HOT TOTEM

EpicPrefs

Contents

Introduction

Welcome and thank you for buying EpicPrefs.

We hope you like our product and to make its usage as easy as possible we put up this extensive documentation. If you still have any questions or suggestions, don't hesitate to contact us at support@hot-totem.com or on Twitter @hottotem and we will get back to you as fast as possible.

If you would like any additional supported types for the EpicPrefs, please contact us too and we will be happy to add them.

This documentation is separated into four parts:

1. Explanation of what EpicPrefs can do
2. Overview of the graphical interface (further referenced to as EpicPrefsEditor)
3. Overview and explanation on how to use the asset by code (C#)
4. Complete method documentation with all savable types

So let's get started !

EpicPrefs Overview

EpicPrefs is a replacement for Unity's PlayerPrefs. Everything you can do with PlayerPrefs, you can do with EpicPrefs too. Plus, much much more.

EpicPrefs has a whole bunch of different types that are supported, from basic ones like string, bools, float over to complex ones like Dictionaries or Lists, but also Unity specifique types like Colors, Quaternions or Vectors. Only to name a few.

In addition to all these types, EpicPrefs offers you salted AES Encryption. This will allow you to secure your Prefs from hackers wanting to modify them, unlocking in app purchases etc.

Disclaimer: No on-device de- and encryption is 100% secure. A dedicated hacker will be able to encrypt, modify or steal your data. What we do is at least prevent average users from doing so, as well as automated tools. Plus, we do our best to make the hackers life as hard as possible, so at least they got to invest a good amount of time.

To round it all up, EpicPrefs also comes with a graphical interface for in-Editor editing, adding and removing EpicPrefs. This editor allows you to set new prefs in your development stage, or test something when the game is running in editor mode. You can also see the EpicPrefs update as you change them by code, so you see what is actually happening and what's not.

In addition to this all, we have built in a feature to export all your prefs that you set at editor time to your build! Normally, if you use a Graphical Editor for PlayerPrefs, your prefs will be lost once you build your game and run it on a device. With our tool, you can simply hit a button and all your prefs will be right in your game!

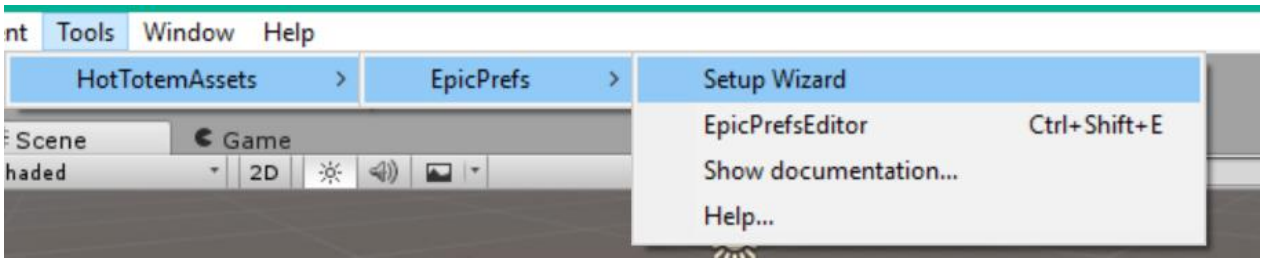
To find out how to use it, read on!

Setup

The very first thing you need to do after importing EpicPrefs is setting up the encryption keys.

If you do not plan on using encryption, you can skip this step, however it is recommended to set them anyway, you might want to start using the encryption later on.

To do so, navigate **to Tools → HotTotemAssets → EpicPrefs → Setup Wizard**



This will bring up the Setup Window:

To setup the keys, simply click on Edit Keys and follow the instructions, save them and you are done!

Be aware that the initialization vector needs to be exactly 16 bytes long!

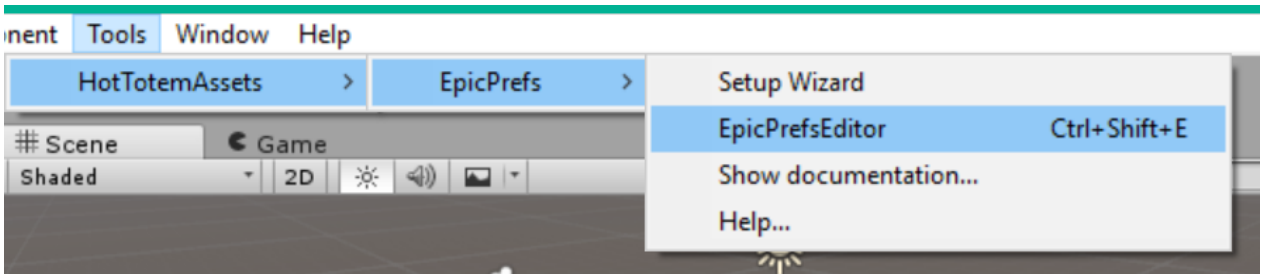
Export

Values set in the editor are by default not exported to your build. This means, if you are setting prefs by code they will run as usually in your game, if you add some Prefs inside the editor though, they will not live in your game. If you want them to be part of the final build, you need to hit the Export EpicPrefs to build button in the setup wizard. It is recommended to do this prior to building your game as it can take some time to complete and only needs to be done before building. This will bring up a dialog where you can select individual EpicPrefs. Every Pref will be added to your game, but only the selected ones will overwrite existing ones. So if you do not select anything, EpicPrefs will make sure that every pref exists in your game. If the game itself did set one pref already though, it will not be overwritten (unless selected in the dialog).

EpicPrefsEditor

1. Open up the EpicPrefsEditor

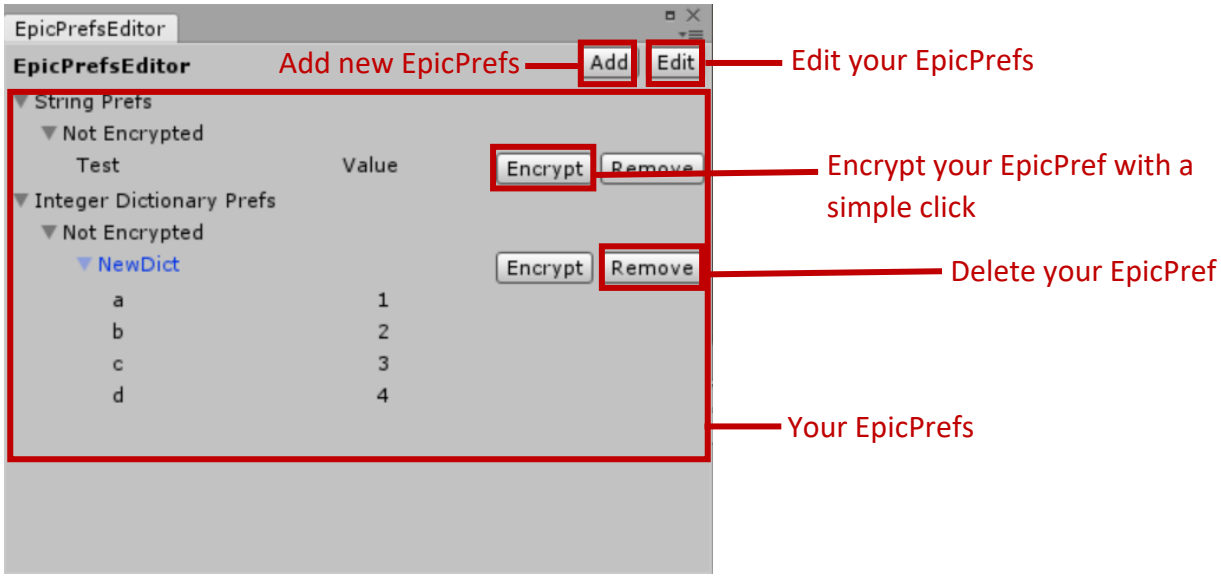
Simply navigate to **Tools → HotTotemAssets → EpicPrefs → EpicPrefsEditor** like shown in the image below, or hit the predefined shortcut **Ctrl(Cmd) + Shift + E**.



This will bring up the new EpicPrefsEditor, where you can then add, view, modify, delete or de-encrypt your prefs!

2. The main window

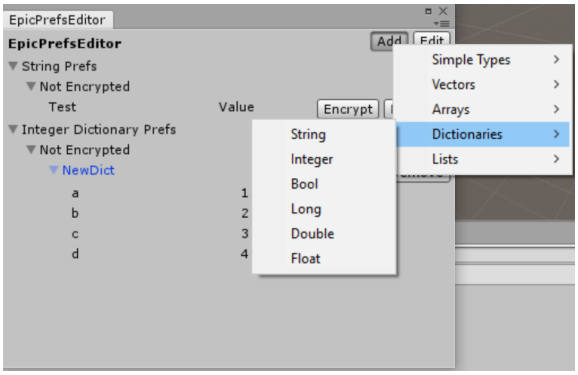
The main window should now appear. You can dock it anywhere like you can do with any other window in Unity. Here is a little overview of it:



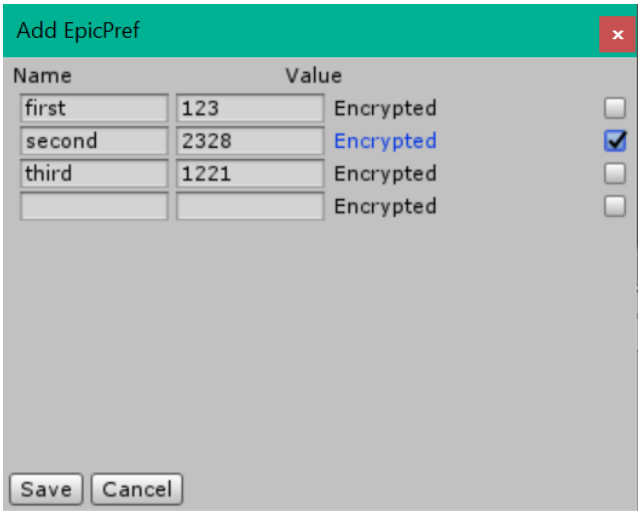
EpicPrefs are grouped under their types. You can unfold them and see them when you want, and hide them when you don't.

1. Add new EpicPrefs

Click the add button to add a new EpicPref. The menu will let you chose from all the various supported types, and will then popup a creation wizard that lets you easily create new Prefs.



As you fill out the fields, new ones will appear and let you add multiple at once. You can also encrypt the values directly.



2. Edit

You can also easily modify existing EpicPrefs by hitting the edit button. Make your changes and then you can hit the save button. This will also let you add new values to your dictionaries, or remove some of them.

EpicPrefs by Code

The usage of EpicPrefs is actually pretty straightforward. You can use it in exactly the same way as you do with Unity's PlayerPrefs.

It's easier to explain it with an example :

```
EpicPrefs.SetString("Key", "Value", true);
```

So everytime you want to access a pref or set one, you call `EpicPrefs.` Followed by whatever you want to do. The syntax follows a pretty easy convention:

To set values call SetType(), where you replace type by the type you want to set.

To get values call GetType().

As you can see in the code, it takes three parameters. The first two are equal to PlayerPrefs, the third is the encryption. If omitted, it defaults to false.

One more thing: When you call any function on the EpicPrefs the first time on your device, the initialization is being made. If you have a large number of exported values, this can cause a little hiccup in your game. To prevent this, you can call the

```
EpicPrefs.Initialize()
```

method anywhere in any Awake() or Start() function. This then executes the initialization and delivers you a perfectly smooth experience at runtime.

That's already everything!

You can see the following complete documentation for further information.

Complete Documentation

Contents

Chapter 1

Namespace Index

1.1 Packages

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Namespace Documentation

4.1 dotBunny Namespace Reference

Namespaces

- namespace Unity

4.2 dotBunny.Unity Namespace Reference

Classes

- class VSCode
- class VSCodeAssetPostprocessor

  VSCode Asset AssetPostprocessor

4.3 HotTotem Namespace Reference

Classes

- class Key

4.4 UnityStandardAssets Namespace Reference

Namespaces

- namespace CrossPlatformInput

4.5 UnityStandardAssets.CrossPlatformInput Namespace Reference

Namespaces

- namespace Inspector

4.6 UnityStandardAssets.CrossPlatformInput.Inspector Namespace Reference

Classes

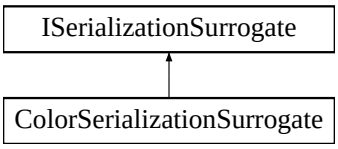
- class CrossPlatformInitialize

Chapter 5

Class Documentation

5.1 ColorSerializationSurrogate Class Reference

Inheritance diagram for ColorSerializationSurrogate:



Public Member Functions

- void **GetObjectData** (System.Object obj, SerializationInfo info, StreamingContext context)
- System.Object **SetObjectData** (System.Object obj, SerializationInfo info, StreamingContext context, I↩ SurrogateSelector selector)

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Plugins/Hot↩ TotemAssets/EpicPrefs/Code/Handlers/CustomSurrogates/ColorSerializationSurrogate.cs

5.2 UnityStandardAssets.CrossPlatformInput.Inspector.CrossPlatformInitialize Class Reference

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Editor/Cross↩ PlatformInput/CrossPlatformInputInitialize.cs

5.3 Cryptor Class Reference

Static Public Member Functions

- static string **Encrypt** (string value)
- static string **Encrypt** (float value)

- static string **Encrypt** (int value)
- static string **Encrypt** (bool value)
- static string **Encrypt** (long value)
- static string **Encrypt** (double value)
- static object **Decrypt** (string value, Serializer.SerializationTypes type)

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Plugins/Hot↩ TotemAssets/EpicPrefs/Code/Handlers/Cryptor.cs

5.4 EpicPrefs Class Reference

EpicPrefs is a replacement for Unity's PlayerPrefs. Everything you can do with PlayerPrefs, you can do with Epic↩ Prefs too. Plus, much much more.

Static Public Member Functions

- static void Initialize ()

  Initialize and setup any imported EpicPrefs.
- static string getPassPhrase ()

  Retrieve the current passPhrase.
- static void setPassPhrase (string value)

  Set the new passPhrase and remove the previously encrypted values if the passPhrase has changed.
- static string getInitVector ()

  Retrieve the current initVector.
- static void setInitVector (string value)

  Set the new initVector and remove the previously encrypted values if the initVector has changed.
- static bool SetString (string name, string value, bool encrypted=false)

  Call this function to save a string to the EpicPrefs. See the parameters for more information on what to pass.
- static string GetString (string name, bool encrypted=false)

  Call this function to retrieve a string from the EpicPrefs. If the Pref does not exist an empty string is returned.
- static string GetString (string name, string defaultValue, bool encrypted=false)

  Call this function to retrieve a string from the EpicPrefs. If the Pref does not exist a default string is returned.
- static void DeleteString (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static bool SetInt (string name, int value, bool encrypted=false)

  Call this function to save an integer to the EpicPrefs. See the parameters for more information on what to pass.
- static int GetInt (string name, bool encrypted=false)

  Call this function to retrieve an integer from the EpicPrefs. If the Pref does not exist -1 is returned.
- static int GetInt (string name, int defaultValue, bool encrypted=false)

  Call this function to retrieve an integer from the EpicPrefs. If the Pref does not exist a default integer is returned.
- static void DeleteInt (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static bool SetBool (string name, bool value, bool encrypted=false)

  Call this function to save a bool to the EpicPrefs. See the parameters for more information on what to pass.
- static bool GetBool (string name, bool encrypted=false)

  Call this function to retrieve a boolean from the EpicPrefs. If the Pref does not exist false is returned.
- static bool GetBool (string name, bool defaultValue, bool encrypted)

  Call this function to retrieve a boolean from the EpicPrefs. If the Pref does not exist a default bool is returned.

- static void DeleteBool (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static bool SetFloat (string name, float value, bool encrypted=false)

  Call this function to save a float to the EpicPrefs. See the parameters for more information on what to pass.
- static float GetFloat (string name, bool encrypted=false)

  Call this function to retrieve a float from the EpicPrefs. If the Pref does not exist -1 is returned.
- static float GetFloat (string name, float defaultValue, bool encrypted=false)

  Call this function to retrieve a float from the EpicPrefs. If the Pref does not exist a default float is returned.
- static void DeleteFloat (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static bool SetLong (string name, long value, bool encrypted=false)

  Call this function to save a long to the EpicPrefs. See the parameters for more information on what to pass.
- static long GetLong (string name, bool encrypted=false)

  Call this function to retrieve a long from the EpicPrefs. If the Pref does not exist -1 is returned.
- static long GetLong (string name, long defaultValue, bool encrypted=false)

  Call this function to retrieve a long from the EpicPrefs. If the Pref does not exist a default long is returned.
- static void DeleteLong (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static bool SetDouble (string name, double value, bool encrypted=false)

  Call this function to save a double to the EpicPrefs. See the parameters for more information on what to pass.
- static double GetDouble (string name, bool encrypted=false)

  Call this function to retrieve a double from the EpicPrefs. If the Pref does not exist -1 is returned.
- static double GetDouble (string name, double defaultValue, bool encrypted=false)

  Call this function to retrieve a double from the EpicPrefs. If the Pref does not exist a default double is returned.
- static void DeleteDouble (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static bool SetDict (string name, Dictionary< string, string > value, bool encrypted=false)

  Call this function to save a Dictionary with its Keys being a string and Values being a string to the EpicPrefs. See the parameters for more information on what to pass.
- static bool SetDict (string name, Dictionary< string, int > value, bool encrypted=false)

  Call this function to save a Dictionary with its Keys being a string and Values being an int to the EpicPrefs. See the parameters for more information on what to pass.
- static bool SetDict (string name, Dictionary< string, float > value, bool encrypted=false)

  Call this function to save a Dictionary with its Keys being a string and Values being a float to the EpicPrefs. See the parameters for more information on what to pass.
- static bool SetDict (string name, Dictionary< string, long > value, bool encrypted=false)

  Call this function to save a Dictionary with its Keys being a string and Values being a long to the EpicPrefs. See the parameters for more information on what to pass.
- static bool SetDict (string name, Dictionary< string, double > value, bool encrypted=false)

  Call this function to save a Dictionary with its Keys being a string and Values being a double to the EpicPrefs. See the parameters for more information on what to pass.
- static bool SetDict (string name, Dictionary< string, bool > value, bool encrypted=false)

  Call this function to save a Dictionary with its Keys being a string and Values being a bool to the EpicPrefs. See the parameters for more information on what to pass.
- static Dictionary< string, string > GetDictStringString (string name, bool encrypted=false)

  Call this function to retrieve a Dictionary with its Keys being strings and its Values being strings from the EpicPrefs. If the Pref does not exist null is returned.
- static Dictionary< string, string > GetDictStringString (string name, Dictionary< string, string > defaultValue, bool encrypted=false)

  Call this function to retrieve a Dictionary with its Keys being strings and its Values being strings from the EpicPrefs. If the Pref does not exist the default value is returned.
- static Dictionary< string, int > GetDictStringInt (string name, bool encrypted=false)

Call this function to retrieve a Dictionary with its Keys being strings and its Values being integers from the EpicPrefs. If the Pref does not exist null is returned.

- static Dictionary< string, int > GetDictStringInt (string name, Dictionary< string, int > defaultValue, bool encrypted=false)

  Call this function to retrieve a Dictionary with its Keys being strings and its Values being strings from the EpicPrefs. If the Pref does not exist the default value is returned.
- static Dictionary< string, float > GetDictStringFloat (string name, bool encrypted=false)

  Call this function to retrieve a Dictionary with its Keys being strings and its Values being floats from the EpicPrefs. If the Pref does not exist null is returned.
- static Dictionary< string, float > GetDictStringFloat (string name, Dictionary< string, float > defaultValue, bool encrypted=false)

  Call this function to retrieve a Dictionary with its Keys being strings and its Values being floats from the EpicPrefs. If the Pref does not exist the default value is returned.
- static Dictionary< string, double > GetDictStringDouble (string name, bool encrypted=false)

  Call this function to retrieve a Dictionary with its Keys being strings and its Values being doubles from the EpicPrefs. If the Pref does not exist null is returned.
- static Dictionary< string, double > GetDictStringDouble (string name, Dictionary< string, double > default↩ Value, bool encrypted=false)

  Call this function to retrieve a Dictionary with its Keys being strings and its Values being doubles from the EpicPrefs. If the Pref does not exist the default value is returned.
- static Dictionary< string, bool > GetDictStringBool (string name, bool encrypted=false)

  Call this function to retrieve a Dictionary with its Keys being strings and its Values being booleans from the EpicPrefs. If the Pref does not exist null is returned.
- static Dictionary< string, bool > GetDictStringBool (string name, Dictionary< string, bool > defaultValue, bool encrypted=false)

  Call this function to retrieve a Dictionary with its Keys being strings and its Values being booleans from the EpicPrefs. If the Pref does not exist the default value is returned.
- static Dictionary< string, long > GetDictStringLong (string name, bool encrypted=false)

  Call this function to retrieve a Dictionary with its Keys being strings and its Values being longs from the EpicPrefs. If the Pref does not exist null is returned.
- static Dictionary< string, long > GetDictStringLong (string name, Dictionary< string, long > defaultValue, bool encrypted=false)

  Call this function to retrieve a Dictionary with its Keys being strings and its Values being longs from the EpicPrefs. If the Pref does not exist the default value is returned.
- static void DeleteDict (string name, bool encrypted)

  Deletes a Dictionary permanently.
- static bool SetColor (string name, Color value, bool encrypted=false)

  Call this function to save a Color to the EpicPrefs. See the parameters for more information on what to pass.
- static Color GetColor (string name, bool encrypted=false)

  Call this function to retrieve a color from the EpicPrefs. If the Pref does not exist white is returned.
- static Color GetColor (string name, Color defaultValue, bool encrypted=false)

  Call this function to retrieve a color from the EpicPrefs. If the Pref does not exist a default color is returned.
- static void DeleteColor (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static bool SetQuaternion (string name, Quaternion value, bool encrypted=false)

  Call this function to save a Quaternion to the EpicPrefs. See the parameters for more information on what to pass.
- static Quaternion GetQuaternion (string name, bool encrypted=false)

  Call this function to retrieve a Quaternion from the EpicPrefs. If the Pref does not exist a zero rotation is returned.
- static Quaternion GetQuaternion (string name, Quaternion defaultValue, bool encrypted=false)

  Call this function to retrieve a Quaternion from the EpicPrefs. If the Pref does not exist a default Quaternion is returned.
- static void DeleteQuaternion (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static bool SetVector2 (string name, Vector2 value, bool encrypted=false)

Call this function to save a Vector2 to the EpicPrefs. See the parameters for more information on what to pass.

- static Vector2 GetVector2 (string name, bool encrypted=false)

  Call this function to retrieve a Vector2 from the EpicPrefs. If the Pref does not exist a zero Vector2 is returned.
- static Vector2 GetVector2 (string name, Vector2 defaultValue, bool encrypted=false)

  Call this function to retrieve a Vector2 from the EpicPrefs. If the Pref does not exist a default Vector2 is returned.
- static void DeleteVector2 (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static bool SetVector3 (string name, Vector3 value, bool encrypted=false)

  Call this function to save a Vector3 to the EpicPrefs. See the parameters for more information on what to pass.
- static Vector3 GetVector3 (string name, bool encrypted=false)

  Call this function to retrieve a Vector3 from the EpicPrefs. If the Pref does not exist a zero Vector3 is returned.
- static Vector3 GetVector3 (string name, Vector3 defaultValue, bool encrypted=false)

  Call this function to retrieve a Vector3 from the EpicPrefs. If the Pref does not exist a default Vector3 is returned.
- static void DeleteVector3 (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static bool SetVector4 (string name, Vector4 value, bool encrypted=false)

  Call this function to save a Vector4 to the EpicPrefs. See the parameters for more information on what to pass.
- static Vector4 GetVector4 (string name, bool encrypted=false)

  Call this function to retrieve a Vector4 from the EpicPrefs. If the Pref does not exist a zero Vector4 is returned.
- static Vector4 GetVector4 (string name, Vector4 defaultValue, bool encrypted=false)

  Call this function to retrieve a Vector4 from the EpicPrefs. If the Pref does not exist a default Vector4 is returned.
- static void DeleteVector4 (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static bool SetList (string name, List< string > value, bool encrypted=false)

  Call this function to save a List of strings to the EpicPrefs. See the parameters for more information on what to pass.
- static bool SetList (string name, List< float > value, bool encrypted=false)

  Call this function to save a List of floats to the EpicPrefs. See the parameters for more information on what to pass.
- static bool SetList (string name, List< bool > value, bool encrypted=false)

  Call this function to save a List of bools to the EpicPrefs. See the parameters for more information on what to pass.
- static bool SetList (string name, List< int > value, bool encrypted=false)

  Call this function to save a List of integers to the EpicPrefs. See the parameters for more information on what to pass.
- static bool SetList (string name, List< double > value, bool encrypted=false)

  Call this function to save a List of doubles to the EpicPrefs. See the parameters for more information on what to pass.
- static bool SetList (string name, List< long > value, bool encrypted=false)

  Call this function to save a List of longs to the EpicPrefs. See the parameters for more information on what to pass.
- static List< string > GetListString (string name, bool encrypted=false)

  Call this function to retrieve a List of strings from the EpicPrefs. If the Pref does not exist a null is returned.
- static List< string > GetListString (string name, List< string > defaultValue, bool encrypted=false)

  Call this function to retrieve a List of strings from the EpicPrefs. If the Pref does not exist the default value is returned.
- static List< float > GetListFloat (string name, bool encrypted=false)

  Call this function to retrieve a List of floats from the EpicPrefs. If the Pref does not exist a null is returned.
- static List< float > GetListFloat (string name, List< float > defaultValue, bool encrypted=false)

  Call this function to retrieve a List of floats from the EpicPrefs. If the Pref does not exist the default value is returned.
- static List< double > GetListDouble (string name, bool encrypted=false)

  Call this function to retrieve a List of doubles from the EpicPrefs. If the Pref does not exist a null is returned.
- static List< double > GetListDouble (string name, List< double > defaultValue, bool encrypted=false)

  Call this function to retrieve a List of doubles from the EpicPrefs. If the Pref does not exist the default value is returned.
- static List< bool > GetListBool (string name, bool encrypted=false)

  Call this function to retrieve a List of booleans from the EpicPrefs. If the Pref does not exist a null is returned.
- static List< bool > GetListBool (string name, List< bool > defaultValue, bool encrypted=false)

Call this function to retrieve a List of booleans from the EpicPrefs. If the Pref does not exist the default value is returned.

- static List< int > GetListInt (string name, bool encrypted=false)

  Call this function to retrieve a List of integers from the EpicPrefs. If the Pref does not exist a null is returned.
- static List< int > GetListInt (string name, List< int > defaultValue, bool encrypted=false)

  Call this function to retrieve a List of integers from the EpicPrefs. If the Pref does not exist the default value is returned.
- static List< long > GetListLong (string name, bool encrypted=false)

  Call this function to retrieve a List of longs from the EpicPrefs. If the Pref does not exist a null is returned.
- static List< long > GetListLong (string name, List< long > defaultValue, bool encrypted=false)

  Call this function to retrieve a List of longs from the EpicPrefs. If the Pref does not exist the default value is returned.
- static void DeleteList (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static bool SetArray (string name, string[ ] value, bool encrypted=false)

  Call this function to save an array of strings to the EpicPrefs. See the parameters for more information on what to pass.
- static bool SetArray (string name, int[ ] value, bool encrypted=false)

  Call this function to save an array of integers to the EpicPrefs. See the parameters for more information on what to pass.
- static bool SetArray (string name, float[ ] value, bool encrypted=false)

  Call this function to save an array of floats to the EpicPrefs. See the parameters for more information on what to pass.
- static bool SetArray (string name, double[ ] value, bool encrypted=false)

  Call this function to save an array of doubles to the EpicPrefs. See the parameters for more information on what to pass.
- static string[ ] GetArrayString (string name, bool encrypted=false)

  Call this function to retrieve an Array of strings from the EpicPrefs. If the Pref does not exist a null is returned.
- static string[ ] GetArrayString (string name, string[ ] defaultValue, bool encrypted=false)

  Call this function to retrieve an Array of strings from the EpicPrefs. If the Pref does not exist the default value is returned.
- static int[ ] GetArrayInt (string name, bool encrypted=false)

  Call this function to retrieve an Array of integers from the EpicPrefs. If the Pref does not exist a null is returned.
- static int[ ] GetArrayInt (string name, int[ ] defaultValue, bool encrypted=false)

  Call this function to retrieve an Array of integers from the EpicPrefs. If the Pref does not exist the default value is returned.
- static float[ ] GetArrayFloat (string name, bool encrypted=false)

  Call this function to retrieve an Array of floats from the EpicPrefs. If the Pref does not exist a null is returned.
- static float[ ] GetArrayFloat (string name, float[ ] defaultValue, bool encrypted=false)

  Call this function to retrieve an Array of floats from the EpicPrefs. If the Pref does not exist the default value is returned.
- static double[ ] GetArrayDouble (string name, bool encrypted=false)

  Call this function to retrieve an Array of doubles from the EpicPrefs. If the Pref does not exist a null is returned.
- static double[ ] GetArrayDouble (string name, double[ ] defaultValue, bool encrypted=false)

  Call this function to retrieve an Array of doubles from the EpicPrefs. If the Pref does not exist the default value is returned.
- static void DeleteArrayFloat (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static void DeleteArrayInt (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static void DeleteArrayString (string name, bool encrypted)

  Deletes an EpicPref permanently.
- static void DeleteArrayDouble (string name, bool encrypted)

  Deletes an EpicPref permanently.

5.4.1 Detailed Description

EpicPrefs is a replacement for Unity's PlayerPrefs. Everything you can do with PlayerPrefs, you can do with EpicPrefs too. Plus, much much more.

5.4.2 Member Function Documentation

5.4.2.1 static void EpicPrefs.Initialize ( ) [static]

Initialize and setup any imported EpicPrefs.

5.4.2.2 static string EpicPrefs.getPassPhrase ( ) [static]

Retrieve the current passPhrase.

Returns

A string containing the passPhrase.

5.4.2.3 static void EpicPrefs.setPassPhrase (string *value*) [static]

Set the new passPhrase and remove the previously encrypted values if the passPhrase has changed.

Parameters

value	The new passPhrase

5.4.2.4 static string EpicPrefs.getInitVector ( ) [static]

Retrieve the current initVector.

Returns

A string containing the initVector.

5.4.2.5 static void EpicPrefs.setInitVector (string *value*) [static]

Set the new initVector and remove the previously encrypted values if the initVector has changed.

Parameters

value	The new initVector. It has to be 16 byte long!

5.4.2.6 static bool EpicPrefs.SetString (string *name*, string *value*, bool *encrypted* = false) [static]

Call this function to save a string to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.

value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.7 static string EpicPrefs.GetString (string *name,* bool *encrypted* = `false`) [static]

Call this function to retrieve a string from the EpicPrefs. If the Pref does not exist an empty string is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved string.

5.4.2.8 static string EpicPrefs.GetString (string *name,* string *defaultValue,* bool *encrypted* = `false`) [static]

Call this function to retrieve a string from the EpicPrefs. If the Pref does not exist a default string is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved string or the default value if not found.

5.4.2.9 static void EpicPrefs.DeleteString (string *name,* bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.10 static bool EpicPrefs.SetInt (string *name,* int *value,* bool *encrypted* = `false`) [static]

Call this function to save an integer to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.

encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.11 static int EpicPrefs.GetInt (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve an integer from the EpicPrefs. If the Pref does not exist -1 is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved integer.

5.4.2.12 static int EpicPrefs.GetInt (string *name*, int *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve an integer from the EpicPrefs. If the Pref does not exist a default integer is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved integer or the default value if not found.

5.4.2.13 static void EpicPrefs.DeleteInt (string *name*, bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.14 static bool EpicPrefs.SetBool (string *name*, bool *value*, bool *encrypted* = `false`) [static]

Call this function to save a bool to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.15 static bool EpicPrefs.GetBool (string *name,* bool *encrypted* = `false`) [static]

Call this function to retrieve a boolean from the EpicPrefs. If the Pref does not exist false is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved boolean.

5.4.2.16 static bool EpicPrefs.GetBool (string *name*, bool *defaultValue*, bool *encrypted*) [static]

Call this function to retrieve a boolean from the EpicPrefs. If the Pref does not exist a default bool is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved boolean or the default value if not found.

5.4.2.17 static void EpicPrefs.DeleteBool (string *name*, bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.18 static bool EpicPrefs.SetFloat (string *name*, float *value*, bool *encrypted* = false) [static]

Call this function to save a float to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.19 static float EpicPrefs.GetFloat (string *name*, bool *encrypted* = false) [static]

Call this function to retrieve a float from the EpicPrefs. If the Pref does not exist -1 is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved float.

5.4.2.20 static float EpicPrefs.GetFloat (string *name*, float *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a float from the EpicPrefs. If the Pref does not exist a default float is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved float or the default value if not found.

5.4.2.21 static void EpicPrefs.DeleteFloat (string *name*, bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.22 static bool EpicPrefs.SetLong (string *name*, long *value*, bool *encrypted* = `false`) [static]

Call this function to save a long to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.23 static long EpicPrefs.GetLong (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a long from the EpicPrefs. If the Pref does not exist -1 is returned.

Parameters

name	The name under which the EpicPref has previously been saved.

encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved long.

5.4.2.24 static long EpicPrefs.GetLong (string *name*, long *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a long from the EpicPrefs. If the Pref does not exist a default long is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved long or the default value if not found.

5.4.2.25 static void EpicPrefs.DeleteLong (string *name*, bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.26 static bool EpicPrefs.SetDouble (string *name*, double *value*, bool *encrypted* = `false`) [static]

Call this function to save a double to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.27 static double EpicPrefs.GetDouble (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a double from the EpicPrefs. If the Pref does not exist -1 is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved double.

5.4.2.28 static double EpicPrefs.GetDouble (string *name*, double *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a double from the EpicPrefs. If the Pref does not exist a default double is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved double or the default value if not found.

5.4.2.29 static void EpicPrefs.DeleteDouble (string *name,* bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.30 static bool EpicPrefs.SetDict (string *name,* Dictionary< string, string > *value,* bool *encrypted* = false) [static]

Call this function to save a Dictionary with its Keys being a string and Values being a string to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.31 static bool EpicPrefs.SetDict (string *name,* Dictionary< string, int > *value,* bool *encrypted* = false) [static]

Call this function to save a Dictionary with its Keys being a string and Values being an int to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.32 static bool EpicPrefs.SetDict (string *name,* Dictionary< string, float > *value,* bool *encrypted* = false) [static]

Call this function to save a Dictionary with its Keys being a string and Values being a float to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.33 static bool EpicPrefs.SetDict (string *name*, Dictionary< string, long > *value*, bool *encrypted* = `false`) [static]

Call this function to save a Dictionary with its Keys being a string and Values being a long to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.34 static bool EpicPrefs.SetDict (string *name*, Dictionary< string, double > *value*, bool *encrypted* = `false`) [static]

Call this function to save a Dictionary with its Keys being a string and Values being a double to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.35 static bool EpicPrefs.SetDict (string *name*, Dictionary< string, bool > *value*, bool *encrypted* = `false`) [static]

Call this function to save a Dictionary with its Keys being a string and Values being a bool to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.

encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.36 static Dictionary<string, string> EpicPrefs.GetDictStringString (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Dictionary with its Keys being strings and its Values being strings from the EpicPrefs. If the Pref does not exist null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Dictionary.

5.4.2.37 static Dictionary<string, string> EpicPrefs.GetDictStringString (string *name*, Dictionary< string, string > *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Dictionary with its Keys being strings and its Values being strings from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default value to be returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Dictionary.

5.4.2.38 static Dictionary<string, int> EpicPrefs.GetDictStringInt (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Dictionary with its Keys being strings and its Values being integers from the EpicPrefs. If the Pref does not exist null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Dictionary.

5.4.2.39 static Dictionary<string, int> EpicPrefs.GetDictStringInt (string *name*, Dictionary< string, int > *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Dictionary with its Keys being strings and its Values being strings from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default value to be returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Dictionary.

5.4.2.40 static Dictionary<string, float> EpicPrefs.GetDictStringFloat (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Dictionary with its Keys being strings and its Values being floats from the EpicPrefs. If the Pref does not exist null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Dictionary.

5.4.2.41 static Dictionary<string, float> EpicPrefs.GetDictStringFloat (string *name*, Dictionary< string, float > *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Dictionary with its Keys being strings and its Values being floats from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default value to be returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Dictionary.

5.4.2.42 static Dictionary<string, double> EpicPrefs.GetDictStringDouble (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Dictionary with its Keys being strings and its Values being doubles from the EpicPrefs. If the Pref does not exist null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Dictionary.

5.4.2.43 static Dictionary<string, double> EpicPrefs.GetDictStringDouble (string *name*, Dictionary< string, double > *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Dictionary with its Keys being strings and its Values being doubles from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default value to be returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Dictionary.

5.4.2.44 static Dictionary<string, bool> EpicPrefs.GetDictStringBool (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Dictionary with its Keys being strings and its Values being booleans from the EpicPrefs. If the Pref does not exist null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Dictionary.

5.4.2.45 static Dictionary<string, bool> EpicPrefs.GetDictStringBool (string *name*, Dictionary< string, bool > *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Dictionary with its Keys being strings and its Values being booleans from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default value to be returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Dictionary.

5.4.2.46 static Dictionary<string, long> EpicPrefs.GetDictStringLong (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Dictionary with its Keys being strings and its Values being longs from the EpicPrefs. If the Pref does not exist null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Dictionary.

5.4.2.47 static Dictionary<string, long> EpicPrefs.GetDictStringLong (string *name,* Dictionary< string, long > *defaultValue,* bool *encrypted* = `false`) [static]

Call this function to retrieve a Dictionary with its Keys being strings and its Values being longs from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default value to be returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Dictionary.

5.4.2.48 static void EpicPrefs.DeleteDict (string *name*, bool *encrypted*) [static]

Deletes a Dictionary permanently.

Parameters

name	The name of the dictionary.
encrypted	Whether it was encrypted or not.

5.4.2.49 static bool EpicPrefs.SetColor (string *name*, Color *value*, bool *encrypted* = false) [static]

Call this function to save a Color to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.50 static Color EpicPrefs.GetColor (string *name*, bool *encrypted* = false) [static]

Call this function to retrieve a color from the EpicPrefs. If the Pref does not exist white is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved color.

5.4.2.51 static Color EpicPrefs.GetColor (string *name*, Color *defaultValue*, bool *encrypted* = false) [static]

Call this function to retrieve a color from the EpicPrefs. If the Pref does not exist a default color is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved color or the default value if not found.

5.4.2.52 static void EpicPrefs.DeleteColor (string *name*, bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.53 static bool EpicPrefs.SetQuaternion (string *name*, Quaternion *value*, bool *encrypted* = false) [static]

Call this function to save a Quaternion to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.54 static Quaternion EpicPrefs.GetQuaternion (string *name*, bool *encrypted* = false) [static]

Call this function to retrieve a Quaternion from the EpicPrefs. If the Pref does not exist a zero rotation is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Quaternion.

5.4.2.55 static Quaternion EpicPrefs.GetQuaternion (string *name*, Quaternion *defaultValue*, bool *encrypted* = false) [static]

Call this function to retrieve a Quaternion from the EpicPrefs. If the Pref does not exist a default Quaternion is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Quaternion or the default value if not found.

5.4.2.56 static void EpicPrefs.DeleteQuaternion (string *name*, bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.57 static bool EpicPrefs.SetVector2 (string *name*, Vector2 *value*, bool *encrypted* = `false`) [static]

Call this function to save a Vector2 to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.58 static Vector2 EpicPrefs.GetVector2 (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Vector2 from the EpicPrefs. If the Pref does not exist a zero Vector2 is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Vector2.

5.4.2.59 static Vector2 EpicPrefs.GetVector2 (string *name*, Vector2 *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Vector2 from the EpicPrefs. If the Pref does not exist a default Vector2 is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Vector2 or the default value if not found.

5.4.2.60 static void EpicPrefs.DeleteVector2 (string *name*, bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.61 static bool EpicPrefs.SetVector3 (string *name*, Vector3 *value*, bool *encrypted* = `false`) [static]

Call this function to save a Vector3 to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.62 static Vector3 EpicPrefs.GetVector3 (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Vector3 from the EpicPrefs. If the Pref does not exist a zero Vector3 is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Vector3.

5.4.2.63 static Vector3 EpicPrefs.GetVector3 (string *name*, Vector3 *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Vector3 from the EpicPrefs. If the Pref does not exist a default Vector3 is returned.

Parameters

name	The name under which the EpicPref has previously been saved.

defaultValue	The default returned if the EpicPref is not found.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Vector3 or the default value if not found.

5.4.2.64 static void EpicPrefs.DeleteVector3 (string *name*, bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.65 static bool EpicPrefs.SetVector4 (string *name*, Vector4 *value*, bool *encrypted* = `false`) [static]

Call this function to save a Vector4 to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.66 static Vector4 EpicPrefs.GetVector4 (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Vector4 from the EpicPrefs. If the Pref does not exist a zero Vector4 is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Vector4.

5.4.2.67 static Vector4 EpicPrefs.GetVector4 (string *name*, Vector4 *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a Vector4 from the EpicPrefs. If the Pref does not exist a default Vector4 is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default returned if the EpicPref is not found.

encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Vector4 or the default value if not found.

5.4.2.68 static void EpicPrefs.DeleteVector4 (string *name*, bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.69 static bool EpicPrefs.SetList (string *name*, List< string > *value*, bool *encrypted* = false) [static]

Call this function to save a List of strings to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.70 static bool EpicPrefs.SetList (string *name*, List< float > *value*, bool *encrypted* = false) [static]

Call this function to save a List of floats to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.71 static bool EpicPrefs.SetList (string *name*, List< bool > *value*, bool *encrypted* = false) [static]

Call this function to save a List of bools to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.

encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.72 static bool EpicPrefs.SetList (string *name*, List< int > *value*, bool *encrypted* = false) [static]

Call this function to save a List of integers to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.73 static bool EpicPrefs.SetList (string *name*, List< double > *value*, bool *encrypted* = false) [static]

Call this function to save a List of doubles to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.74 static bool EpicPrefs.SetList (string *name*, List< long > *value*, bool *encrypted* = false) [static]

Call this function to save a List of longs to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.75 static List<string> EpicPrefs.GetListString (string *name*, bool *encrypted* = false) [static]

Call this function to retrieve a List of strings from the EpicPrefs. If the Pref does not exist a null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved List of strings.

5.4.2.76 static List<string> EpicPrefs.GetListString (string *name*, List< string > *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a List of strings from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default value that will be returned if the EpicPref does not exist.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved List of strings.

5.4.2.77 static List<float> EpicPrefs.GetListFloat (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a List of floats from the EpicPrefs. If the Pref does not exist a null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved List of floats.

5.4.2.78 static List<float> EpicPrefs.GetListFloat (string *name*, List< float > *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a List of floats from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default value that will be returned if the EpicPref does not exist.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved List of floats.

5.4.2.79 static List<double> EpicPrefs.GetListDouble (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a List of doubles from the EpicPrefs. If the Pref does not exist a null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved List of doubles.

5.4.2.80 static List<double> EpicPrefs.GetListDouble (string *name*, List< double > *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a List of doubles from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default value that will be returned if the EpicPref does not exist.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved List of doubles.

5.4.2.81 static List<bool> EpicPrefs.GetListBool (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a List of booleans from the EpicPrefs. If the Pref does not exist a null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved List of booleans.

5.4.2.82 static List<bool> EpicPrefs.GetListBool (string *name*, List< bool > *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a List of booleans from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default value that will be returned if the EpicPref does not exist.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved List of booleans.

5.4.2.83 static List<int> EpicPrefs.GetListInt (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a List of integers from the EpicPrefs. If the Pref does not exist a null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved List of integers.

5.4.2.84 static List<int> EpicPrefs.GetListInt (string *name*, List< int > *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a List of integers from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default value that will be returned if the EpicPref does not exist.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved List of integers.

5.4.2.85 static List<long> EpicPrefs.GetListLong (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve a List of longs from the EpicPrefs. If the Pref does not exist a null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved List of longs.

5.4.2.86 static List<long> EpicPrefs.GetListLong (string *name*, List< long > *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve a List of longs from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	The default value that will be returned if the EpicPref does not exist.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved List of longs.

5.4.2.87 static void EpicPrefs.DeleteList (string *name*, bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.88 static bool EpicPrefs.SetArray (string *name*, string[ ] *value*, bool *encrypted* = `false`) [static]

Call this function to save an array of strings to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.89 static bool EpicPrefs.SetArray (string *name*, int[ ] *value*, bool *encrypted* = `false`) [static]

Call this function to save an array of integers to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.90 static bool EpicPrefs.SetArray (string *name*, float[ ] *value*, bool *encrypted* = `false`) [static]

Call this function to save an array of floats to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.91 static bool EpicPrefs.SetArray (string *name*, double[ ] *value*, bool *encrypted* = `false`) [static]

Call this function to save an array of doubles to the EpicPrefs. See the parameters for more information on what to pass.

Parameters

name	The name under which the pref is saved, and can be retrieved with.
value	The value that is saved.
encrypted	Whether to use encryption or not. Default is false and can be left away in that case.

Returns

Returns a bool stating if the saving has been successfull.

5.4.2.92 static string [ ] EpicPrefs.GetArrayString (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve an Array of strings from the EpicPrefs. If the Pref does not exist a null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Array of strings.

5.4.2.93 static string [ ] EpicPrefs.GetArrayString (string *name*, string[ ] *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve an Array of strings from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Array of strings.

5.4.2.94 static int [ ] EpicPrefs.GetArrayInt (string *name*, bool *encrypted* = `false`) [static]

Call this function to retrieve an Array of integers from the EpicPrefs. If the Pref does not exist a null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Array of integers.

5.4.2.95 static int [ ] EpicPrefs.GetArrayInt (string *name*, int[ ] *defaultValue*, bool *encrypted* = `false`) [static]

Call this function to retrieve an Array of integers from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Array of integers.

5.4.2.96 static float [ ] EpicPrefs.GetArrayFloat (string *name,* bool *encrypted* = `false`) [static]

Call this function to retrieve an Array of floats from the EpicPrefs. If the Pref does not exist a null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Array of floats.

5.4.2.97 static float [ ] EpicPrefs.GetArrayFloat (string *name,* float[ ] *defaultValue,* bool *encrypted* = `false`) [static]

Call this function to retrieve an Array of floats from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Array of floats.

5.4.2.98 static double [ ] EpicPrefs.GetArrayDouble (string *name,* bool *encrypted* = `false`) [static]

Call this function to retrieve an Array of doubles from the EpicPrefs. If the Pref does not exist a null is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Array of doubles.

5.4.2.99 static double [ ] EpicPrefs.GetArrayDouble (string *name,* double[ ] *defaultValue,* bool *encrypted* = `false`) [static]

Call this function to retrieve an Array of doubles from the EpicPrefs. If the Pref does not exist the default value is returned.

Parameters

name	The name under which the EpicPref has previously been saved.
defaultValue	
encrypted	Whether the EpicPref was previoulsy encrypted or not.

Returns

Returns the previously saved Array of doubles.

5.4.2.100 static void EpicPrefs.DeleteArrayFloat (string *name,* bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.101 static void EpicPrefs.DeleteArrayInt (string *name,* bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.102 static void EpicPrefs.DeleteArrayString (string *name,* bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

name	The name of the pref.
encrypted	Whether it was encrypted or not.

5.4.2.103 static void EpicPrefs.DeleteArrayDouble (string *name,* bool *encrypted*) [static]

Deletes an EpicPref permanently.

Parameters

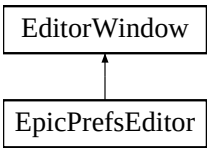
name	The name of the pref.
encrypted	Whether it was encrypted or not.

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Plugins/Hot↩ TotemAssets/EpicPrefs/Code/EpicPrefs.cs

5.5 EpicPrefsEditor Class Reference

Inheritance diagram for EpicPrefsEditor:



Public Member Functions

- void **OnInspectorUpdate** ()

Static Public Member Functions

- static void **SetupStyles** ()
- static void **Separator** ()

Public Attributes

- Font **passedFont**

Static Public Attributes

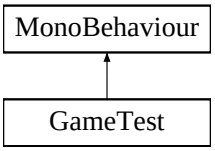
- static Color **backgroundColor** = new Color(51f / 255f, 77f / 255f, 92f / 255f)

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Editor/Hot↩ TotemAssets/EpicPrefs/EpicPrefsEditor.cs

5.6 GameTest Class Reference

Inheritance diagram for GameTest:



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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/HotTotem↩ Assets/EpicPrefs/Demo/GameTest.cs

5.7 HotTotem.Key Class Reference

Static Public Member Functions

- static string **getKey** ()
- static void **setKey** (string value)

- static string **getVector** ()
- static void **setVector** (string value)

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Plugins/Hot↩ TotemAssets/EpicPrefs/Code/Handlers/Key.cs

5.8 Operators Class Reference

Static Public Member Functions

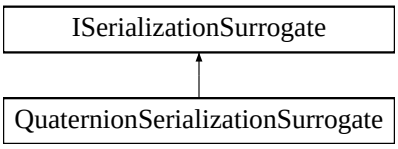
- static bool **IsInteger** (string sValue)
- static bool **IsFloat** (string sValue)
- static bool **ToBool** (string value)
- static float **ToFloat** (string value)
- static double **ToDouble** (string value)
- static int **ToInt** (string value)
- static long **ToLong** (string value)
- static Color **StringToColor** (string value)
- static string **ColorToString** (Color value)
- static void **DirectoryCopy** (string sourceDirName, string destDirName, bool copySubDirs)
- static void **DeleteDirectory** (string path, bool recursively)
- static void **setupPrefs** ()

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Plugins/Hot↩ TotemAssets/EpicPrefs/Code/Helpers/Operators.cs

5.9 QuaternionSerializationSurrogate Class Reference

Inheritance diagram for QuaternionSerializationSurrogate:



Public Member Functions

- void **GetObjectData** (System.Object obj, SerializationInfo info, StreamingContext context)
- System.Object **SetObjectData** (System.Object obj, SerializationInfo info, StreamingContext context, I↩ SurrogateSelector selector)

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Plugins/Hot↩ TotemAssets/EpicPrefs/Code/Handlers/CustomSurrogates/QuaternionSerializationSurrogate.cs

5.10 Serializer Class Reference

Public Types

- enum **SerializationTypes** {
  Integer, **String**, **Float**, **Long**,
  Double, **Bool**, **Vector2**, **Vector3**,
  Vector4, **List**, **Dict**, **Transform**,
  Quaternion, **Color**, **ArrayString**, **ArrayInt**,
  ArrayFloat, **ArrayDouble**, **Editor**, **DictS**,
  DictI, **DictB**, **DictL**, **DictD**,
  DictF, **ListS**, **ListI**, **ListB**,
  ListL, **ListD**, **ListF** }

Static Public Member Functions

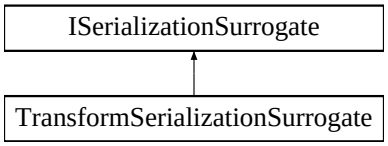
- static void **ReInitialize** ()
- static bool **Serialize** (string name, object value, SerializationTypes type, bool encrypted)
- static object **Deserialize** (string name, SerializationTypes type, bool encrypted)
- static bool **Delete** (string name, SerializationTypes type, bool encrypted)

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Plugins/Hot↩ TotemAssets/EpicPrefs/Code/Handlers/Serializer.cs

5.11 TransformSerializationSurrogate Class Reference

Inheritance diagram for TransformSerializationSurrogate:



Public Member Functions

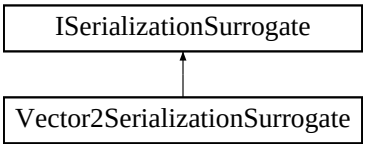
- void **GetObjectData** (System.Object obj, SerializationInfo info, StreamingContext context)
- System.Object **SetObjectData** (System.Object obj, SerializationInfo info, StreamingContext context, I↩ SurrogateSelector selector)

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Plugins/Hot↩ TotemAssets/EpicPrefs/Code/Handlers/CustomSurrogates/TransfromSerializationSurrogate.cs

5.12 Vector2SerializationSurrogate Class Reference

Inheritance diagram for Vector2SerializationSurrogate:



Public Member Functions

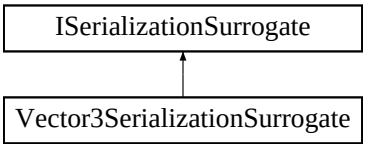
- void **GetObjectData** (System.Object obj, SerializationInfo info, StreamingContext context)
- System.Object **SetObjectData** (System.Object obj, SerializationInfo info, StreamingContext context, ISurrogateSelector selector)

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Plugins/HotTotemAssets/EpicPrefs/Code/Handlers/CustomSurrogates/VectorSerializationSurrogate.cs

5.13 Vector3SerializationSurrogate Class Reference

Inheritance diagram for Vector3SerializationSurrogate:



Public Member Functions

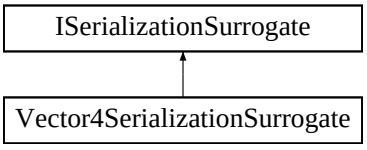
- void **GetObjectData** (System.Object obj, SerializationInfo info, StreamingContext context)
- System.Object **SetObjectData** (System.Object obj, SerializationInfo info, StreamingContext context, ISurrogateSelector selector)

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Plugins/HotTotemAssets/EpicPrefs/Code/Handlers/CustomSurrogates/VectorSerializationSurrogate.cs

5.14 Vector4SerializationSurrogate Class Reference

Inheritance diagram for Vector4SerializationSurrogate:



Public Member Functions

- void **GetObjectData** (System.Object obj, SerializationInfo info, StreamingContext context)
- System.Object **SetObjectData** (System.Object obj, SerializationInfo info, StreamingContext context, ISurrogateSelector selector)

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Plugins/HotTotemAssets/EpicPrefs/Code/Handlers/CustomSurrogates/VectorSerializationSurrogate.cs

5.15 dotBunny.Unity.VSCode Class Reference

Static Public Member Functions

- static void SyncSolution ()

  Force Unity To Write Project File
- static void UpdateSolution ()

  Update the solution files so that they work with VS Code

Public Attributes

- const float Version = 2.45f

  Current Version Number
- const string VersionCode = "-RELEASE"

  Current Version Code
- const string UnityDebuggerURL = "https://raw.githubusercontent.com/dotBunny/VSCode-Test/master/Downloads/unity-debug-101.vsix"

  Download URL for Unity Debbuger

Properties

- static bool Debug `[get, set]`

  Should debug information be displayed in the Unity terminal?
- static bool Enabled `[get, set]`

  Is the Visual Studio Code Integration Enabled?
- static bool **UseUnityDebugger** `[get, set]`
- static bool WriteLaunchFile `[get, set]`

  Should the launch.json file be written?

5.15.1 Member Function Documentation

5.15.1.1 static void dotBunny.Unity.VSCode.SyncSolution ( ) [static]

Force Unity To Write Project File

Reflection!

5.15.1.2 static void dotBunny.Unity.VSCode.UpdateSolution ( ) [static]

Update the solution files so that they work with VS Code

5.15.2 Member Data Documentation

5.15.2.1 const float dotBunny.Unity.VSCode.Version = 2.45f

Current Version Number

5.15.2.2 const string dotBunny.Unity.VSCode.VersionCode = "-RELEASE"

Current Version Code

5.15.2.3 const string dotBunny.Unity.VSCode.UnityDebuggerURL = "https://raw.githubusercontent.com/dotBunny/VSCode-Test/master/Downloads/unity-debug-101.vsix"

Download URL for Unity Debbuger

5.15.3 Property Documentation

5.15.3.1 bool dotBunny.Unity.VSCode.Debug [static], [get], [set]

Should debug information be displayed in the Unity terminal?

5.15.3.2 bool dotBunny.Unity.VSCode.Enabled [static], [get], [set]

Is the Visual Studio Code Integration Enabled?

We do not want to automatically turn it on, for in larger projects not everyone is using VSCode

5.15.3.3 bool dotBunny.Unity.VSCode.WriteLaunchFile [static], [get], [set]

Should the launch.json file be written?

Useful to disable if someone has their own custom one rigged up

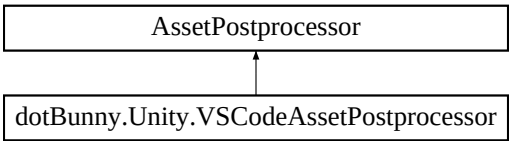
The documentation for this class was generated from the following file:

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Plugins/↩ Editor/VSCode.cs

5.16 dotBunny.Unity.VSCodeAssetPostprocessor Class Reference

VSCode Asset AssetPostprocessor

Inheritance diagram for dotBunny.Unity.VSCodeAssetPostprocessor:



5.16.1 Detailed Description

VSCode Asset AssetPostprocessor

This will ensure any time that the project files are generated the VSCode versions will be made

Undocumented Event

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

- C:/Users/user/Documents/Unity/Assets/RetailProjects/EpicPrefs/epicprefs/EpicPrefs/Assets/Plugins/↩ Editor/VSCode.cs

Index