

Cross Platform Easy Save

Generated by Doxygen 1.8.15

1 Namespace Index	1
1.1 Packages	1
2 Class Index	3
2.1 Class List	3
3 Namespace Documentation	5
3.1 VoxelBusters Namespace Reference	5
3.1.1 Detailed Description	5
3.2 VoxelBusters.Serialization Namespace Reference	5
4 Class Documentation	7
4.1 VoxelBusters.Serialization.SerializationManager Class Reference	7
4.1.1 Detailed Description	8
4.1.2 Member Function Documentation	8
4.1.2.1 BeginDeserializeGroup()	8
4.1.2.2 BeginDeserializeToByteArrayGroup() [1/2]	9
4.1.2.3 BeginDeserializeToByteArrayGroup() [2/2]	9
4.1.2.4 BeginSerializeGroup()	10
4.1.2.5 BeginSerializeToByteArrayGroup() [1/2]	10
4.1.2.6 BeginSerializeToByteArrayGroup() [2/2]	11
4.1.2.7 BufferSize()	11
4.1.2.8 ClearCache()	11
4.1.2.9 DeleteAll()	11
4.1.2.10 DeleteKey()	11
4.1.2.11 Deserialize< T >()	12
4.1.2.12 DeserializeFromByteArray< T >() [1/2]	13
4.1.2.13 DeserializeFromByteArray< T >() [2/2]	13
4.1.2.14 DeserializeInt32()	14
4.1.2.15 DeserializeSingle()	15
4.1.2.16 DeserializeString()	15
4.1.2.17 EndDeserializeGroup()	16
4.1.2.18 EndDeserializeToByteArrayGroup()	16
4.1.2.19 EndSerializeGroup()	16
4.1.2.20 EndSerializeToByteArrayGroup()	17
4.1.2.21 HasKey()	17
4.1.2.22 SerializationMethod()	17
4.1.2.23 Serialize< T >()	17
4.1.2.24 SerializeInt32()	18
4.1.2.25 SerializeSingle()	19
4.1.2.26 SerializeString()	19
4.1.2.27 SerializeToByteArray< T >() [1/2]	20
4.1.2.28 SerializeToByteArray< T >() [2/2]	21

4.1.2.29 StorageTarget()	21
4.2 VoxelBusters.Serialization.SerializationUtility Class Reference	22
4.2.1 Detailed Description	22
4.2.2 Member Function Documentation	22
4.2.2.1 AddComponent()	22
4.2.2.2 AddComponent< T >()	23
4.2.2.3 CreateGameObject() [1/3]	23
4.2.2.4 CreateGameObject() [2/3]	23
4.2.2.5 CreateGameObject() [3/3]	24
4.2.2.6 Destroy() [1/2]	24
4.2.2.7 Destroy() [2/2]	24
4.2.2.8 Instantiate() [1/4]	25
4.2.2.9 Instantiate() [2/4]	25
4.2.2.10 Instantiate() [3/4]	25
4.2.2.11 Instantiate() [4/4]	26
Index	27

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

VoxelBusters	
VoxelBusters.Serialization module implements classes that can be used for saving and restoring object values between game session	5
VoxelBusters.Serialization	5

Chapter 2

Class Index

2.1 Class List

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

[VoxelBusters.Serialization.SerializationManager](#)

The [SerializationManager](#) class is the interface for saving and restoring object values between game sessions [7](#)

[VoxelBusters.Serialization.SerializationUtility](#)

Utility functions for implementing and extending the serialization support for Unity objects created at runtime [22](#)

Chapter 3

Namespace Documentation

3.1 VoxelBusters Namespace Reference

[VoxelBusters.Serialization](#) module implements classes that can be used for saving and restoring object values between game session.

3.1.1 Detailed Description

[VoxelBusters.Serialization](#) module implements classes that can be used for saving and restoring object values between game session.

3.2 VoxelBusters.Serialization Namespace Reference

Classes

- class [SerializationManager](#)
The [SerializationManager](#) class is the interface for saving and restoring object values between game sessions.
- class [SerializationUtility](#)
Utility functions for implementing and extending the serialization support for Unity objects created at runtime.

Chapter 4

Class Documentation

4.1 VoxBusters.Serialization.SerializationManager Class Reference

The [SerializationManager](#) class is the interface for saving and restoring object values between game sessions.

Static Public Member Functions

- static void [BeginSerializeGroup](#) (string key, params SerializationOption[] options)
Begins the serialize group.
- static void [SerializeInt32](#) (string key, int value, params SerializationOption[] options)
Saves the given integer value.
- static void [SerializeSingle](#) (string key, float value, params SerializationOption[] options)
Saves the given float value.
- static void [SerializeString](#) (string key, string value, params SerializationOption[] options)
Saves the given string value.
- static void [Serialize< T >](#) (string key, T value, params SerializationOption[] options)
Saves the given value.
- static void [EndSerializeGroup](#) ()
Closes a group started with BeginSerializeGroup.
- static void [BeginDeserializeGroup](#) (string key)
Begins the deserialize group.
- static int [DeserializeInt32](#) (string key, int defaultValue=default(int))
Returns the integer value associated with the given key.
- static float [DeserializeSingle](#) (string key, float defaultValue=default(float))
Returns the float value associated with the given key.
- static string [DeserializeString](#) (string key, string defaultValue=default(string))
Returns the string value associated with the given key.
- static T [Deserialize< T >](#) (string key, T defaultValue=default(T))
Returns the value associated with the given key.
- static void [EndDeserializeGroup](#) ()
Closes a group started with BeginDeserializeGroup.
- static byte [] [SerializeToByteArray< T >](#) (T value, params SerializationOption[] options)
Returns the serialized data of the given value.
- static byte [] [SerializeToByteArray< T >](#) (string key, T value, params SerializationOption[] options)
Returns the serialized data of the given value.

- static T [DeserializeFromByteArray< T >](#) (byte[] data)
Returns the value associated with the serialized data.
- static T [DeserializeFromByteArray< T >](#) (string name, byte[] data)
Returns the value associated with the serialized data.
- static void [BeginSerializeToByteArrayGroup](#) (params SerializationOption[] options)
Begins the serialize group.
- static void [BeginSerializeToByteArrayGroup](#) (string key, params SerializationOption[] options)
Begins the serialize group.
- static byte [] [EndSerializeToByteArrayGroup](#) ()
Closes a group started with [BeginSerializeToByteArrayGroup](#).
- static void [BeginDeserializeToByteArrayGroup](#) (byte[] data)
Begins the deserialize group.
- static void [BeginDeserializeToByteArrayGroup](#) (string key, byte[] data)
Begins the deserialize group.
- static void [EndDeserializeToByteArrayGroup](#) ()
Closes a group started with [BeginDeserializeToByteArrayGroup](#).
- static bool [HasKey](#) (string key)
Determines whether storage contains value associated with given key.
- static void [DeleteKey](#) (string key)
Removes the value identified by the key.
- static void [DeleteAll](#) ()
Removes all the data stored in the system. Use it with caution.
- static void [ClearCache](#) ()
Removes all the data that have been cached by the serialization system.
- static SerializationOption [BufferSize](#) (int value)
Custom serializaion option passed to specify the stream buffer size.
- static SerializationOption [SerializationMethod](#) (SerializationMethodOptions value)
Custom serializaion option passed to specify the serialization (save) method used while serializing object.
- static SerializationOption [StorageTarget](#) (eStorageTarget value)
Custom serializaion option passed to specify the storage location where data will be saved.

4.1.1 Detailed Description

The [SerializationManager](#) class is the interface for saving and restoring object values between game sessions.

4.1.2 Member Function Documentation

4.1.2.1 BeginDeserializeGroup()

```
static void VoxelBusters.Serialization.SerializationManager.BeginDeserializeGroup (
    string key ) [static]
```

Begins the deserialize group.

Parameters

<i>key</i>	Name of the key associated with saved group.
------------	--

```
// use this script to save and read multiple attributes of player from a single document
using UnityEngine;
using System.Collections;
public class ExampleClass : MonoBehaviour
{
    void SaveProfile()
    {
        // shows how to batch multiple data components and save it in a single file
        SerializationManager.BeginSerializeGroup("profile");           // creates a new document
        called profile
        SerializationManager.SerializeString("name", "player1");       // adds name info
        SerializationManager.SerializeInt32("level", 1);              // adds level info
        SerializationManager.SerializeSingle("progress", 0.9f);       // adds progress info
        SerializationManager.EndSerializeGroup();                      // marks end of the document
        and saves the data
    }
    void ReadProfile()
    {
        // shows how to read data fields from batched document
        SerializationManager.BeginDeserializeGroup("profile");         // open saved document
        called profile
        string name = SerializationManager.DeserializeString("name");  // adds name info
        int level = SerializationManager.DeserializeInt32("level");    // adds level info
        float progress= SerializationManager.DeserializeSingle("progress"); // adds progress info
        SerializationManager.EndDeserializeGroup();                    // marks end of the document
        and saves the data
    }
}
```

4.1.2.2 BeginDeserializeToByteArrayGroup() [1/2]

```
static void VoxelBusters.Serialization.SerializationManager.BeginDeserializeToByteArrayGroup (
    byte [] data ) [static]
```

Begins the deserialize group.

Parameters

<i>data</i>	Serialized data.
-------------	------------------

4.1.2.3 BeginDeserializeToByteArrayGroup() [2/2]

```
static void VoxelBusters.Serialization.SerializationManager.BeginDeserializeToByteArrayGroup (
    string key,
    byte [] data ) [static]
```

Begins the deserialize group.

Parameters

<i>key</i>	Name of the key associated with value.
<i>data</i>	Serialized data.

4.1.2.4 BeginSerializeGroup()

```
static void VoxelBusters.Serialization.SerializationManager.BeginSerializeGroup (
    string key,
    params SerializationOption [] options ) [static]
```

Begins the serialize group.

Begins the serialize group.

All serialize calls enclosed inside this element will be saved in a single document. The group must be closed with a call to EndSerializeGroup.

Parameters

<i>key</i>	A string value associated with the value. If specified key already exists, value replaces the existing value. If key is not found, new copy of value will be created in the specified storage.
<i>options</i>	An optional array of serialization option specifies custom settings used for this specific operation. These options overrides the SerializationSettings values.

```
// use this script to save and read multiple attributes of player from a single document
using UnityEngine;
using System.Collections;
public class ExampleClass : MonoBehaviour
{
    void SaveProfile()
    {
        // shows how to batch multiple data components and save it in a single file
        SerializationManager.BeginSerializeGroup("profile"); // creates a new document
        // called profile
        SerializationManager.SerializeString("name", "player1"); // adds name info
        SerializationManager.SerializeInt32("level", 1); // adds level info
        SerializationManager.SerializeSingle("progress", 0.9f); // adds progress info
        SerializationManager.EndSerializeGroup(); // marks end of the document
        // and saves the data
    }
    void ReadProfile()
    {
        // shows how to read data fields from batched document
        SerializationManager.BeginDeserializeGroup("profile"); // open saved document
        // called profile
        string name = SerializationManager.DeserializeString("name"); // adds name info
        int level = SerializationManager.DeserializeInt32("level"); // adds level info
        float progress = SerializationManager.DeserializeSingle("progress"); // adds progress info
        SerializationManager.EndDeserializeGroup(); // marks end of the document
        // and saves the data
    }
}
```

4.1.2.5 BeginSerializeToByteArrayGroup() [1/2]

```
static void VoxelBusters.Serialization.SerializationManager.BeginSerializeToByteArrayGroup (
    params SerializationOption [] options ) [static]
```

Begins the serialize group.

Parameters

<i>options</i>	An optional array of serialization option specifies custom settings used for this specific operation. These options overrides the SerializationSettings values.
----------------	---

4.1.2.6 BeginSerializeToByteArrayGroup() [2/2]

```
static void VoxelBusters.Serialization.SerializationManager.BeginSerializeToByteArrayGroup (
    string key,
    params SerializationOption [] options ) [static]
```

Begins the serialize group.

Parameters

<i>key</i>	Name of the key associated with value.
<i>options</i>	An optional array of serialization option specifies custom settings used for this specific operation. These options overrides the SerializationSettings values.

4.1.2.7 BufferSize()

```
static SerializationOption VoxelBusters.Serialization.SerializationManager.BufferSize (
    int value ) [static]
```

Custom serializaion option passed to specify the stream buffer size.

This option can be used for serialization mode.

Parameters

<i>value</i>	Value.
--------------	--------

4.1.2.8 ClearCache()

```
static void VoxelBusters.Serialization.SerializationManager.ClearCache ( ) [static]
```

Removes all the data that have been cached by the serialization system.

4.1.2.9 DeleteAll()

```
static void VoxelBusters.Serialization.SerializationManager.DeleteAll ( ) [static]
```

Removes all the data stored in the system. Use it with caution.

Call this function to delete all the saved information. Be careful while using this. You cannot undo this action.

4.1.2.10 DeleteKey()

```
static void VoxelBusters.Serialization.SerializationManager.DeleteKey (
    string key ) [static]
```

Removes the value identified by the key.

Parameters

<i>key</i>	Name of the key associated with value.
------------	--

4.1.2.11 Deserialize< T >()

```
static T VoxelBusters.Serialization.SerializationManager.Deserialize< T > (
    string key,
    T defaultValue = default(T) ) [static]
```

Returns the value associated with the given key.

Returns the value associated with the given key.

If the value doesn't already exist in the storage the function will return defaultValue.

Returns

The value previously stored.

Parameters

<i>key</i>	Name of the key associated with value.
<i>defaultValue</i>	Value to return if the specified key is not found in the storage.

```
// use this script to save and read custom object
using UnityEngine;
using System.Collections;
public class PlayerProfile
{
    // fields
    public string      playerName;
    public int         playerLevel;
    public float       playerProgress;
    // constructors
    public PlayerProfile()
    {
    }
    public PlayerProfile(string name, int level, float progress)
    {
        playerName      = name;
        playerLevel     = level;
        playerProgress  = progress;
    }
}
public class ExampleClass : MonoBehaviour
{
    // fields
    PlayerProfile profile = new PlayerProfile("player1", 1, 0.9f);
    // methods
    void SaveProfile()
    {
        // shows how to save custom object
        SerializationManager.Serialize("profile", profile);
    }
    void ReadProfile()
    {
        // shows how to read custom object
        profile = SerializationManager.Deserialize<PlayerProfile>("profile");
    }
}
```


4.1.2.12 DeserializeFromByteArray< T >() [1/2]

```
static T VoxelBusters.Serialization.SerializationManager.DeserializeFromByteArray< T > (
    byte [] data ) [static]
```

Returns the value associated with the serialized data.

Returns

The value previously serialized.

Parameters

<i>data</i>	Serialized data.
-------------	------------------

```
// use this script to save and read custom object
using UnityEngine;
using System.Collections;
public class PlayerProfile
{
    // fields
    public string      playerName;
    public int         playerLevel;
    public float       playerProgress;
    // constructors
    public PlayerProfile()
    {}
    public PlayerProfile(string name, int level, float progress)
    {
        playerName      = name;
        playerLevel     = level;
        playerProgress  = progress;
    }
}
public class ExampleClass : MonoBehaviour
{
    // fields
    PlayerProfile profile = new PlayerProfile("player1", 1, 0.9f);
    byte[] rawData = null;
    // methods
    void SaveProfile()
    {
        // shows how to save custom object
        rawData = SerializationManager.SerializeToByteArray(profile);
    }
    void ReadProfile()
    {
        // shows how to read custom object
        profile = SerializationManager.DeserializeFromByteArray<PlayerProfile>(rawData);
    }
}
```

4.1.2.13 DeserializeFromByteArray< T >() [2/2]

```
static T VoxelBusters.Serialization.SerializationManager.DeserializeFromByteArray< T > (
    string name,
    byte [] data ) [static]
```

Returns the value associated with the serialized data.

Returns

The value previously serialized.

Parameters

key	Name of the key associated with value.
data	Serialized data.

```
// use this script to save and read custom object
using UnityEngine;
using System.Collections;
public class PlayerProfile
{
    // fields
    public string      playerName;
    public int         playerLevel;
    public float       playerProgress;
    // constructors
    public PlayerProfile()
    {}
    public PlayerProfile(string name, int level, float progress)
    {
        playerName      = name;
        playerLevel     = level;
        playerProgress  = progress;
    }
}
public class ExampleClass : MonoBehaviour
{
    // fields
    PlayerProfile profile = new PlayerProfile("player1", 1, 0.9f);
    byte[] rawData = null;
    // methods
    void SaveProfile()
    {
        // shows how to save custom object
        rawData = SerializationManager.SerializeToArray(profile);
    }
    void ReadProfile()
    {
        // shows how to read custom object
        profile = SerializationManager.DeserializeFromByteArray<PlayerProfile>(rawData);
    }
}
```

4.1.2.14 DeserializeInt32()

```
static int VoxelBusters.Serialization.SerializationManager.DeserializeInt32 (
    string key,
    int defaultValue = default(int) ) [static]
```

Returns the integer value associated with the given key.

Returns the integer value associated with the given key.

If the value doesn't already exist in the storage the function will return defaultValue.

Returns

The value previously stored.

Parameters

key	Name of the key associated with integer value.
defaultValue	Integer value to return if the specified key is not found in the storage.

```
// use this script to save and read player level info (int value)
using UnityEngine;
using System.Collections;
```

```

public class ExampleClass : MonoBehaviour
{
    void OnReachedCheckpoint9()
    {
        // saves new level info
        int newLevel = 10;
        SerializationManager.SerializeInt32("level", newLevel);
    }
    int GetPlayerLevel()
    {
        // read saved value
        return SerializationManager.DeserializeInt32("level");
    }
}

```

4.1.2.15 DeserializeSingle()

```

static float VoxelBusters.Serialization.SerializationManager.DeserializeSingle (
    string key,
    float defaultValue = default(float) ) [static]

```

Returns the float value associated with the given key.

Returns the float value associated with the given key.

If the value doesn't already exist in the storage the function will return defaultValue.

Returns

The value previously stored.

Parameters

<i>key</i>	Name of the key associated with float value.
<i>defaultValue</i>	Float value to return if the specified key is not found in the storage.

```

// use this script to save and read player progress (float value)
using UnityEngine;
using System.Collections;
public class ExampleClass : MonoBehaviour
{
    void OnPlayerProgressChanged(float newValue)
    {
        // saves progress info
        SerializationManager.SerializeSingle("progress", newValue);
    }
    float GetPlayerProgress()
    {
        // read saved value
        return SerializationManager.DeserializeSingle("progress");
    }
}

```

4.1.2.16 DeserializeString()

```

static string VoxelBusters.Serialization.SerializationManager.DeserializeString (
    string key,
    string defaultValue = default(string) ) [static]

```

Returns the string value associated with the given key.

Returns the string value associated with the given key.

If the value doesn't already exist in the storage the function will return defaultValue.

Returns

The value previously stored.

Parameters

<i>key</i>	Name of the key associated with string value.
<i>defaultValue</i>	String value to return if the specified key is not found in the storage.

```
// use this script to save and read player name (string value)
using UnityEngine;
using System.Collections;
public class ExampleClass : MonoBehaviour
{
    void OnPlayerNameChanged(string newName)
    {
        // saves name info
        SerializationManager.SerializeString("name", newName);
    }
    string GetPlayerName()
    {
        // read saved value
        return SerializationManager.DeserializeString("name");
    }
}
```

4.1.2.17 EndDeserializeGroup()

```
static void VoxelBusters.Serialization.SerializationManager.EndDeserializeGroup ( ) [static]
```

Closes a group started with BeginDeserializeGroup.

4.1.2.18 EndDeserializeToByteArrayGroup()

```
static void VoxelBusters.Serialization.SerializationManager.EndDeserializeToByteArrayGroup ( )
[static]
```

Closes a group started with [BeginDeserializeToByteArrayGroup](#).

4.1.2.19 EndSerializeGroup()

```
static void VoxelBusters.Serialization.SerializationManager.EndSerializeGroup ( ) [static]
```

Closes a group started with BeginSerializeGroup.

4.1.2.20 EndSerializeToByteArrayGroup()

```
static byte [] VoxelBusters.Serialization.SerializationManager.EndSerializeToByteArrayGroup (
) [static]
```

Closes a group started with [BeginSerializeToByteArrayGroup](#).

4.1.2.21 HasKey()

```
static bool VoxelBusters.Serialization.SerializationManager.HasKey (
    string key ) [static]
```

Determines whether storage contains value associated with given key.

Returns

true if storage has the specified key; otherwise, false.

Parameters

<i>key</i>	A string value used to uniquely identify the stored value.
------------	--

4.1.2.22 SerializationMethod()

```
static SerializationOption VoxelBusters.Serialization.SerializationManager.SerializationMethod
(
    SerializationMethodOptions value ) [static]
```

Custom serializaion option passed to specify the serialization (save) method used while serializing object.

This option can be used for serialization mode.

Parameters

<i>value</i>	Value.
--------------	--------

4.1.2.23 Serialize< T >()

```
static void VoxelBusters.Serialization.SerializationManager.Serialize< T > (
    string key,
    T value,
    params SerializationOption [] options ) [static]
```

Saves the given value.

Parameters

<i>key</i>	A string value associated with the value. If specified key already exists, value replaces the existing value. If key is not found, new copy of value will be created in the specified storage.
<i>value</i>	The value to be saved.
<i>options</i>	An optional array of serialization option specifies custom settings used for this specific operation. These options overrides the SerializationSettings values.

```
// use this script to save and read custom object (here, PlayerProfile)
using UnityEngine;
using System.Collections;
public class PlayerProfile
{
    // fields
    public string      playerName;
    public int         playerLevel;
    public float       playerProgress;
    // constructors
    public PlayerProfile()
    {}
    public PlayerProfile(string name, int level, float progress)
    {
        playerName      = name;
        playerLevel     = level;
        playerProgress  = progress;
    }
}
public class ExampleClass : MonoBehaviour
{
    // fields
    PlayerProfile profile = new PlayerProfile("player1", 1, 0.9f);
    // methods
    void SaveProfile()
    {
        // shows how to save custom object
        SerializationManager.Serialize("profile", profile);
    }
    void ReadProfile()
    {
        // shows how to read custom object
        profile = SerializationManager.Deserialize<PlayerProfile>("profile");
    }
}
```

4.1.2.24 SerializeInt32()

```
static void VoxelBusters.Serialization.SerializationManager.SerializeInt32 (
    string key,
    int value,
    params SerializationOption [] options ) [static]
```

Saves the given integer value.

Parameters

<i>key</i>	A string value associated with the value. If specified key already exists, value replaces the existing value. If key is not found, new copy of value will be created in the specified storage.
<i>value</i>	The value to be saved.
<i>options</i>	An optional array of serialization option specifies custom settings used for this specific operation. These options overrides the SerializationSettings values.

```
// use this script to save and read player level info (int value)
using UnityEngine;
using System.Collections;
public class ExampleClass : MonoBehaviour
{
    void OnReachedCheckpoint9()
    {
```

```

        // saves new level info
        int newLevel = 10;
        SerializationManager.SerializeInt32("level", newLevel);
    }
    int GetPlayerLevel()
    {
        // read saved value
        return SerializationManager.DeserializeInt32("level");
    }
}

```

4.1.2.25 SerializeSingle()

```

static void VoxelBusters.Serialization.SerializationManager.SerializeSingle (
    string key,
    float value,
    params SerializationOption [] options ) [static]

```

Saves the given float value.

Parameters

<i>key</i>	A string value associated with the value. If specified key already exists, value replaces the existing value. If key is not found, new copy of value will be created in the specified storage.
<i>value</i>	The value to be saved.
<i>options</i>	An optional array of serialization option specifies custom settings used for this specific operation. These options overrides the <code>SerializationSettings</code> values.

```

// use this script to save and read player progress (float value)
using UnityEngine;
using System.Collections;
public class ExampleClass : MonoBehaviour
{
    void OnPlayerProgressChanged(float newValue)
    {
        // saves progress info
        SerializationManager.SerializeSingle("progress", newValue);
    }
    float GetPlayerProgress()
    {
        // read saved value
        return SerializationManager.DeserializeSingle("progress");
    }
}

```

4.1.2.26 SerializeString()

```

static void VoxelBusters.Serialization.SerializationManager.SerializeString (
    string key,
    string value,
    params SerializationOption [] options ) [static]

```

Saves the given string value.

Parameters

<i>key</i>	A string value associated with the value. If specified key already exists, value replaces the existing value. If key is not found, new copy of value will be created in the specified storage.
<i>value</i>	The value to be saved.
<i>options</i>	An optional array of serialization option specifies custom settings used for this specific operation. These options overrides the <code>SerializationSettings</code> values.

```
// use this script to save and read player name (string value)
using UnityEngine;
using System.Collections;
public class ExampleClass : MonoBehaviour
{
    void OnPlayerNameChanged(string newName)
    {
        // saves name info
        SerializationManager.SerializeString("name", newName);
    }
    string GetPlayerName()
    {
        // read saved value
        return SerializationManager.DeserializeString("name");
    }
}
```

4.1.2.27 SerializeToByteArray< T >() [1/2]

```
static byte [] VoxelBusters.Serialization.SerializationManager.SerializeToByteArray< T > (
    T value,
    params SerializationOption [] options ) [static]
```

Returns the serialized data of the given value.

This method doesn't save the data in any form (as file or PlayerPrefs). Instead user is responsible to manage this data. Provide this byte array as input to DeserializeFromByteArray method inorder to retrieve back original value.

Parameters

<i>value</i>	The value to be saved.
<i>options</i>	An optional array of serialization option specifies custom settings used for this specific operation. These options overrides the SerializationSettings values.

```
// use this script to save and read custom object (here, PlayerProfile)
using UnityEngine;
using System.Collections;
public class PlayerProfile
{
    // fields
    public string      playerName;
    public int         playerLevel;
    public float       playerProgress;
    // constructors
    public PlayerProfile()
    {
    }
    public PlayerProfile(string name, int level, float progress)
    {
        playerName      = name;
        playerLevel     = level;
        playerProgress  = progress;
    }
}
public class ExampleClass : MonoBehaviour
{
    // fields
    PlayerProfile profile = new PlayerProfile("player1", 1, 0.9f);
    byte[] rawData = null;
    // methods
    void SaveProfile()
    {
        // shows how to save custom object
        rawData = SerializationManager.SerializeToByteArray(profile);
    }
    void ReadProfile()
    {
        // shows how to read custom object
        profile = SerializationManager.DeserializeFromByteArray<PlayerProfile>(rawData);
    }
}
```


4.1.2.28 SerializeToByteArray< T >() [2/2]

```
static byte [] VoxelBusters.Serialization.SerializationManager.SerializeToByteArray< T > (
    string key,
    T value,
    params SerializationOption [] options ) [static]
```

Returns the serialized data of the given value.

This method doesn't save the data in any form (as file or PlayerPrefs). Instead user is responsible to manage this data. Provide this byte array as input to DeserializeFromByteArray method inorder to retrieve back original value.

Parameters

<i>key</i>	Name of the key associated with value.
<i>value</i>	The value to be saved.
<i>options</i>	An optional array of serialization option specifies custom settings used for this specific operation. These options overrides the SerializationSettings values.

```
// use this script to save and read custom object (here, PlayerProfile)
using UnityEngine;
using System.Collections;
public class PlayerProfile
{
    // fields
    public string      playerName;
    public int         playerLevel;
    public float       playerProgress;
    // constructors
    public PlayerProfile()
    {
    }
    public PlayerProfile(string name, int level, float progress)
    {
        playerName      = name;
        playerLevel     = level;
        playerProgress  = progress;
    }
}
public class ExampleClass : MonoBehaviour
{
    // fields
    PlayerProfile profile = new PlayerProfile("player1", 1, 0.9f);
    byte[] rawData = null;
    // methods
    void SaveProfile()
    {
        // shows how to save custom object
        rawData = SerializationManager.SerializeToByteArray("key", profile);
    }
    void ReadProfile()
    {
        // shows how to read custom object
        profile = SerializationManager.DeserializeFromByteArray<PlayerProfile>("key", rawData);
    }
}
```

4.1.2.29 StorageTarget()

```
static SerializationOption VoxelBusters.Serialization.SerializationManager.StorageTarget (
    eStorageTarget value ) [static]
```

Custom serializaion option passed to specify the storage location where data will be saved.

Parameters

<i>value</i>	Value.
--------------	--------

The documentation for this class was generated from the following file:

- /Volumes/Work/Projects/Products/AssetStore/Cross Platform Easy Save/Unity Project/Assets/Plugins/↵
VoxelBusters/CrossPlatformEasySave/Scripts/SerializationManager.cs

4.2 VoxelBusters.Serialization.SerializationUtility Class Reference

Utility functions for implementing and extending the serialization support for Unity objects created at runtime.

Static Public Member Functions

- static GameObject [CreateGameObject](#) ()
Creates a new game object.
- static GameObject [CreateGameObject](#) (string name)
Creates a new game object with specified name.
- static GameObject [CreateGameObject](#) (string name, params Type[] components)
Creates the game object with specified name and components.
- static GameObject [Instantiate](#) (GameObject original)
Clones the object original and returns the clone.
- static GameObject [Instantiate](#) (GameObject original, Vector3 position, Quaternion rotation)
Clones the object original and returns the clone.
- static GameObject [Instantiate](#) (GameObject original, Transform parent, bool worldPositionStays)
Clones the object original and returns the clone.
- static GameObject [Instantiate](#) (GameObject original, Vector3 position, Quaternion rotation, Transform parent)
Clones the object original and returns the clone.
- static Component [AddComponent](#) (GameObject gameObject, Type componentType)
Adds a component class of type componentType to the game object.
- static T [AddComponent](#)< T > (GameObject gameObject)
Adds a component class of type T to the game object.
- static void [Destroy](#) (GameObject gameObject)
Removes the specified gameobject.
- static void [Destroy](#) (Component component)
Removes the specified component.

4.2.1 Detailed Description

Utility functions for implementing and extending the serialization support for Unity objects created at runtime.

4.2.2 Member Function Documentation

4.2.2.1 AddComponent()

```
static Component VoxelBusters.Serialization.SerializationUtility.AddComponent (
    GameObject gameObject,
    Type componentType ) [static]
```

Adds a component class of type componentType to the game object.

Returns

The component attached to the game object.

Parameters

<i>gameObject</i>	Game object.
<i>componentType</i>	Component type.

4.2.2.2 AddComponent< T >()

```
static T VoxelBusters.Serialization.SerializationUtility.AddComponent< T > (  
    GameObject gameObject ) [static]
```

Adds a component class of type T to the game object.

Returns

The component attached to the game object.

Parameters

<i>gameObject</i>	Game object.
-------------------	--------------

Type Constraints

T* : *Component

4.2.2.3 CreateGameObject() [1/3]

```
static GameObject VoxelBusters.Serialization.SerializationUtility.CreateGameObject ( ) [static]
```

Creates a new game object.

Returns

The game object.

4.2.2.4 CreateGameObject() [2/3]

```
static GameObject VoxelBusters.Serialization.SerializationUtility.CreateGameObject (  
    string name ) [static]
```

Creates a new game object with specified name.

Returns

The game object.

Parameters

<i>name</i>	The name that the GameObject is created with.
-------------	---

4.2.2.5 CreateGameObject() [3/3]

```
static GameObject VoxelBusters.Serialization.SerializationUtility.CreateGameObject (
    string name,
    params Type [] components ) [static]
```

Creates the game object with specified name and components.

Returns

The game object.

Parameters

<i>name</i>	The name that the GameObject is created with.
<i>components</i>	An array of Components to add to the GameObject on creation.

4.2.2.6 Destroy() [1/2]

```
static void VoxelBusters.Serialization.SerializationUtility.Destroy (
    GameObject gameObject ) [static]
```

Removes the specified gameobject.

Parameters

<i>gameObject</i>	The game object to be destroyed.
-------------------	----------------------------------

4.2.2.7 Destroy() [2/2]

```
static void VoxelBusters.Serialization.SerializationUtility.Destroy (
    Component component ) [static]
```

Removes the specified component.

Parameters

<i>gameObject</i>	The component to be destroyed.
-------------------	--------------------------------

4.2.2.8 Instantiate() [1/4]

```
static GameObject VoxelBusters.Serialization.SerializationUtility.Instantiate (
    GameObject original ) [static]
```

Clones the object original and returns the clone.

Parameters

<i>original</i>	An existing object that you want to make a copy of.
-----------------	---

4.2.2.9 Instantiate() [2/4]

```
static GameObject VoxelBusters.Serialization.SerializationUtility.Instantiate (
    GameObject original,
    Vector3 position,
    Quaternion rotation ) [static]
```

Clones the object original and returns the clone.

Parameters

<i>original</i>	An existing object that you want to make a copy of.
<i>position</i>	Position for the new object.
<i>rotation</i>	Orientation of the new object.

4.2.2.10 Instantiate() [3/4]

```
static GameObject VoxelBusters.Serialization.SerializationUtility.Instantiate (
    GameObject original,
    Transform parent,
    bool worldPositionStays ) [static]
```

Clones the object original and returns the clone.

Parameters

<i>original</i>	An existing object that you want to make a copy of.
<i>parent</i>	Parent that will be assigned to the new object.
<i>worldPositionStays</i>	Pass true when assigning a parent Object to maintain the world position of the Object, instead of setting its position relative to the new parent. Pass false to set the Object's position relative to its new parent.

4.2.2.11 Instantiate() [4/4]

```
static GameObject VoxelBusters.Serialization.SerializationUtility.Instantiate (
    GameObject original,
    Vector3 position,
    Quaternion rotation,
    Transform parent ) [static]
```

Clones the object original and returns the clone.

Parameters

<i>original</i>	An existing object that you want to make a copy of.
<i>position</i>	Position for the new object.
<i>rotation</i>	Orientation of the new object.
<i>parent</i>	Parent that will be assigned to the new object.

The documentation for this class was generated from the following file:

- /Volumes/Work/Projects/Products/AssetStore/Cross Platform Easy Save/Unity Project/Assets/Plugins/↵
VoxelBusters/CrossPlatformEasySave/Scripts/SerializationUtility.cs

Index

- AddComponent
 - VoxelBusters.Serialization.SerializationUtility, [22](#)
- AddComponent< T >
 - VoxelBusters.Serialization.SerializationUtility, [23](#)
- BeginDeserializeGroup
 - VoxelBusters.Serialization.SerializationManager, [8](#)
- BeginDeserializeToByteArrayGroup
 - VoxelBusters.Serialization.SerializationManager, [9](#)
- BeginSerializeGroup
 - VoxelBusters.Serialization.SerializationManager, [9](#)
- BeginSerializeToByteArrayGroup
 - VoxelBusters.Serialization.SerializationManager, [10](#)
- BufferSize
 - VoxelBusters.Serialization.SerializationManager, [11](#)
- ClearCache
 - VoxelBusters.Serialization.SerializationManager, [11](#)
- CreateGameObject
 - VoxelBusters.Serialization.SerializationUtility, [23](#), [24](#)
- DeleteAll
 - VoxelBusters.Serialization.SerializationManager, [11](#)
- DeleteKey
 - VoxelBusters.Serialization.SerializationManager, [11](#)
- Deserialize< T >
 - VoxelBusters.Serialization.SerializationManager, [12](#)
- DeserializeFromByteArray< T >
 - VoxelBusters.Serialization.SerializationManager, [12](#), [13](#)
- DeserializeInt32
 - VoxelBusters.Serialization.SerializationManager, [14](#)
- DeserializeSingle
 - VoxelBusters.Serialization.SerializationManager, [15](#)
- DeserializeString
 - VoxelBusters.Serialization.SerializationManager, [15](#)
- Destroy
 - VoxelBusters.Serialization.SerializationUtility, [24](#)
- EndDeserializeGroup
 - VoxelBusters.Serialization.SerializationManager, [16](#)
- EndDeserializeToByteArrayGroup
 - VoxelBusters.Serialization.SerializationManager, [16](#)
- EndSerializeGroup
 - VoxelBusters.Serialization.SerializationManager, [16](#)
- EndSerializeToByteArrayGroup
 - VoxelBusters.Serialization.SerializationManager, [16](#)
- HasKey
 - VoxelBusters.Serialization.SerializationManager, [17](#)
- Instantiate
 - VoxelBusters.Serialization.SerializationUtility, [25](#), [26](#)
- SerializationMethod
 - VoxelBusters.Serialization.SerializationManager, [17](#)
- Serialize< T >
 - VoxelBusters.Serialization.SerializationManager, [17](#)
- SerializeInt32
 - VoxelBusters.Serialization.SerializationManager, [18](#)
- SerializeSingle
 - VoxelBusters.Serialization.SerializationManager, [19](#)
- SerializeString
 - VoxelBusters.Serialization.SerializationManager, [19](#)
- SerializeToByteArray< T >
 - VoxelBusters.Serialization.SerializationManager, [20](#)
- StorageTarget
 - VoxelBusters.Serialization.SerializationManager, [21](#)
- VoxelBusters, [5](#)
- VoxelBusters.Serialization, [5](#)
- VoxelBusters.Serialization.SerializationManager, [7](#)
 - BeginDeserializeGroup, [8](#)
 - BeginDeserializeToByteArrayGroup, [9](#)
 - BeginSerializeGroup, [9](#)
 - BeginSerializeToByteArrayGroup, [10](#)
 - BufferSize, [11](#)

- ClearCache, [11](#)
- DeleteAll, [11](#)
- DeleteKey, [11](#)
- Deserialize< T >, [12](#)
- DeserializeFromByteArray< T >, [12](#), [13](#)
- DeserializeInt32, [14](#)
- DeserializeSingle, [15](#)
- DeserializeString, [15](#)
- EndDeserializeGroup, [16](#)
- EndDeserializeToByteArrayGroup, [16](#)
- EndSerializeGroup, [16](#)
- EndSerializeToByteArrayGroup, [16](#)
- HasKey, [17](#)
- SerializationMethod, [17](#)
- Serialize< T >, [17](#)
- SerializeInt32, [18](#)
- SerializeSingle, [19](#)
- SerializeString, [19](#)
- SerializeToByteArray< T >, [20](#)
- StorageTarget, [21](#)
- VoxelBusters.Serialization.SerializationUtility, [22](#)
 - AddComponent, [22](#)
 - AddComponent< T >, [23](#)
 - CreateGameObject, [23](#), [24](#)
 - Destroy, [24](#)
 - Instantiate, [25](#), [26](#)