](#) with the properties required to serialize target object.
- abstract object [ReadSerializationData](#) (object _object, [RuntimeSerializationInfo](#) _info)
Retrieve serialized properties of target object from [RuntimeSerializationInfo](#).

5.3.1 Detailed Description

Implement this abstract class to support runtime serialization for classes which belong to external assembly.

5.3.2 Member Function Documentation

5.3.2.1 virtual object CreateInstance ([RuntimeSerializationInfo](#) _info) [virtual]

Creates the instance of formerly serialized object type.

Returns

Serialized object instance.

Parameters

<code>_info</code>	The RuntimeSerializationInfo provides the interface to access all the previously serialized initializer values.
--------------------	---

5.3.2.2 `abstract void WriteSerializationData (object _object, RuntimeSerializationInfo _info) [pure virtual]`

Populate [RuntimeSerializationInfo](#) with the properties required to serialize target object.

Parameters

<code>_object</code>	Target object to be serialized
<code>_info</code>	The RuntimeSerializationInfo provides interface to add properties to be serialized.

5.3.2.3 `abstract object ReadSerializationData (object _object, RuntimeSerializationInfo _info) [pure virtual]`

Retrieve serialized properties of target object from [RuntimeSerializationInfo](#).

Returns

The deserialized object.

Parameters

<code>_object</code>	Target object to be deserialized.
<code>_info</code>	The RuntimeSerializationInfo provides interface to retrieve serialized values.

5.4 IRuntimeSerializationCallback

Implement this interface to receive implicit callbacks on [RuntimeSerializableAttribute](#) object.

Public Member Functions

- void [OnAfterRuntimeSerialize](#) ()
Event triggered after [RuntimeSerializableAttribute](#) object is serialized.
- void [OnAfterRuntimeDeserialize](#) ()
Event triggered after [RuntimeSerializableAttribute](#) object is deserialized.

5.4.1 Detailed Description

Implement this interface to receive implicit callbacks on [RuntimeSerializableAttribute](#) object.

5.4.2 Member Function Documentation

5.4.2.1 `void OnAfterRuntimeSerialize ()`

Event triggered after [RuntimeSerializableAttribute](#) object is serialized.

5.4.2.2 `void OnAfterRuntimeDeserialize ()`

Event triggered after [RuntimeSerializableAttribute](#) object is deserialized.

5.5 IRuntimeSerializationEventListener

Implement this interface to observe serialization process of [RuntimeSerializableAttribute](#) object.

Public Member Functions

- void [OnAfterRuntimeSerialize](#) (string _key, object _object)
Event triggered after [RuntimeSerializableAttribute](#) object serialization is completed.
- void [OnAfterRuntimeDeserialize](#) (string _key, object _object)
Event triggered after [RuntimeSerializableAttribute](#) object deserialization is completed.

5.5.1 Detailed Description

Implement this interface to observe serialization process of [RuntimeSerializableAttribute](#) object.

5.5.2 Member Function Documentation

5.5.2.1 void OnAfterRuntimeSerialize (string _key, object _object)

Event triggered after [RuntimeSerializableAttribute](#) object serialization is completed.

Parameters

<code>_key</code>	A key string used to identify object's serialization.
<code>_object</code>	The object which was just serialized.

5.5.2.2 void OnAfterRuntimeDeserialize (string _key, object _object)

Event triggered after [RuntimeSerializableAttribute](#) object deserialization is completed.

Parameters

<code>_key</code>	A key string used to identify object's deserialization.
<code>_object</code>	The object retrieved from deserializing serialization data.

5.6 NonRuntimeSerializedFieldAttribute

Indicates that a field of a [RuntimeSerializableAttribute](#) class should not be serialized at runtime.

Inherits [Attribute](#).

5.6.1 Detailed Description

Indicates that a field of a [RuntimeSerializableAttribute](#) class should not be serialized at runtime.

5.7 RSExtensionManager

Manages extension selection to delegate serialization and deserialization process. You can Add, Remove and Get extensions of a specific type.

Static Public Member Functions

- static void [AddNewExtension](#) (Type _objectType, object _extension)
Adds the new serialization extension to its collection. If extension information is manually provided, then serialization system avoids using Reflection for finding type's extension at runtime.
- static void [RemoveExtension](#) (Type _objectType)
Removes the extension from its collections.
- static RSExtension [GetExtension](#) (Type _objectType)
Returns the extension for a particular type.

5.7.1 Detailed Description

Manages extension selection to delegate serialization and deserialization process. You can Add, Remove and Get extensions of a specific type.

5.7.2 Member Function Documentation

5.7.2.1 static void AddNewExtension (Type _objectType, object _extension) [static]

Adds the new serialization extension to its collection. If extension information is manually provided, then serialization system avoids using Reflection for finding type's extension at runtime.

Parameters

<code>_objectType</code>	The Type for which the extension is required.
<code>_extensionType</code>	The object called while serializing and deserializing.

5.7.2.2 static void RemoveExtension (Type _objectType) [static]

Removes the extension from its collections.

Parameters

<code>_objectType</code>	The Type for which the extension has to be removed.
--------------------------	---

5.7.2.3 static RSExtension GetExtension (Type _objectType) [static]

Returns the extension for a particular type.

Returns

The extension for a particular type.

Parameters

<code>_objectType</code>	The type for which extension is requested.
--------------------------	--

5.8 RManager

[RManager](#) class is responsible for serializing and deserializing objects at runtime.

Inherits SingletonPattern< RManager >.

Public Member Functions

- int [GetSerializationFormatVersion](#) ()
Gets the serialization format version.

Static Public Member Functions

- static string [Serialize](#)< T > (T _object, string _key=null)
Returns serialization data of target object as Base64 string. After serialization, associated serialization data doesn't get saved by . And it is user's responsibility to provide this data while deserializing object using method .
- static T [DeserializeData](#)< T > (string _serializationDataString, string _key=null, T _targetObject=default(T))
Deserializes the serialization data and recreates the object of specified type.
- static void [Serialize](#)< T > (T _object, string _key, [eSaveTarget](#) _saveTarget)
Serialize the specified object and then save its serialization data to mentioned target location. Also given key value is used for identifying objects serialization data.
- static T [Deserialize](#)< T > (string _key, T _targetObject=default(T))
Deserializes the data serialization associated with given key and recreates the object of specified type.
- static bool [ContainsKey](#) (string _key)
Determines whether [RSManger](#) contains serialization data for specified key.
- static string [GetSerializationData](#) (string _key)
Returns serialization data associated with specified key. Ideal for supporting multi device login, wherein you can save serialization data remotely and restore it on other devices using method [RSManger.RestoreSerializationData](#).
- static bool [RestoreSerializationData](#) (string _serializationDataString, string _key, [eSaveTarget](#) _saveTarget)
Saves Base64String format serialization data to specified target location where it is associated with specified key.
- static void [RegisterEventListener](#) (string _key, [IRuntimeSerializationEventListener](#) _newListener)
Register object to receive serialization events.
- static void [UnRegisterEventListener](#) (string _key, [IRuntimeSerializationEventListener](#) _listener)
Unregister object from receiving serialization events.
- static void [Save](#) ()
Writes all serialization data to disk. By default [RSManger](#) writes serialization data to PlayerPrefs/File on Application Pause and on Application Quit.
- static void [Remove](#) (string _key)
Removes serialization data associated with specified key.
- static void [RemoveAll](#) ()
Clears all existing serialization data.
- static void [Purge](#) (Type _objectType)
Clears all the cached information of specified object type.

5.8.1 Detailed Description

[RSManger](#) class is responsible for serializing and deserializing objects at runtime.

5.8.2 Member Function Documentation

5.8.2.1 static string [Serialize](#)< T > (T _object, string _key = null) [static]

Returns serialization data of target object as Base64 string. After serialization, associated serialization data doesn't get saved by . And it is user's responsibility to provide this data while deserializing object using method .

Parameters

<code>_object</code>	The object to be serialized.
<code>_key</code>	A key string used to identify object's serialization. An optional parameter, when supplied it is used for firing serialization finished callback.

Template Parameters

<code>T</code>	The type of the object being serialized.
----------------	--

5.8.2.2 `static T DeserializeData< T > (string _serializationDataString, string _key = null, T _targetObject = default(T)) [static]`

Deserializes the serialization data and recreates the object of specified type.

Returns

The deserialized object of specified type.

Parameters

<code>_serializationDataString</code>	Serialization data in Base64String format.
<code>_key</code>	A key string used to identify object's deserialization. An optional parameter, when supplied it is used for firing deserialization finished callback.
<code>_targetObject</code>	If value is non-null, then all the properties are deserialized back to this object.

Template Parameters

<code>T</code>	The type of the object returned after it is deserialized.
----------------	---

5.8.2.3 `static void Serialize< T > (T _object, string _key, eSaveTarget _saveTarget) [static]`

Serialize the specified object and then save its serialization data to mentioned target location. Also given key value is used for identifying objects serialization data.

Parameters

<code>_object</code>	The object to be serialized.
<code>_key</code>	A key string used to identify object's serialization.
<code>_saveTarget</code>	Preferred target where serialization data is saved.

Template Parameters

<code>T</code>	The type of the object being serialized.
----------------	--

5.8.2.4 `static T Deserialize< T > (string _key, T _targetObject = default(T)) [static]`

Deserializes the data serialization associated with given key and recreates the object of specified type.

Parameters

<code>_key</code>	A key string used to identify object's deserialization.
<code>_targetObject</code>	If value is non-null, then all the properties are deserialized back to this object.

Template Parameters

<i>T</i>	The type of the object returned after it is deserialized.
----------	---

5.8.2.5 static bool ContainsKey (string _key) [static]

Determines whether [RManager](#) contains serialization data for specified key.

Returns

true, if serialization data was found, false otherwise.

Parameters

<i>_key</i>	A key string used to identify object's serialization data.
-------------	--

5.8.2.6 static string GetSerializationData (string _key) [static]

Returns serialization data associated with specified key. Ideal for supporting multi device login, wherein you can save serialization data remotely and restore it on other devices using method [RManager.RestoreSerializationData](#).

Returns

Serialization data as Base64String.

Parameters

<i>_key</i>	A key string used to identify object's serialization data.
-------------	--

5.8.2.7 static bool RestoreSerializationData (string _serializationDataString, string _key, eSaveTarget _saveTarget) [static]

Saves Base64String format serialization data to specified target location where it is associated with specified key.

Parameters

<i>_serializationDataString</i>	Serialization data in Base64String format.
<i>_key</i>	A key string used to identify object's serialization data.
<i>_saveTarget</i>	Serialization data save target.

5.8.2.8 static void RegisterEventListener (string _key, IRuntimeSerializationEventListener _newListener) [static]

Register object to receive serialization events.

Parameters

<i>_key</i>	A key string used to identify object serialization and deserialization.
<i>_newListener</i>	Register listener object for the events associated with given key.

5.8.2.9 static void UnRegisterEventListener (string _key, IRuntimeSerializationEventListener _listener) [static]

Unregister object from receiving serialization events.

Parameters

<code>_key</code>	A key string used to identify object serialization and deserialization.
<code>_newListener</code>	Unregister listener object from the events associated with given key.

5.8.2.10 `static void Save () [static]`

Writes all serialization data to disk. By default [RSManager](#) writes serialization data to PlayerPrefs/File on Application Pause and on Application Quit.

5.8.2.11 `static void Remove (string _key) [static]`

Removes serialization data associated with specified key.

Parameters

<code>_key</code>	A key string used to identify object's serialization data.
-------------------	--

5.8.2.12 `static void RemoveAll () [static]`

Clears all existing serialization data.

5.8.2.13 `static void Purge (Type _objectType) [static]`

Clears all the cached information of specified object type.

5.8.2.14 `int GetSerializationFormatVersion ()`

Gets the serialization format version.

Returns

The serialization format version.

5.9 RSUtility

Utility class which has methods to support serialization for objects created at runtime.

Static Public Member Functions

- `static GameObject Instantiate (GameObject _gameobject)`
Creates an exact copy of original game object.
- `static GameObject Instantiate (GameObject _gameobject, Vector3 _position, Quaternion _rotation)`
Creates an exact copy of original game object.
- `static GameObject CreateGameObject (string _name, params System.Type[] _components)`
Creates a new game object with specified name and attaches the specified components.
- `static T AddComponent< T > (GameObject _gameObject)`
Adds a component class to the game object.

5.9.1 Detailed Description

Utility class which has methods to support serialization for objects created at runtime.

5.9.2 Member Function Documentation

5.9.2.1 static GameObject Instantiate (GameObject *_gameObject*) [static]

Creates an exact copy of original game object.

When you clone a game object, all child objects and components will also be cloned with their properties set like those of the original object. Additionally, UIDSystem component will be attached to every game object (only if it doesn't exist). Also new set of unique identifiers are assigned for each game object and its components.

Parameters

<i>_gameObject</i>	The object that you want to clone.
--------------------	------------------------------------

5.9.2.2 static GameObject Instantiate (GameObject *_gameObject*, Vector3 *_position*, Quaternion *_rotation*) [static]

Creates an exact copy of original game object.

When you clone a game object, all child objects and components will also be cloned with their properties set like those of the original object. Additionally, UIDSystem component will be attached to every game object (only if it doesn't exist). Also new set of unique identifiers are assigned for each game object and its components.

Parameters

<i>_gameObject</i>	The object that you want to clone.
<i>_position</i>	Position for the new object.
<i>_rotation</i>	Orientation of the new object.

5.9.2.3 static GameObject CreateGameObject (string *_name*, params System.Type[] *_components*) [static]

Creates a new game object with specified name and attaches the specified components.

Additionally, UIDSystem component is attached and along with it, new set of unique identifiers are assigned to the game object as well as its components.

Returns

The newly created game object.

Parameters

<i>_name</i>	The string value used for naming this game object.
<i>_components</i>	Type of components that needs to be attached to this newly created game object.

5.9.2.4 static T AddComponent< T > (GameObject *_gameObject*) [static]

Adds a component class to the game object.

Returns

The newly created component instance.

Parameters

<code>_gameObject</code>	The game object to which this new component has to be added.
--------------------------	--

Template Parameters

<code>T</code>	Type of the component to be added.
----------------	------------------------------------

Type Constraints

***T* : `Component`**

5.10 `RuntimeSerializableAttribute`

Indicates that a class can be serialized at runtime.

Inherits `Attribute`.

Public Member Functions

- [`RuntimeSerializableAttribute`](#) (`bool` `_serializeAllPublicVariables`, `bool` `_serializeAllNonPublicVariables`)
Initializes a new instance of the [`RuntimeSerializableAttribute`](#) class.
- [`RuntimeSerializableAttribute`](#) (`Type` `_extensionDependencyObjectType`=`null`, `bool` `_serializeAllPublicVariables`=`true`, `bool` `_serializeAllNonPublicVariables`=`false`)
Initializes a new instance of the [`RuntimeSerializableAttribute`](#) class.

5.10.1 Detailed Description

Indicates that a class can be serialized at runtime.

5.10.2 Constructor & Destructor Documentation

5.10.2.1 `RuntimeSerializableAttribute` (`bool` `_serializeAllPublicVariables`, `bool` `_serializeAllNonPublicVariables`)

Initializes a new instance of the [`RuntimeSerializableAttribute`](#) class.

Parameters

<code>_serializeAllPublicVariables</code>	If set to <code>true</code> all public variables will be serialized at runtime.
<code>_serializeAllNonPublicVariables</code>	If set to <code>true</code> all non public variables will be serialized at runtime.

5.10.2.2 `RuntimeSerializableAttribute` (`Type` `_extensionDependencyObjectType` = `null`, `bool` `_serializeAllPublicVariables` = `true`, `bool` `_serializeAllNonPublicVariables` = `false`)

Initializes a new instance of the [`RuntimeSerializableAttribute`](#) class.

Parameters

<code>_extension↔ Dependency↔ ObjectType</code>	Its possible that class might be deriving properties from a class which belongs to an external library. So at that point, this property can be used to create a virtual link between derived class and serialization extension of parent class to ensure that properties are properly serialized.
<code>_serializeAll↔ PublicVariables</code>	If set to <code>true</code> all public variables will be serialized at runtime.
<code>_serializeAll↔ NonPublic↔ Variables</code>	If set to <code>true</code> all non public variables will be serialized at runtime.

5.11 RuntimeSerializationInfo

Stores all the data required to serialize or deserialize an object.

Public Member Functions

- void [AddValue< T >](#) (string _name, T _value, bool _isObjectInitializer=false)
Adds the specified object into the [RuntimeSerializationInfo](#) for serialization, where it is associated with name.
- void [AddValue](#) (string _name, object _value, Type _valueType, bool _isObjectInitializer=false)
Adds the specified object into the [RuntimeSerializationInfo](#) for serialization, where it is associated with name.
- T [GetValue< T >](#) (string _name, bool _isObjectInitializer=false)
Gets the value from [RuntimeSerializationInfo](#) using specified name.
- object [GetValue](#) (string _name, Type _type, bool _isObjectInitializer=false)
Gets the value from [RuntimeSerializationInfo](#) using specified name.
- bool [TryGetValue< T >](#) (string _name, out T _value, bool _isObjectInitializer=false)
Gets the value from [RuntimeSerializationInfo](#) using specified name.
- bool [TryGetValue](#) (string _name, out object _value, Type _type, bool _isObjectInitializer=false)
Gets the value from [RuntimeSerializationInfo](#) using specified name.
- bool [ContainsValue](#) (string _name, bool _isObjectInitializer=false)
Determines whether the [RuntimeSerializationInfo](#) contains a specific value.

5.11.1 Detailed Description

Stores all the data required to serialize or deserialize an object.

5.11.2 Member Function Documentation

5.11.2.1 void AddValue< T > (string _name, T _value, bool _isObjectInitializer = false)

Adds the specified object into the [RuntimeSerializationInfo](#) for serialization, where it is associated with name.

Parameters

<code>_name</code>	The name associated with the value to be stored in RuntimeSerializationInfo .
<code>_value</code>	The value to be serialized. Any children of this object will automatically be serialized.
<code>_isObject↔ Initializer</code>	The flag indicates whether this value is object initializer. Object initializers are the values that are available while creating object instance.

Template Parameters

<i>T</i>	The Type associated with the current object. This must always be the type of the object itself.
----------	---

5.11.2.2 void AddValue (string _name, object _value, Type _valueType, bool _isObjectInitializer = false)

Adds the specified object into the [RuntimeSerializationInfo](#) for serialization, where it is associated with name.

Parameters

<i>_name</i>	The name associated with the value to be stored in RuntimeSerializationInfo .
<i>_value</i>	The value to be serialized. Any children of this object will automatically be serialized.
<i>_valueType</i>	The Type associated with the current object. This must always be the type of the object itself.
<i>_isObjectInitializer</i>	The flag indicates whether this value is object initializer. Object initializers are the values that are available while creating object instance.

5.11.2.3 T GetValue< T > (string _name, bool _isObjectInitializer = false)

Gets the value from [RuntimeSerializationInfo](#) using specified name.

Returns

The object of specified type associated with name.

Parameters

<i>_name</i>	The name associated with the value to be retrieved from RuntimeSerializationInfo .
<i>_isObjectInitializer</i>	The flag indicates whether this value is object initializer. Object initializers are the values that are available while creating object instance.

Template Parameters

<i>T</i>	The Type of the value to retrieve.
----------	------------------------------------

5.11.2.4 object GetValue (string _name, Type _type, bool _isObjectInitializer = false)

Gets the value from [RuntimeSerializationInfo](#) using specified name.

Returns

The object of specified type associated with name.

Parameters

<i>_name</i>	The name associated with the value to be retrieved from RuntimeSerializationInfo .
--------------	--

<param name="_type">The Type of the value to retrieve.

Parameters

<i>_isObjectInitializer</i>	The flag indicates whether this value is object initializer. Object initializers are the values that are available while creating object instance.
-----------------------------	--

5.11.2.5 bool TryGetValue< T > (string _name, out T _value, bool _isObjectInitializer = false)

Gets the value from [RuntimeSerializationInfo](#) using specified name.

Returns

`true`, if value was found in [RuntimeSerializationInfo](#), `false` otherwise.

Parameters

<code>_name</code>	The name associated with the value to be retrieved from RuntimeSerializationInfo .
<code>_value</code>	The value associated with the specified name.
<code>_isObjectInitializer</code>	The flag indicates whether this value is object initializer. Object initializers are the values that are available while creating object instance.

Template Parameters

<code>T</code>	The Type of the value to retrieve.
----------------	------------------------------------

5.11.2.6 bool TryGetValue (string _name, out object _value, Type _type, bool _isObjectInitializer = false)

Gets the value from [RuntimeSerializationInfo](#) using specified name.

Returns

`true`, if value was found in [RuntimeSerializationInfo](#), `false` otherwise.

Parameters

<code>_name</code>	The name associated with the value to be fetched from RuntimeSerializationInfo .
<code>_value</code>	The value associated with the specified name.

<param name="_type">The Type of the value to be fetched.

Parameters

<code>_isObjectInitializer</code>	The flag indicates whether this value is object initializer. Object initializers are the values that are available while creating object instance.
-----------------------------------	--

5.11.2.7 bool ContainsValue (string _name, bool _isObjectInitializer = false)

Determines whether the [RuntimeSerializationInfo](#) contains a specific value.

Returns

`true`, if value with specified name exists in [RuntimeSerializationInfo](#), `false` otherwise.

Parameters

<code>_name</code>	The name associated with the value to be fetched from RuntimeSerializationInfo .
<code>_isObjectInitializer</code>	The flag indicates whether this value is object initializer. Object initializers are the values that are available while creating object instance.

5.12 RuntimeSerializeFieldAttribute

Indicates that a field of a [RuntimeSerializableAttribute](#) class should be serialized at runtime.

Inherits Attribute.

5.12.1 Detailed Description

Indicates that a field of a [RuntimeSerializableAttribute](#) class should be serialized at runtime.

5.13 UIDSystem

Unique Identifier System is used for uniquely identify a GameObject and Component attached to it. Please attach this component to all the GameObject that will undergo serialization and deserialization. When this component is attached to a GameObject, it recursively creates [UIDSystem](#) component in all descendants and assigns UID. All the assigned UID's are cached and used for RS system for identifying an object in Scene hierarchy. Note that UID for each object, is assigned only once until and unless user forcefully reset's component. Please dont forget to enable IsPrefab flag, which is used for differentiating between normal GameObject and Prefab. UID's for Prefab are generated and assigned at runtime only.

Inherits MonoBehaviour, MonoBehaviour, and MonoBehaviour.

Public Member Functions

- void [UpdateUIDs](#) (bool _recursive)
Assigns new UID for all the new Components attached to this GameObject.
- void [ReassignUIDs](#) (bool _recursive)
Flushes all the existing UID's and reassigns new UID's to this GameObject and its components.
- void [AssignUIDToNewComponent](#) (Component _component)
Generates and assigns a new UID to the given component. Operation fails, if UID was already assigned to this component.
- void [UpdateCanSerializeComponentStatus](#) (Component _component, bool _canSerialize)
Mark the components you would like to serialize while serializing the game object.

5.13.1 Detailed Description

Unique Identifier System is used for uniquely identify a GameObject and Component attached to it. Please attach this component to all the GameObject that will undergo serialization and deserialization. When this component is attached to a GameObject, it recursively creates [UIDSystem](#) component in all descendants and assigns UID. All the assigned UID's are cached and used for RS system for identifying an object in Scene hierarchy. Note that UID for each object, is assigned only once until and unless user forcefully reset's component. Please dont forget to enable IsPrefab flag, which is used for differentiating between normal GameObject and Prefab. UID's for Prefab are generated and assigned at runtime only.

5.13.2 Member Function Documentation

5.13.2.1 void UpdateUIDs (bool _recursive)

Assigns new UID for all the new Components attached to this GameObject.

5.13.2.2 void ReassignUIDs (bool _recursive)

Flushes all the existing UID's and reassigns new UID's to this GameObject and its components.

5.13.2.3 void AssignUIDToNewComponent (Component _component)

Generates and assigns a new UID to the given component. Operation fails, if UID was already assigned to this component.

Parameters

<code>_component</code>	Component to which UID needs to be assigned.
-------------------------	--

5.13.2.4 void UpdateCanSerializeComponentStatus (Component *_component*, bool *_canSerialize*)

Mark the components you would like to serialize while serializing the game object.

Parameters

<code>_component</code>	Component attached to this gameobject.
<code>_canSerialize</code>	If set to <code>true</code> then component is serialized.

Index

- AddComponent< T >
 - RSUtility, [18](#)
- AddNewExtension
 - RSExtensionManager, [13](#)
- AddValue
 - VoxelBusters::RuntimeSerialization::Runtime↔
SerializationInfo, [21](#)
- AddValue< T >
 - VoxelBusters::RuntimeSerialization::Runtime↔
SerializationInfo, [20](#)
- AssignUIDToNewComponent
 - VoxelBusters::RuntimeSerialization::UIDSystem,
[23](#)
- ContainsKey
 - RManager, [16](#)
- ContainsValue
 - VoxelBusters::RuntimeSerialization::Runtime↔
SerializationInfo, [22](#)
- CreateGameObject
 - RSUtility, [18](#)
- CreateInstance
 - VoxelBusters::RuntimeSerialization::IRuntime↔
SerializableExtension, [10](#)
- Deserialize< T >
 - RManager, [15](#)
- DeserializeData< T >
 - RManager, [15](#)
- eSaveTarget
 - VoxelBusters::RuntimeSerialization, [8](#)
- FILE_SYSTEM
 - VoxelBusters::RuntimeSerialization, [8](#)
- GetExtension
 - RSExtensionManager, [13](#)
- GetSerializationData
 - RManager, [16](#)
- GetSerializationFormatVersion
 - RManager, [17](#)
- GetValue
 - VoxelBusters::RuntimeSerialization::Runtime↔
SerializationInfo, [21](#)
- GetValue< T >
 - VoxelBusters::RuntimeSerialization::Runtime↔
SerializationInfo, [21](#)
- IRuntimeSerializable, [9](#)
- IRuntimeSerializableActivator, [10](#)
- IRuntimeSerializableExtension, [10](#)
- IRuntimeSerializationCallback, [11](#)
- IRuntimeSerializationEventListener, [12](#)
- Instantiate
 - RSUtility, [18](#)
- NonRuntimeSerializedFieldAttribute, [12](#)
- OnAfterRuntimeDeserialize
 - VoxelBusters::RuntimeSerialization::IRuntime↔
SerializationCallback, [11](#)
 - VoxelBusters::RuntimeSerialization::IRuntime↔
SerializationEventListener, [12](#)
- OnAfterRuntimeSerialize
 - VoxelBusters::RuntimeSerialization::IRuntime↔
SerializationCallback, [11](#)
 - VoxelBusters::RuntimeSerialization::IRuntime↔
SerializationEventListener, [12](#)
- PLAYER_PREFS
 - VoxelBusters::RuntimeSerialization, [8](#)
- Purge
 - RManager, [17](#)
- RSExtensionManager, [12](#)
 - AddNewExtension, [13](#)
 - GetExtension, [13](#)
 - RemoveExtension, [13](#)
- RManager, [13](#)
 - ContainsKey, [16](#)
 - Deserialize< T >, [15](#)
 - DeserializeData< T >, [15](#)
 - GetSerializationData, [16](#)
 - GetSerializationFormatVersion, [17](#)
 - Purge, [17](#)
 - RegisterEventListener, [16](#)
 - Remove, [17](#)
 - RemoveAll, [17](#)
 - RestoreSerializationData, [16](#)
 - Save, [17](#)
 - Serialize< T >, [14](#), [15](#)
 - UnRegisterEventListener, [16](#)
- RSUtility, [17](#)
 - AddComponent< T >, [18](#)
 - CreateGameObject, [18](#)
 - Instantiate, [18](#)
- ReadSerializationData
 - VoxelBusters::RuntimeSerialization::IRuntime↔
Serializable, [9](#)
 - VoxelBusters::RuntimeSerialization::IRuntime↔
SerializableExtension, [11](#)

- ReassignUIDs
 - VoxelBusters::RuntimeSerialization::UIDSystem, [23](#)
- RegisterEventListener
 - RManager, [16](#)
- Remove
 - RManager, [17](#)
- RemoveAll
 - RManager, [17](#)
- RemoveExtension
 - RSExtensionManager, [13](#)
- RestoreSerializationData
 - RManager, [16](#)
- RuntimeSerializableAttribute, [19](#)
 - VoxelBusters::RuntimeSerialization::Runtime↔SerializableAttribute, [19](#)
- RuntimeSerializationInfo, [20](#)
- RuntimeSerializeFieldAttribute, [22](#)
- Save
 - RManager, [17](#)
- Serialize< T >
 - RManager, [14](#), [15](#)
- TryGetValue
 - VoxelBusters::RuntimeSerialization::Runtime↔SerializationInfo, [22](#)
- TryGetValue< T >
 - VoxelBusters::RuntimeSerialization::Runtime↔SerializationInfo, [21](#)
- UIDSystem, [23](#)
- UnRegisterEventListener
 - RManager, [16](#)
- UnityEngine, [7](#)
- UpdateCanSerializeComponentStatus
 - VoxelBusters::RuntimeSerialization::UIDSystem, [24](#)
- UpdateUIDs
 - VoxelBusters::RuntimeSerialization::UIDSystem, [23](#)
- VoxelBusters, [7](#)
- VoxelBusters.RuntimeSerialization, [7](#)
- VoxelBusters::RuntimeSerialization
 - eSaveTarget, [8](#)
 - FILE_SYSTEM, [8](#)
 - PLAYER_PREFS, [8](#)
- VoxelBusters::RuntimeSerialization::IRuntimeSerializable
 - ReadSerializationData, [9](#)
 - WriteSerializationData, [9](#)
- VoxelBusters::RuntimeSerialization::IRuntimeSerializable↔Extension
 - CreateInstance, [10](#)
 - ReadSerializationData, [11](#)
 - WriteSerializationData, [11](#)
- VoxelBusters::RuntimeSerialization::IRuntimeSerialization↔Callback
 - OnAfterRuntimeDeserialize, [11](#)
 - OnAfterRuntimeSerialize, [11](#)
- VoxelBusters::RuntimeSerialization::IRuntimeSerialization↔EventListener
 - OnAfterRuntimeDeserialize, [12](#)
 - OnAfterRuntimeSerialize, [12](#)
- VoxelBusters::RuntimeSerialization::RuntimeSerializable↔Attribute
 - RuntimeSerializableAttribute, [19](#)
- VoxelBusters::RuntimeSerialization::RuntimeSerialization↔Info
 - AddValue, [21](#)
 - AddValue< T >, [20](#)
 - ContainsValue, [22](#)
 - GetValue, [21](#)
 - GetValue< T >, [21](#)
 - TryGetValue, [22](#)
 - TryGetValue< T >, [21](#)
- VoxelBusters::RuntimeSerialization::UIDSystem
 - AssignUIDToNewComponent, [23](#)
 - ReassignUIDs, [23](#)
 - UpdateCanSerializeComponentStatus, [24](#)
 - UpdateUIDs, [23](#)
- WriteSerializationData
 - VoxelBusters::RuntimeSerialization::IRuntime↔Serializable, [9](#)
 - VoxelBusters::RuntimeSerialization::IRuntime↔SerializableExtension, [11](#)