

## Core Manager

1.1.3

Generated by Doxygen 1.9.1



<b>1 CoreManager</b>	<b>1</b>
1.1 Technical details	1
1.2 Features	1
1.3 Improvements	2
1.4 Breaking changes	2
1.4.1 v1.1.2	2
1.4.2 v1.1.3	2
<b>2 Namespace Index</b>	<b>3</b>
2.1 Namespace List	3
<b>3 Hierarchical Index</b>	<b>5</b>
3.1 Class Hierarchy	5
<b>4 Class Index</b>	<b>7</b>
4.1 Class List	7
<b>5 Namespace Documentation</b>	<b>11</b>
5.1 CorePlugin Namespace Reference	11
5.2 CorePlugin.Attributes Namespace Reference	11
5.3 CorePlugin.Attributes.Editor Namespace Reference	11
5.4 CorePlugin.Attributes.Editor.Drawers Namespace Reference	11
5.5 CorePlugin.Attributes.Headers Namespace Reference	11
5.6 CorePlugin.Attributes.Validation Namespace Reference	12
5.7 CorePlugin.Attributes.Validation.Base Namespace Reference	12
5.8 CorePlugin.Console Namespace Reference	12
5.9 CorePlugin.Console.ConsoleElements Namespace Reference	13
5.10 CorePlugin.Core Namespace Reference	13
5.11 CorePlugin.Core.Interface Namespace Reference	13
5.12 CorePlugin.Cross Namespace Reference	13
5.13 CorePlugin.Cross.Events Namespace Reference	13
5.14 CorePlugin.Cross.Events.Interface Namespace Reference	14
5.15 CorePlugin.Cross.SceneData Namespace Reference	14
5.16 CorePlugin.Cross.SceneData.Interface Namespace Reference	14
5.17 CorePlugin.Dispatchers Namespace Reference	14
5.18 CorePlugin.Editor Namespace Reference	14
5.19 CorePlugin.Extensions Namespace Reference	15
5.20 CorePlugin.FileSystem Namespace Reference	15
5.21 CorePlugin.Logger Namespace Reference	15
5.22 CorePlugin.ReferenceDistribution Namespace Reference	15
5.23 CorePlugin.ReferenceDistribution.Interface Namespace Reference	15
5.24 CorePlugin.Samples Namespace Reference	16
5.25 CorePlugin.Samples.Scripts Namespace Reference	16
5.26 CorePlugin.Samples.Scripts.Demo Namespace Reference	16

5.27 CorePlugin.Samples.Scripts.EventTypes Namespace Reference . . . . .	16
5.28 CorePlugin.Samples.Scripts.Managers Namespace Reference . . . . .	16
5.29 CorePlugin.Samples.Scripts.Model Namespace Reference . . . . .	16
5.30 CorePlugin.SceneManagement Namespace Reference . . . . .	16
5.31 CorePlugin.Serializable Namespace Reference . . . . .	16
5.32 CorePlugin.Serializable.Interface Namespace Reference . . . . .	16
5.33 CorePlugin.Singletons Namespace Reference . . . . .	17
5.34 CorePlugin.UISystem Namespace Reference . . . . .	17
5.35 CorePlugin.UISystem.UI Namespace Reference . . . . .	17
<b>6 Class Documentation</b>	<b>19</b>
6.1 BaseCore Class Reference . . . . .	19
6.1.1 Detailed Description . . . . .	19
6.1.2 Member Function Documentation . . . . .	19
6.1.2.1 InitializeElements() . . . . .	20
6.2 ClassValidationAttribute Class Reference . . . . .	20
6.2.1 Detailed Description . . . . .	20
6.3 ConsoleIcons Class Reference . . . . .	20
6.3.1 Detailed Description . . . . .	21
6.4 ConsoleInitializer Class Reference . . . . .	21
6.4.1 Detailed Description . . . . .	21
6.5 ConsoleMessage Class Reference . . . . .	21
6.5.1 Detailed Description . . . . .	22
6.5.2 Member Function Documentation . . . . .	22
6.5.2.1 ClearHighlight() . . . . .	22
6.5.2.2 HighlightText() . . . . .	22
6.5.2.3 Initialize() . . . . .	22
6.5.2.4 SetActive() . . . . .	23
6.5.2.5 SubscribeOnButtonClick() . . . . .	23
6.6 ConsoleTextSettings Class Reference . . . . .	23
6.6.1 Detailed Description . . . . .	24
6.7 CoreManager Class Reference . . . . .	24
6.7.1 Detailed Description . . . . .	24
6.8 CoreManagerEditor Class Reference . . . . .	24
6.8.1 Detailed Description . . . . .	25
6.9 CountDisplayer Class Reference . . . . .	25
6.9.1 Detailed Description . . . . .	25
6.9.2 Member Function Documentation . . . . .	25
6.9.2.1 Initialize() . . . . .	25
6.9.2.2 OnLogCountChanged() . . . . .	26
6.9.2.3 SetInteractionAction() . . . . .	26
6.10 DebugLogger Class Reference . . . . .	27

6.10.1 Detailed Description	27
6.11 EventInitializer Class Reference	27
6.11.1 Detailed Description	27
6.11.2 Member Function Documentation	28
6.11.2.1 AddHandler()	28
6.11.2.2 InitializeSubscriptions()	28
6.11.2.3 InvokeBase()	28
6.11.2.4 RemoveHandler()	28
6.11.2.5 Subscribe()	29
6.11.2.6 Unsubscribe()	29
6.12 EventTypes Class Reference	29
6.12.1 Detailed Description	29
6.13 FieldValidationAttribute Class Reference	30
6.13.1 Detailed Description	30
6.14 FileSystem Class Reference	30
6.14.1 Detailed Description	31
6.14.2 Constructor & Destructor Documentation	31
6.14.2.1 FileSystem() [1/3]	31
6.14.2.2 FileSystem() [2/3]	31
6.14.2.3 FileSystem() [3/3]	31
6.14.3 Member Function Documentation	32
6.14.3.1 Load< T >() [1/2]	32
6.14.3.2 Load< T >() [2/2]	32
6.14.3.3 Save< T >() [1/2]	33
6.14.3.4 Save< T >() [2/2]	33
6.15 HasComponentAttribute Class Reference	34
6.15.1 Detailed Description	34
6.16 ICore Interface Reference	34
6.16.1 Detailed Description	35
6.16.2 Member Function Documentation	35
6.16.2.1 InitializeElements()	35
6.17 IDistributingReference Interface Reference	35
6.17.1 Detailed Description	35
6.18 IEventHandler Interface Reference	35
6.18.1 Detailed Description	36
6.18.2 Member Function Documentation	36
6.18.2.1 InvokeEvents()	36
6.18.2.2 Subscribe()	36
6.18.2.3 Unsubscribe()	37
6.19 IEventSubscriber Interface Reference	37
6.19.1 Detailed Description	37
6.19.2 Member Function Documentation	38

6.19.2.1 GetSubscribers()	38
6.20 ISceneData Interface Reference	38
6.20.1 Detailed Description	38
6.21 ItemHider Class Reference	38
6.21.1 Detailed Description	39
6.22 IUnique Interface Reference	39
6.22.1 Detailed Description	39
6.23 LoadSceneOptions Class Reference	39
6.23.1 Detailed Description	39
6.24 MinimizeCountDisplay Class Reference	40
6.24.1 Detailed Description	40
6.24.2 Member Function Documentation	40
6.24.2.1 Initialize()	40
6.24.2.2 SetInteractionAction()	41
6.25 MinimizedConsole Class Reference	41
6.25.1 Detailed Description	41
6.25.2 Member Function Documentation	41
6.25.2.1 Initialize()	41
6.25.2.2 SetActive()	42
6.26 MovableInsideScreen Class Reference	42
6.26.1 Detailed Description	42
6.27 Named< TName, TKey, TValue > Class Template Reference	42
6.27.1 Detailed Description	42
6.28 NotNullAttribute Class Reference	43
6.28.1 Detailed Description	43
6.29 OneAndOnlyAttribute Class Reference	43
6.29.1 Detailed Description	44
6.30 PrefabHeaderAttribute Class Reference	44
6.30.1 Detailed Description	44
6.31 PrefabRequiredAttribute Class Reference	44
6.31.1 Detailed Description	44
6.32 ReferenceDistributor Class Reference	45
6.32.1 Detailed Description	45
6.32.2 Member Function Documentation	45
6.32.2.1 AskReference< T >()	45
6.32.2.2 GetInterfaceReference< T >()	46
6.32.2.3 GetInterfaceReferences< T >()	46
6.32.2.4 GetReference< T >()	47
6.32.2.5 GetReferences< T >()	47
6.32.2.6 Initialize()	48
6.33 ReferencesHeaderAttribute Class Reference	48
6.33.1 Detailed Description	48

6.34 RequireInterfaceAttribute Class Reference	48
6.34.1 Detailed Description	49
6.34.2 Constructor & Destructor Documentation	49
6.34.2.1 RequireInterfaceAttribute()	49
6.35 RequireInterfaceDrawer Class Reference	49
6.35.1 Detailed Description	49
6.35.2 Member Function Documentation	49
6.35.2.1 OnGUI()	49
6.36 RuntimeConsole Class Reference	50
6.36.1 Detailed Description	50
6.36.2 Member Function Documentation	50
6.36.2.1 Initialize()	50
6.36.2.2 SetActive()	51
6.37 SceneDataHandler Class Reference	51
6.37.1 Detailed Description	52
6.37.2 Member Function Documentation	52
6.37.2.1 AddData< T >()	52
6.37.2.2 GetData< T >()	52
6.37.2.3 RemoveData< T >()	53
6.38 SceneLoader Class Reference	53
6.38.1 Detailed Description	53
6.38.2 Member Function Documentation	54
6.38.2.1 LoadSceneAsync() [1/2]	54
6.38.2.2 LoadSceneAsync() [2/2]	54
6.39 SceneLoaderAsset Class Reference	54
6.39.1 Detailed Description	55
6.40 SceneLoaderExtensions Class Reference	55
6.40.1 Detailed Description	55
6.40.2 Member Function Documentation	55
6.40.2.1 SceneLoadOperation()	55
6.40.2.2 SceneUnloadOperation() [1/2]	56
6.40.2.3 SceneUnloadOperation() [2/2]	56
6.40.2.4 Until()	57
6.41 SceneLoaderSettings Class Reference	57
6.41.1 Detailed Description	58
6.42 SceneLoaderSettingsEditor Class Reference	58
6.42.1 Detailed Description	58
6.43 SceneLoaderSettingsValidator Class Reference	58
6.43.1 Detailed Description	59
6.44 SettingsHeaderAttribute Class Reference	59
6.44.1 Detailed Description	59
6.45 Singleton< T > Class Template Reference	59

6.45.1 Detailed Description	59
6.46 StateHeaderAttribute Class Reference	60
6.46.1 Detailed Description	60
6.47 StaticObjectSingleton< T > Class Template Reference	60
6.47.1 Detailed Description	60
6.48 SubPagesController Class Reference	60
6.48.1 Detailed Description	61
6.48.2 Member Function Documentation	61
6.48.2.1 AddPage() [1/2]	61
6.48.2.2 AddPage() [2/2]	61
6.48.2.3 DelayedOpenPage() [1/2]	62
6.48.2.4 DelayedOpenPage() [2/2]	62
6.48.2.5 HideAllTables()	62
6.48.2.6 OpenPage()	63
6.49 SymbolDefiner Class Reference	63
6.49.1 Detailed Description	63
6.49.2 Member Function Documentation	63
6.49.2.1 OnEnable()	64
6.49.2.2 ShowSymbolsButtons()	64
6.50 ToggleCountDisplayer Class Reference	64
6.50.1 Detailed Description	64
6.50.2 Member Function Documentation	64
6.50.2.1 Initialize()	64
6.50.2.2 SetInteractionAction()	65
6.51 UIManager Class Reference	65
6.51.1 Detailed Description	66
6.51.2 Member Function Documentation	66
6.51.2.1 InitializeElements()	66
6.52 UIPage Class Reference	66
6.52.1 Detailed Description	66
6.52.2 Member Function Documentation	66
6.52.2.1 Initialize()	67
6.53 UIStateTools Class Reference	67
6.53.1 Detailed Description	67
6.53.2 Member Function Documentation	67
6.53.2.1 ChangeCursorState()	67
6.53.2.2 ChangeGroupState() [1/2]	68
6.53.2.3 ChangeGroupState() [2/2]	68
6.54 Unique Class Reference	68
6.54.1 Detailed Description	69
6.55 UnityEditorExtension Class Reference	69
6.55.1 Detailed Description	69



6.55.2 Member Function Documentation	69
6.55.2.1 HelpBox() [1/2]	69
6.55.2.2 HelpBox() [2/2]	70
6.55.2.3 IconName()	70
6.56 UnityExtensions Class Reference	70
6.56.1 Detailed Description	71
6.56.2 Member Function Documentation	71
6.56.2.1 Clear< T >()	71
6.56.2.2 CopyToClipboard()	72
6.56.2.3 IsNotNullAndEqual()	72
6.56.2.4 IsNotNullAndNotEqual()	72
6.56.2.5 RemoveRange< T >()	73
6.56.2.6 SnapToLatest()	73
6.56.2.7 TryFindObjectOfType< T >()	73
6.56.2.8 TryFindObjectsOfType< T >() [1/2]	74
6.56.2.9 TryFindObjectsOfType< T >() [2/2]	74
6.57 ValidationAttribute Class Reference	75
6.57.1 Detailed Description	75
6.58 ValidationAttributeEditor Class Reference	75
6.58.1 Detailed Description	75
<b>Index</b>	<b>77</b>



# Chapter 1

## CoreManager

Reasons to use this package:

- It allows avoiding reference serialization via Inspector.
- It removes the need for GOD objects.
- Provides high script flexibility and low cohesion.

### 1.1 Technical details

Package created on version 2020.3.6f1 but compatible with the versions starting from 2020.x.x

Any OS supported. Unity version 2020 and above. (Package requires C# 8.0)

### 1.2 Features

1. **Core Manager** - used for scene, subscription, and reference initialization.
2. **Cross Events** - replacement for the traditional event serialization and subscription.
3. **Reference Distributor** - reference container for data distribution in one scene.
4. **Cross Scene Data Handler** - data container(not references) which allows data distribution between scenes. Supports both classes and structures.
5. **Custom Validation Attributes** - allows validating serialized data. This plugin contains both predefined validation attributes and mechanisms to implement new validation attributes. Supports both class and field attributes.
6. **Custom Editor** - used for attribute validation and displaying errors in Inspector.
7. **Custom Logger** - use this logger if you want to show logs in debug build/editor, but not in release build.
8. **Custom Play Mode** entering - prevents Play Mode start if the current scene contains validation failure.
9. **Custom Build** start - prevents application build if scenes included in the build or prefabs with validation attributes contain validation failures.
10. **SaveSystem** - system to save/load JSON files.

11. **UIStateTools & UIManager** - base UI system for page-based UI.
12. **Extensions** - for Editor and base classes.
13. **Scene Loader** - allows to asynchronously load scene through an intermediate scene and allows to serialize SceneAssets through Inspector (use SceneLoaderAsset).
14. **Runtime console** - console with Unity logs for debug and/or release builds. Allows display console logs like in Unity Editor. Strips from release builds if other not predetermined.
15. **Scene Loader** - add callbacks on scene changing.

## 1.3 Improvements

1. **RequireInterfaceAttribute**
2. **Delegate** combination for subscribing (checkout samples to learn how to use)

## 1.4 Breaking changes

### 1.4.1 v1.1.2

1. Event interfaces accept arrays instead of **IEnumerable**.

### 1.4.2 v1.1.3

1. Event interfaces accept **params** arrays.

## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

<a href="#">CorePlugin</a>	11
<a href="#">CorePlugin.Attributes</a>	11
<a href="#">CorePlugin.Attributes.Editor</a>	11
<a href="#">CorePlugin.Attributes.Editor.Drawers</a>	11
<a href="#">CorePlugin.Attributes.Headers</a>	11
<a href="#">CorePlugin.Attributes.Validation</a>	12
<a href="#">CorePlugin.Attributes.Validation.Base</a>	12
<a href="#">CorePlugin.Console</a>	12
<a href="#">CorePlugin.Console.ConsoleElements</a>	13
<a href="#">CorePlugin.Core</a>	13
<a href="#">CorePlugin.Core.Interface</a>	13
<a href="#">CorePlugin.Cross</a>	13
<a href="#">CorePlugin.Cross.Events</a>	13
<a href="#">CorePlugin.Cross.Events.Interface</a>	14
<a href="#">CorePlugin.Cross.SceneData</a>	14
<a href="#">CorePlugin.Cross.SceneData.Interface</a>	14
<a href="#">CorePlugin.Dispatchers</a>	14
<a href="#">CorePlugin.Editor</a>	14
<a href="#">CorePlugin.Extensions</a>	15
<a href="#">CorePlugin.FileSystem</a>	15
<a href="#">CorePlugin.Logger</a>	15
<a href="#">CorePlugin.ReferenceDistribution</a>	15
<a href="#">CorePlugin.ReferenceDistribution.Interface</a>	15
<a href="#">CorePlugin.Samples</a>	16
<a href="#">CorePlugin.Samples.Scripts</a>	16
<a href="#">CorePlugin.Samples.Scripts.Demo</a>	16
<a href="#">CorePlugin.Samples.Scripts.EventTypes</a>	16
<a href="#">CorePlugin.Samples.Scripts.Managers</a>	16
<a href="#">CorePlugin.Samples.Scripts.Model</a>	16
<a href="#">CorePlugin.SceneManagement</a>	16
<a href="#">CorePlugin.Serializable</a>	16
<a href="#">CorePlugin.Serializable.Interface</a>	16
<a href="#">CorePlugin.Singletons</a>	17
<a href="#">CorePlugin.UISystem</a>	17
<a href="#">CorePlugin.UISystem.UI</a>	17



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

ConsoleIcons . . . . .	20
ConsoleInitializer . . . . .	21
ConsoleMessage . . . . .	21
ConsoleTextSettings . . . . .	23
CoreManager . . . . .	24
CountDisplay . . . . .	25
MinimizeCountDisplay . . . . .	40
ToggleCountDisplay . . . . .	64
DebugLogger . . . . .	27
EventInitializer . . . . .	27
EventTypes . . . . .	29
FileSystem . . . . .	30
ICore . . . . .	34
BaseCore . . . . .	19
UIManager . . . . .	65
IDistributingReference . . . . .	35
IEventHandler . . . . .	35
IEventSubscriber . . . . .	37
ISceneData . . . . .	38
ItemHider . . . . .	38
IUnique . . . . .	39
Unique . . . . .	68
LoadSceneOptions . . . . .	39
MinimizedConsole . . . . .	41
MovableInsideScreen . . . . .	42
Named< TName, TKey, TValue > . . . . .	42
Named< string, Object > . . . . .	42
PrefabHeaderAttribute . . . . .	44
ReferenceDistributor . . . . .	45
ReferencesHeaderAttribute . . . . .	48
RequireInterfaceAttribute . . . . .	48
RequireInterfaceDrawer . . . . .	49
RuntimeConsole . . . . .	50
SceneLoader . . . . .	53

SceneLoaderAsset . . . . .	54
SceneLoaderExtensions . . . . .	55
SceneLoaderSettings . . . . .	57
SceneLoaderSettingsEditor . . . . .	58
SceneLoaderSettingsValidator . . . . .	58
SettingsHeaderAttribute . . . . .	59
Singleton< T > . . . . .	59
Singleton< SceneDataHandler > . . . . .	59
SceneDataHandler . . . . .	51
StateHeaderAttribute . . . . .	60
StaticObjectSingleton< T > . . . . .	60
StaticObjectSingleton< MainThreadDispatcher > . . . . .	60
StaticObjectSingleton< StaticCoroutineDispatcher > . . . . .	60
SubPagesController . . . . .	60
SymbolDefiner . . . . .	63
UIPage . . . . .	66
UIStateTools . . . . .	67
UnityEditorExtension . . . . .	69
UnityExtensions . . . . .	70
ValidationAttribute . . . . .	75
ClassValidationAttribute . . . . .	20
OneAndOnlyAttribute . . . . .	43
FieldValidationAttribute . . . . .	30
HasComponentAttribute . . . . .	34
NotNullAttribute . . . . .	43
PrefabRequiredAttribute . . . . .	44
ValidationAttributeEditor . . . . .	75
CoreManagerEditor . . . . .	24



## Chapter 4

# Class Index

### 4.1 Class List

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

<a href="#">BaseCore</a>	
Base implementation of <a href="#">IManager</a> . . . . .	19
<a href="#">ClassValidationAttribute</a>	
Base attribute for class validation . . . . .	20
<a href="#">20</a>	
<a href="#">ConsoleInitializer</a>	
Initialize minimized and maximized console . . . . .	21
<a href="#">ConsoleMessage</a>	
Log message for <a href="#">CorePlugin.Console.RuntimeConsole</a> . . . . .	21
<a href="#">ConsoleTextSettings</a>	
Settings class for <a href="#">CorePlugin.Console.RuntimeConsole</a> . . . . .	23
<a href="#">24</a>	
<a href="#">24</a>	
<a href="#">CountDisplayer</a>	
Log count displayer for <a href="#">CorePlugin.Console.RuntimeConsole</a> . . . . .	25
<a href="#">DebugLogger</a>	
Custom logger solution for logs . . . . .	27

27

[EventTypes](#)

Declare your delegate type to convert method group to delegate . . . . . 29

[FieldValidationAttribute](#)[Base](#) attribute for field validation . . . . . 30

30

[HasComponentAttribute](#)

Attribute validating whether Object in field or all objects in the list have desired component . . . . . 34

[ICore](#)[Interface](#) for Manager declaration . . . . . 34[IDistributingReference](#)

Flag interface for MonoBehaviour classes what need to referenced from another class . . . . . 35

[IEventHandler](#)[Interface](#) for event handler . . . . . 35[IEventSubscriber](#)[Interface](#) for subscribers . . . . . 37

38

[ItemHider](#)[UI](#) element hider. Useful than needed to hide elements on [UI](#) change its' size . . . . . 38[IUnique](#)[Interface](#) for unique objects. For example, you can use it for objects that need to be saved in Json or objects with the same data, but with different identifiers . . . . . 39[LoadSceneOptions](#)

Loading Scene Options . . . . . 39

[MinimizeCountDisplayer](#)Log count displayer for [CorePlugin.Console.RuntimeConsole](#) . . . . . 40[MinimizedConsole](#)

Minimized console . . . . . 41

[MovableInsideScreen](#)Class for moving [UI](#) objects inside screen . . . . . 42[Named< TName, TKey, TValue >](#)

Replacement for dictionary in Unity Inspector . . . . . 42

<a href="#">NotNullAttribute</a>	Attribute validating whether field or element in the list equals null . . . . .	43
<a href="#">OneAndOnlyAttribute</a>	Attribute validating whether there is only one copy of this class in the scene . . . . .	43
<a href="#">PrefabHeaderAttribute</a>	Replacement for Header("Prefabs") . . . . .	44
<a href="#">PrefabRequiredAttribute</a>	Attribute validating whether the object or all items in list are prefabs . . . . .	44
<a href="#">ReferenceDistributor</a>	Class responsible for reference distribution inside one scene . . . . .	45
<a href="#">ReferencesHeaderAttribute</a>	Replacement for Header("References") . . . . .	48
<a href="#">RequireInterfaceAttribute</a>	Attribute that require implementation of the provided interface . . . . .	48
<a href="#">RequireInterfaceDrawer</a>	Drawer for the RequireInterface attribute . . . . .	49
<a href="#">RuntimeConsole</a>	Main class for <a href="#">RuntimeConsole</a> . . . . .	50
<a href="#">51</a>		
<a href="#">SceneLoader</a>	Scene Loader at run-time . . . . .	53
<a href="#">54</a>		
<a href="#">55</a>		
<a href="#">SceneLoaderSettings</a>	Scene Loader Settings . . . . .	57
<a href="#">validation</a>	<a href="#">58</a>	
<a href="#">58</a>		

<a href="#">SettingsHeaderAttribute</a>	
Replacement for Header("Settings") . . . . .	59
<a href="#">Singleton&lt; T &gt;</a>	
Base for all singletons. Strongly recommended to use singletons as little as possible . . . . .	59
<a href="#">StateHeaderAttribute</a>	
Replacement for Header("State") . . . . .	60
<a href="#">StaticObjectSingleton&lt; T &gt;</a>	
Base for static objects singletons. Strongly recommended to use singletons as little as possible . . . . .	60
<a href="#">SubPagesController</a>	
Class designated for <a href="#">UI</a> subpages . . . . .	60
63	
<a href="#">ToggleCountDisplay</a>	
Log toggle for <a href="#">CorePlugin.Console.RuntimeConsole</a> . . . . .	64
<a href="#">UIManager</a>	
Base implementation of <a href="#">UIManager</a> . . . . .	65
<a href="#">UIPage</a>	
<a href="#">UI</a> page implementation . . . . .	66
<a href="#">UIStateTools</a>	
<a href="#">UI</a> state tool for canvas groups . . . . .	67
<a href="#">Unique</a>	
Base class for classes that need to be saved in Json file . . . . .	68
<a href="#">UnityEditorExtension</a>	
<a href="#">Extensions</a> for Unity <a href="#">Editor</a> classes . . . . .	69
<a href="#">UnityExtensions</a>	
Extension class for default Unity classes . . . . .	70
<a href="#">ValidationAttribute</a>	
<a href="#">Base</a> attribute for validation . . . . .	75

## Chapter 5

# Namespace Documentation

### 5.1 CorePlugin Namespace Reference

### 5.2 CorePlugin.Attributes Namespace Reference

### 5.3 CorePlugin.Attributes.Editor Namespace Reference

#### Classes

- class [ValidationAttributeEditor](#)

*Custom editor for Unity.Object class. If you want to create your own custom editor inherit from this class.  
See also*

*ValidationAttribute, FieldValidationAttribute, ClassValidationAttribute*

### 5.4 CorePlugin.Attributes.Editor.Drawers Namespace Reference

#### Classes

- class [RequireInterfaceDrawer](#)

*Drawer for the RequireInterface attribute.*

### 5.5 CorePlugin.Attributes.Headers Namespace Reference

#### Classes

- class [PrefabHeaderAttribute](#)  
*Replacement for Header("Prefabs")*
- class [ReferencesHeaderAttribute](#)  
*Replacement for Header("References")*
- class [SettingsHeaderAttribute](#)  
*Replacement for Header("Settings")*
- class [StateHeaderAttribute](#)  
*Replacement for Header("State")*

## 5.6 CorePlugin.Attributes.Validation Namespace Reference

### Classes

- class [HasComponentAttribute](#)  
*Attribute validating whether Object in field or all objects in the list have desired component.*
- class [NotNullAttribute](#)  
*Attribute validating whether field or element in the list equals null.*
- class [OneAndOnlyAttribute](#)  
*Attribute validating whether there is only one copy of this class in the scene.*
- class [PrefabRequiredAttribute](#)  
*Attribute validating whether the object or all items in list are prefabs.*
- class [RequireInterfaceAttribute](#)  
*Attribute that require implementation of the provided interface.*

## 5.7 CorePlugin.Attributes.Validation.Base Namespace Reference

### Classes

- class [ClassValidationAttribute](#)  
*Base attribute for class validation.*
- class [FieldValidationAttribute](#)  
*Base attribute for field validation.*
- class [ValidationAttribute](#)  
*Base attribute for validation*

## 5.8 CorePlugin.Console Namespace Reference

### Classes

- class [ConsoleInitializer](#)  
*Initialize minimized and maximized console*
- class [MinimizedConsole](#)  
*Minimized console*
- class [RuntimeConsole](#)  
*Main class for [RuntimeConsole](#)*

## 5.9 CorePlugin.Console.ConsoleElements Namespace Reference

### Classes

- class [ConsoleIcons](#)  
*List of icons for*  
*See also*  
  
[CorePlugin.Console.RuntimeConsole](#)
- class [ConsoleMessage](#)  
*Log message for [CorePlugin.Console.RuntimeConsole](#)*
- class [ConsoleTextSettings](#)  
*Settings class for [CorePlugin.Console.RuntimeConsole](#)*
- class [CountDisplayer](#)  
*Log count displayer for [CorePlugin.Console.RuntimeConsole](#)*
- class [MinimizeCountDisplayer](#)  
*Log count displayer for [CorePlugin.Console.RuntimeConsole](#)*
- class [ToggleCountDisplayer](#)  
*Log toggle for [CorePlugin.Console.RuntimeConsole](#)*

## 5.10 CorePlugin.Core Namespace Reference

### Classes

- class [BaseCore](#)  
*Base implementation of [IManager](#).*
- class [CoreManager](#)  
*Manager for initialization of sub manager in the scene.*  
*See also*  
  
[CorePlugin.ReferenceDistribution.ReferenceDistributor](#), [CorePlugin.Core.Interface.ICore](#)
- class [EventInitializer](#)  
*Class responsible for reference initialization*  
*See also*  
  
[CorePlugin.Cross.Events.Interface.IEventHandler](#), [CorePlugin.Cross.Events.Interface.IEventSubscriber](#)

## 5.11 CorePlugin.Core.Interface Namespace Reference

### Classes

- interface [ICore](#)  
*Interface for Manager declaration.*

## 5.12 CorePlugin.Cross Namespace Reference

## 5.13 CorePlugin.Cross.Events Namespace Reference

### Classes

- class [EventTypes](#)  
*Declare your delegate type to convert method group to delegate.*

## 5.14 CorePlugin.Cross.Events.Interface Namespace Reference

### Classes

- interface [IEventHandler](#)  
*Interface for event handler.*
- interface [IEventSubscriber](#)  
*Interface for subscribers.*

## 5.15 CorePlugin.Cross.SceneData Namespace Reference

### Classes

- class [SceneDataHandler](#)  
*Singleton for passing data between scenes*  
*See also*  
[CorePlugin.Cross.SceneData.Interface.ISceneData](#)

## 5.16 CorePlugin.Cross.SceneData.Interface Namespace Reference

### Classes

- interface [ISceneData](#)  
*Flag interface for classes or structs to pass between scenes*  
*See also*  
[CorePlugin.Cross.SceneData.SceneDataHandler](#)

## 5.17 CorePlugin.Dispatchers Namespace Reference

## 5.18 CorePlugin.Editor Namespace Reference

### Classes

- class [CoreManagerEditor](#)  
*Custom [Editor](#) [CoreManager](#) class.*  
*See also*  
[CorePlugin.Core.CoreManager](#)
- class [SceneLoaderSettingsEditor](#)  
*[Editor](#) for*  
*See also*  
[CorePlugin.SceneManagement.SceneLoaderSettings](#)  
*validation*
- class [SceneLoaderSettingsValidator](#)  
*Validator class for*  
*See also*  
[CorePlugin.SceneManagement.SceneLoaderSettings](#)
- class [SymbolDefiner](#)  
*Class for "Scripting Define Symbols" defining from [CoreManager](#) Inspector.*  
*See also*  
[CorePlugin.Core.CoreManager](#)



## 5.19 CorePlugin.Extensions Namespace Reference

### Classes

- class [Named](#)  
*Replacement for dictionary in Unity Inspector*
- class [UIStateTools](#)  
*UI state tool for canvas groups*
- class [UnityEditorExtension](#)  
*Extensions for Unity Editor classes*
- class [UnityExtensions](#)  
*Extension class for default Unity classes*

## 5.20 CorePlugin.FileSystem Namespace Reference

### Classes

- class [FileSystem](#)  
*Class for saving Json file to disk.  
See also*  
*[CorePlugin.Serializable.Unique](#)*

## 5.21 CorePlugin.Logger Namespace Reference

### Classes

- class [DebugLogger](#)  
*Custom logger solution for logs.*

## 5.22 CorePlugin.ReferenceDistribution Namespace Reference

### Classes

- class [ReferenceDistributor](#)  
*Class responsible for reference distribution inside one scene.*

## 5.23 CorePlugin.ReferenceDistribution.Interface Namespace Reference

### Classes

- interface [IDistributingReference](#)  
*Flag interface for MonoBehaviour classes what need to be referenced from another class.*

## 5.24 CorePlugin.Samples Namespace Reference

## 5.25 CorePlugin.Samples.Scripts Namespace Reference

## 5.26 CorePlugin.Samples.Scripts.Demo Namespace Reference

## 5.27 CorePlugin.Samples.Scripts.EventTypes Namespace Reference

## 5.28 CorePlugin.Samples.Scripts.Managers Namespace Reference

## 5.29 CorePlugin.Samples.Scripts.Model Namespace Reference

## 5.30 CorePlugin.SceneManagement Namespace Reference

### Classes

- class [LoadSceneOptions](#)  
*Loading Scene Options*
- class [SceneLoader](#)  
*Scene Loader at run-time*
- class [SceneLoaderAsset](#)  
*Class represents SceneAsset for*  
*See also*  
[CorePlugin.SceneManagement.SceneLoader](#)
- class [SceneLoaderExtensions](#)  
*Extensions for*  
*See also*  
[CorePlugin.SceneManagement.SceneLoader](#)
- class [SceneLoaderSettings](#)  
*Scene Loader Settings*

## 5.31 CorePlugin.Serializable Namespace Reference

### Classes

- class [Unique](#)  
*Base class for classes that need to be saved in Json file.*

## 5.32 CorePlugin.Serializable.Interface Namespace Reference

### Classes

- interface [IUnique](#)  
*Interface for unique objects. For example, you can use it for objects that need to be saved in Json or objects with the same data, but with different identifiers*

## 5.33 CorePlugin.Singletons Namespace Reference

### Classes

- class [Singleton](#)  
*Base for all singletons. Strongly recommended to use singletons as little as possible.*
- class [StaticObjectSingleton](#)  
*Base for static objects singletons. Strongly recommended to use singletons as little as possible.*

## 5.34 CorePlugin.UISystem Namespace Reference

### Classes

- class [ItemHider](#)  
*[UI](#) element hider. Useful than needed to hide elements on [UI](#) change its' size*
- class [MovableInsideScreen](#)  
*Class for moving [UI](#) objects inside screen*
- class [UIManager](#)  
*Base implementation of [UIManager](#).*

## 5.35 CorePlugin.UISystem.UI Namespace Reference

### Classes

- class [SubPagesController](#)  
*Class designated for [UI](#) subpages.*
- class [UIPage](#)  
*[UI](#) page implementation*



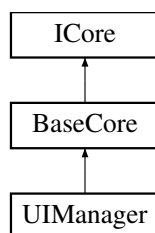
## Chapter 6

# Class Documentation

### 6.1 BaseCore Class Reference

Base implementation of IManager.

Inheritance diagram for BaseCore:



#### Public Member Functions

- virtual void [InitializeElements](#) ()  
*Initializing all managers elements.*

#### 6.1.1 Detailed Description

Base implementation of IManager.

#### 6.1.2 Member Function Documentation

### 6.1.2.1 InitializeElements()

```
virtual void InitializeElements ( ) [inline], [virtual]
```

Initializing all managers elements.

Implements [ICore](#).

Reimplemented in [UIManager](#).

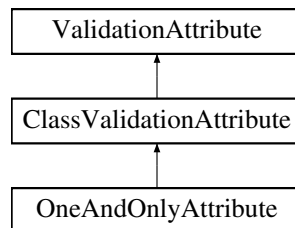
The documentation for this class was generated from the following file:

- BaseCore.cs

## 6.2 ClassValidationAttribute Class Reference

[Base](#) attribute for class validation.

Inheritance diagram for ClassValidationAttribute:



### 6.2.1 Detailed Description

[Base](#) attribute for class validation.

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

- ClassValidationAttribute.cs

## 6.3 ConsoleIcons Class Reference

List of icons for

See also

[CorePlugin.Console.RuntimeConsole](#)

Inherits [ScriptableObject](#).

### 6.3.1 Detailed Description

List of icons for

See also

[CorePlugin.Console.RuntimeConsole](#)

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

- ConsoleIcons.cs

## 6.4 ConsoleInitializer Class Reference

Initialize minimized and maximized console

Inherits MonoBehaviour.

### 6.4.1 Detailed Description

Initialize minimized and maximized console

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

- ConsoleInitializer.cs

## 6.5 ConsoleMessage Class Reference

Log message for [CorePlugin.Console.RuntimeConsole](#)

Inherits MonoBehaviour.

### Public Member Functions

- void [ClearHighlight](#) ()  
*Clear highlight marks*
- [ConsoleMessage HighlightText](#) (string text)  
*Sets highlight marks*
- [ConsoleMessage SetActive](#) (bool state)  
*Setting active message in console*
- [ConsoleMessage Initialize](#) (string logText, string stackTrace, LogType logType, [ConsoleTextSettings](#) settings)  
*Initializing console message*
- [ConsoleMessage SubscribeOnButtonClick](#) (Action< string > onClickAction)  
*Subscribes action to message button*

### 6.5.1 Detailed Description

Log message for [CorePlugin.Console.RuntimeConsole](#)

### 6.5.2 Member Function Documentation

#### 6.5.2.1 ClearHighlight()

```
void ClearHighlight ( ) [inline]
```

Clear highlight marks

#### 6.5.2.2 HighlightText()

```
ConsoleMessage HighlightText (
    string text ) [inline]
```

Sets highlight marks

#### 6.5.2.3 Initialize()

```
ConsoleMessage Initialize (
    string logText,
    string stackTrace,
    LogType logType,
    ConsoleTextSettings settings ) [inline]
```

Initializing console message

##### Parameters

<i>logText</i>	
<i>stackTrace</i>	
<i>logType</i>	
<i>settings</i>	

##### Returns



## Exceptions

<i>ArgumentOutOfRangeException</i>	
------------------------------------	--

#### 6.5.2.4 SetActive()

```
ConsoleMessage SetActive (
    bool state ) [inline]
```

Setting active message in console

## Parameters

<i>state</i>	
--------------	--

References `UIStateTools.ChangeGroupState()`.

#### 6.5.2.5 SubscribeOnButtonClick()

```
ConsoleMessage SubscribeOnButtonClick (
    Action< string > onClickAction ) [inline]
```

Subscribes action to message button

## Parameters

<i>onClickAction</i>	
----------------------	--

## Returns

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

- `ConsoleMessage.cs`

## 6.6 ConsoleTextSettings Class Reference

Settings class for [CorePlugin.Console.RuntimeConsole](#)

### 6.6.1 Detailed Description

Settings class for [CorePlugin.Console.RuntimeConsole](#)

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

- ConsoleTextSettings.cs

## 6.7 CoreManager Class Reference

Manager for initialization of sub manager in the scene.

See also

[CorePlugin.ReferenceDistribution.ReferenceDistributor](#), [CorePlugin.Core.Interface.ICore](#)

Inherits MonoBehaviour.

### 6.7.1 Detailed Description

Manager for initialization of sub manager in the scene.

See also

[CorePlugin.ReferenceDistribution.ReferenceDistributor](#), [CorePlugin.Core.Interface.ICore](#)

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

- CoreManager.cs

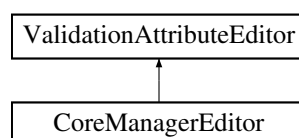
## 6.8 CoreManagerEditor Class Reference

Custom [Editor](#) CoreManager class.

See also

[CorePlugin.Core.CoreManager](#)

Inheritance diagram for CoreManagerEditor:



### 6.8.1 Detailed Description

Custom [Editor](#) CoreManager class.

See also

[CorePlugin.Core.CoreManager](#)

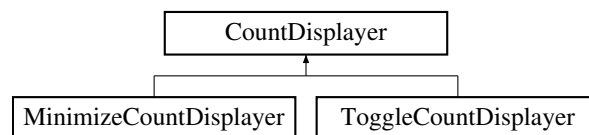
The documentation for this class was generated from the following file:

- CoreManagerEditor.cs

## 6.9 CountDisplayer Class Reference

Log count displayer for [CorePlugin.Console.RuntimeConsole](#)

Inheritance diagram for CountDisplayer:



### Public Member Functions

- virtual [CountDisplayer Initialize](#) ([ConsoleIcons](#) icons)  
*Initialize [CountDisplayer](#) with icons*
- abstract [CountDisplayer SetInteractionAction](#) (Action< LogType, bool > onInteractWithDisplayer)  
*Setting action when interaction with [CountDisplayer](#) happens*
- virtual void [OnLogCountChanged](#) (HashSet< LogType > types, int count)  
*Displaying new count*

### 6.9.1 Detailed Description

Log count displayer for [CorePlugin.Console.RuntimeConsole](#)

### 6.9.2 Member Function Documentation

#### 6.9.2.1 Initialize()

```
virtual CountDisplayer Initialize (
    ConsoleIcons icons ) [inline], [virtual]
```

Initialize [CountDisplayer](#) with icons

**Parameters**

<i>icons</i>	
--------------	--

**Returns**

Reimplemented in [ToggleCountDisplay](#), and [MinimizeCountDisplay](#).

Referenced by `ToggleCountDisplay.Initialize()`.

**6.9.2.2 OnLogCountChanged()**

```
virtual void OnLogCountChanged (
    HashSet< LogType > types,
    int count ) [inline], [virtual]
```

Displaying new count

**Parameters**

<i>types</i>	
<i>count</i>	

**6.9.2.3 SetInteractionAction()**

```
abstract CountDisplay SetInteractionAction (
    Action< LogType, bool > onInteractWithDisplay ) [pure virtual]
```

Setting action when interaction with [CountDisplay](#) happens

**Parameters**

<i>onInteractWithDisplay</i>	
------------------------------	--

**Returns**

Implemented in [ToggleCountDisplay](#), and [MinimizeCountDisplay](#).

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

- `CountDisplay.cs`

## 6.10 DebugLogger Class Reference

Custom logger solution for logs.

### 6.10.1 Detailed Description

Custom logger solution for logs.

Logs in this class are dependant on `DEBUG` and `ENABLE_RELEASE_LOGS`. If `ENABLE_RELEASE_LOGS` defined logs will displayed in Release Build. Otherwise only [Editor](#) and Developer Build will display logs. For defining preprocessor open CoreManager or write down in PlayerSettings in field "Scripting Define Symbols".

See also

[CorePlugin.Core.CoreManager](#)

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

- `DebugLogger.cs`

## 6.11 EventInitializer Class Reference

Class responsible for reference initialization

See also

[CorePlugin.Cross.Events.Interface.IEventHandler](#), [CorePlugin.Cross.Events.Interface.IEventSubscriber](#)

### Static Public Member Functions

- static void [InitializeSubscriptions](#) ()  
*Initialising cross subscriptions for all handlers in the scene.*
- static void [Subscribe](#) ([IEventSubscriber](#) subscriber)  
*Subscribing event subscriber after scene has Awoken to event handlers.*
- static void [Unsubscribe](#) ([IEventSubscriber](#) subscriber)  
*Unsubscribing event subscriber after scene has Awoken from event handlers.*
- static void [AddHandler](#) ([IEventHandler](#) handler, bool subscriptionsNeeded=true, bool invokeNeeded=false)  
*Adding new handler after scene has Awoken to list of event handlers.*
- static void [RemoveHandler](#) ([IEventHandler](#) handler)  
*Removing event handler after scene has Awoken from list*
- static void [InvokeBase](#) ()  
*Invoking event on handlers.*

### 6.11.1 Detailed Description

Class responsible for reference initialization

See also

[CorePlugin.Cross.Events.Interface.IEventHandler](#), [CorePlugin.Cross.Events.Interface.IEventSubscriber](#)

## 6.11.2 Member Function Documentation

### 6.11.2.1 AddHandler()

```
static void AddHandler (
    IEventHandler handler,
    bool subscriptionsNeeded = true,
    bool invokeNeeded = false ) [inline], [static]
```

Adding new handler after scene has Awoken to list of event handlers.

#### Parameters

<i>handler</i>	
<i>subscriptionsNeeded</i>	If false invokeNeeded will not be called
<i>invokeNeeded</i>	

References IEventHandler.InvokeEvents(), and IEventHandler.Subscribe().

### 6.11.2.2 InitializeSubscriptions()

```
static void InitializeSubscriptions ( ) [inline], [static]
```

Initialising cross subscriptions for all handlers in the scene.

### 6.11.2.3 InvokeBase()

```
static void InvokeBase ( ) [inline], [static]
```

Invoking event on handlers.

### 6.11.2.4 RemoveHandler()

```
static void RemoveHandler (
    IEventHandler handler ) [inline], [static]
```

Removing event handler after scene has Awoken from list

## Parameters

<i>handler</i>	
----------------	--

**6.11.2.5 Subscribe()**

```
static void Subscribe (
    IEventSubscriber subscriber ) [inline], [static]
```

Subscribing event subscriber after scene has Awoken to event handlers.

## Parameters

<i>subscriber</i>	
-------------------	--

**6.11.2.6 Unsubscribe()**

```
static void Unsubscribe (
    IEventSubscriber subscriber ) [inline], [static]
```

Unsubscribing event subscriber after scene has Awoken from event handlers.

## Parameters

<i>subscriber</i>	
-------------------	--

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

- EventInitializer.cs

**6.12 EventTypes Class Reference**

Declare your delegate type to convert method group to delegate.

Inherited by CustomEventTypes.

**6.12.1 Detailed Description**

Declare your delegate type to convert method group to delegate.

```
public delegate void DelegateClass(Type type1);
public delegate void DelegateClass2(Type type2, Type type3);
```

See also

[CorePlugin.Cross.Events.Interface.IEventHandler](#), [CorePlugin.Cross.Events.Interface.IEventSubscriber](#)

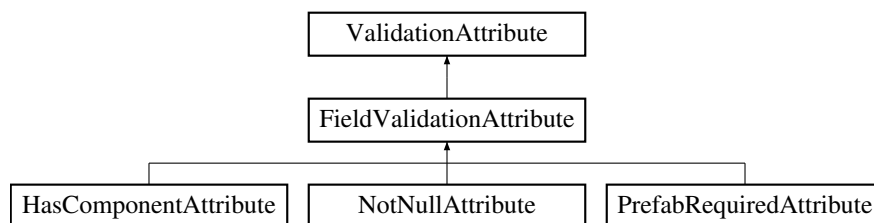
The documentation for this class was generated from the following file:

- EventTypes.cs

## 6.13 FieldValidationAttribute Class Reference

[Base](#) attribute for field validation.

Inheritance diagram for FieldValidationAttribute:



### 6.13.1 Detailed Description

[Base](#) attribute for field validation.

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

- FieldValidationAttribute.cs

## 6.14 FileSystem Class Reference

Class for saving Json file to disk.

See also

[CorePlugin.Serializable.Unique](#)

### Public Member Functions

- [FileSystem](#) ()  
*Default constructor.*
- [FileSystem](#) (string path)  
*Overloaded constructor which changes default path.*
- [FileSystem](#) (string path, string extension)  
*Overloaded constructor which changes default path and file extension.*
- void [Save](#)< T > (T data, Action< Exception > onError, Object context=null)  
*Saves class to the file with the name of class.*
- void [Save](#)< T > (T data, string fileName, Action< Exception > onError, Object context=null)  
*Saves your class to file.*
- void [Load](#)< T > (Action< T > onLoaded, Action< Exception > onError, Object context=null)  
*Loads file.*
- void [Load](#)< T > (string fileName, Action< T > onLoaded, Action< Exception > onError, Object context=null)  
*Loads file with different file name from passed class.*



### 6.14.1 Detailed Description

Class for saving Json file to disk.

See also

[CorePlugin.Serializable.Unique](#)

### 6.14.2 Constructor & Destructor Documentation

#### 6.14.2.1 FileSystem() [1/3]

```
FileSystem ( ) [inline]
```

Default constructor.

#### 6.14.2.2 FileSystem() [2/3]

```
FileSystem (
    string path ) [inline]
```

Overloaded constructor which changes default path.

Parameters

<i>path</i>	
-------------	--

#### 6.14.2.3 FileSystem() [3/3]

```
FileSystem (
    string path,
    string extension ) [inline]
```

Overloaded constructor which changes default path and file extension.

Parameters

<i>path</i>	
<i>extension</i>	

### 6.14.3 Member Function Documentation

#### 6.14.3.1 Load< T >() [1/2]

```
void Load< T > (
    Action< T > onLoaded,
    Action< Exception > onError,
    Object context = null ) [inline]
```

Loads file.

##### Parameters

<i>onLoaded</i>	
<i>onError</i>	
<i>context</i>	Required if errors should be shown on Object

##### Template Parameters

<i>T</i>	
----------	--

##### Type Constraints

***T* : Unique**

#### 6.14.3.2 Load< T >() [2/2]

```
void Load< T > (
    string fileName,
    Action< T > onLoaded,
    Action< Exception > onError,
    Object context = null ) [inline]
```

Loads file with different file name from passed class.

##### Parameters

<i>fileName</i>	
<i>onLoaded</i>	
<i>onError</i>	
<i>context</i>	Required if errors should be shown on Object

##### Template Parameters

<i>T</i>	
----------	--

## Type Constraints

***T : Unique*****6.14.3.3 Save< T >() [1/2]**

```
void Save< T > (
    T data,
    Action< Exception > onError,
    Object context = null ) [inline]
```

Saves class to the file with the name of class.

## Parameters

<i>data</i>	
<i>onError</i>	
<i>context</i>	Required if errors should be shown on Object

## Template Parameters

<i>T</i>	
----------	--

## Type Constraints

***T : Unique*****6.14.3.4 Save< T >() [2/2]**

```
void Save< T > (
    T data,
    string fileName,
    Action< Exception > onError,
    Object context = null ) [inline]
```

Saves your class to file.

## Parameters

<i>data</i>	
<i>fileName</i>	
<i>onError</i>	
<i>context</i>	Required if errors should be shown on Object

## Template Parameters

<i>T</i>	
----------	--

## Type Constraints

***T : Unique***

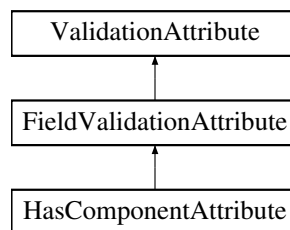
The documentation for this class was generated from the following file:

- FileSystem.cs

## 6.15 HasComponentAttribute Class Reference

Attribute validating whether Object in field or all objects in the list have desired component.

Inheritance diagram for HasComponentAttribute:



### 6.15.1 Detailed Description

Attribute validating whether Object in field or all objects in the list have desired component.

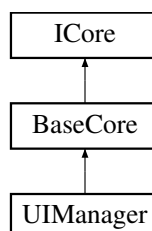
The documentation for this class was generated from the following file:

- HasComponentAttribute.cs

## 6.16 ICore Interface Reference

[Interface](#) for Manager declaration.

Inheritance diagram for ICore:



## Public Member Functions

- void [InitializeElements](#) ()  
*Initializing all managers elements.*

### 6.16.1 Detailed Description

[Interface](#) for Manager declaration.

### 6.16.2 Member Function Documentation

#### 6.16.2.1 InitializeElements()

```
void InitializeElements ( )
```

Initializing all managers elements.

Implemented in [UIManager](#), and [BaseCore](#).

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

- ICore.cs

## 6.17 IDistributingReference Interface Reference

Flag interface for MonoBehaviour classes what need to referenced from another class.

### 6.17.1 Detailed Description

Flag interface for MonoBehaviour classes what need to referenced from another class.

Strongly recommended to use [CorePlugin.Cross.Events](#) and [CorePlugin.Cross.Events.Interface](#) instead of direct reference serialization.

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

- IDistributingReference.cs

## 6.18 IEventHandler Interface Reference

[Interface](#) for event handler.

Inherited by [CrossSceneDataSenderDemo](#).

## Public Member Functions

- void [InvokeEvents](#) ()  
*Invoking events that need to be invoked on scene initialization.*
- void [Subscribe](#) (params Delegate[] subscribers)  
*Subscribing delegates to event*
- void [Unsubscribe](#) (params Delegate[] unsubscribers)  
*Unsubscribing delegates to event*

### 6.18.1 Detailed Description

[Interface](#) for event handler.

```
public void Subscribe(params Delegate[] subscriber)
{
    foreach (var action in subscriber.OfType<EventTypes.DelegateClass>())
    {
        MyEvent += action;
    }
}
```

See also

[CorePlugin.Cross.Events.EventTypes](#)

### 6.18.2 Member Function Documentation

#### 6.18.2.1 InvokeEvents()

```
void InvokeEvents ( )
```

Invoking events that need to be invoked on scene initialization.

Referenced by `EventInitializer.AddHandler()`.

#### 6.18.2.2 Subscribe()

```
void Subscribe (
    params Delegate[] subscribers )
```

Subscribing delegates to event

## Parameters

<i>subscribers</i>	
--------------------	--

Referenced by EventInitializer.AddHandler().

### 6.18.2.3 Unsubscribe()

```
void Unsubscribe (
    params Delegate[] unsubscribers )
```

Unsubscribing delegates to event

## Parameters

<i>unsubscribers</i>	
----------------------	--

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

- IEventHandler.cs

## 6.19 IEventSubscriber Interface Reference

[Interface](#) for subscribers.

Inherited by SceneSwitcher.

### Public Member Functions

- Delegate[] [GetSubscribers](#) ()  
*Returns IEnumerable with all methods which need to be subscribed.*

### 6.19.1 Detailed Description

[Interface](#) for subscribers.

```
public IEnumerable<Delegate> GetSubscribers()
{
    var list = new Delegate[] { (CrossEventTypes.DelegateClass) MyMethod, (Cross↵
EventTypes.DelegateClass2) MyMethod2};
    return list;
}
```

See also

[CorePlugin.Cross.Events.EventTypes](#)

## 6.19.2 Member Function Documentation

### 6.19.2.1 GetSubscribers()

```
Delegate [] GetSubscribers ( )
```

Returns IEnumerable with all methods which need to be subscribed.

#### Returns

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

- IEventSubscriber.cs

## 6.20 ISceneData Interface Reference

Flag interface for classes or structs to pass between scenes

#### See also

[CorePlugin.Cross.SceneData.SceneDataHandler](#)

Inherited by DataTransfer.

### 6.20.1 Detailed Description

Flag interface for classes or structs to pass between scenes

#### See also

[CorePlugin.Cross.SceneData.SceneDataHandler](#)

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

- ISceneData.cs

## 6.21 ItemHider Class Reference

UI element hider. Useful than needed to hide elements on UI change its' size

Inherits MonoBehaviour.



### 6.21.1 Detailed Description

UI element hider. Useful than needed to hide elements on UI change its' size

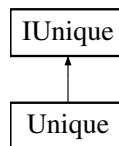
The documentation for this class was generated from the following file:

- ItemHider.cs

## 6.22 IUnique Interface Reference

Interface for unique objects. For example, you can use it for objects that need to be saved in Json or objects with the same data, but with different identifiers

Inheritance diagram for IUnique:



### 6.22.1 Detailed Description

Interface for unique objects. For example, you can use it for objects that need to be saved in Json or objects with the same data, but with different identifiers

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

- IUnique.cs

## 6.23 LoadSceneOptions Class Reference

Loading Scene Options

### 6.23.1 Detailed Description

Loading Scene Options

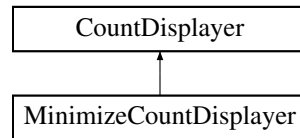
The documentation for this class was generated from the following file:

- LoadSceneOptions.cs

## 6.24 MinimizeCountDisplayer Class Reference

Log count displayer for [CorePlugin.Console.RuntimeConsole](#)

Inheritance diagram for MinimizeCountDisplayer:



### Public Member Functions

- override [CountDisplayer Initialize](#) ([ConsoleIcons](#) icons)  
*Initialize [CountDisplayer](#) with icons*
- override [CountDisplayer SetInteractionAction](#) (Action< LogType, bool > onInteractWithDisplayer)  
*Setting action when interaction with [CountDisplayer](#) happens*

### 6.24.1 Detailed Description

Log count displayer for [CorePlugin.Console.RuntimeConsole](#)

### 6.24.2 Member Function Documentation

#### 6.24.2.1 Initialize()

```
override CountDisplayer Initialize (  
    ConsoleIcons icons ) [inline], [virtual]
```

Initialize [CountDisplayer](#) with icons

#### Parameters

<i>icons</i>	
--------------	--

#### Returns

Reimplemented from [CountDisplayer](#).

### 6.24.2.2 SetInteractionAction()

```
override CountDisplayer SetInteractionAction (
    Action< LogType, bool > onInteractWithDisplayer ) [inline], [virtual]
```

Setting action when interaction with [CountDisplayer](#) happens

#### Parameters

<i>onInteractWithDisplayer</i>	
--------------------------------	--

#### Returns

Implements [CountDisplayer](#).

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

- MinimizeCountDisplayer.cs

## 6.25 MinimizedConsole Class Reference

Minimized console

Inherits [MonoBehaviour](#), [IPointerClickHandler](#), and [IBeginDragHandler](#).

### Public Member Functions

- [MinimizedConsole Initialize](#) (Action onMaximized, [ConsoleIcons](#) icons)  
*Initializing [MinimizedConsole](#)*
- void [SetActive](#) (bool state)  
*Hides or Show console*

### 6.25.1 Detailed Description

Minimized console

Works together with

See also

[CorePlugin.Console.RuntimeConsole](#)

### 6.25.2 Member Function Documentation

#### 6.25.2.1 Initialize()

```
MinimizedConsole Initialize (
    Action onMaximized,
    ConsoleIcons icons ) [inline]
```

Initializing [MinimizedConsole](#)

**Parameters**

<i>onMaximized</i>	
<i>icons</i>	

**Returns****6.25.2.2 SetActive()**

```
void SetActive (
    bool state ) [inline]
```

Hides or Show console

**Parameters**

<i>state</i>	
--------------	--

References `UIStateTools.ChangeGroupState()`.

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

- `MinimizedConsole.cs`

**6.26 MovableInsideScreen Class Reference**

Class for moving [UI](#) objects inside screen

Inherits `MonoBehaviour`, `IDragHandler`, `IEndDragHandler`, and `IBeginDragHandler`.

**6.26.1 Detailed Description**

Class for moving [UI](#) objects inside screen

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

- `MovableInsideScreen.cs`

**6.27 Named< TName, TKey, TValue > Class Template Reference**

Replacement for dictionary in Unity Inspector

**6.27.1 Detailed Description**

Replacement for dictionary in Unity Inspector

More complex list for dictionary

## Template Parameters

<i>TKey</i>	
<i>TValue</i>	

## Template Parameters

<i>TName</i>	
<i>TKey</i>	
<i>TValue</i>	

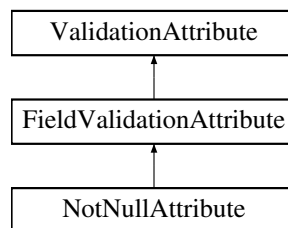
The documentation for this class was generated from the following file:

- Named.cs

## 6.28 NotNullAttribute Class Reference

Attribute validating whether field or element in the list equals null.

Inheritance diagram for NotNullAttribute:



### 6.28.1 Detailed Description

Attribute validating whether field or element in the list equals null.

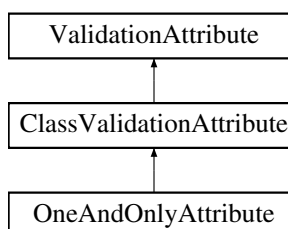
The documentation for this class was generated from the following file:

- NotNullAttribute.cs

## 6.29 OneAndOnlyAttribute Class Reference

Attribute validating whether there is only one copy of this class in the scene.

Inheritance diagram for OneAndOnlyAttribute:



### 6.29.1 Detailed Description

Attribute validating whether there is only one copy of this class in the scene.

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

- OneAndOnlyAttribute.cs

## 6.30 PrefabHeaderAttribute Class Reference

Replacement for Header("Prefabs")

Inherits HeaderAttribute.

### 6.30.1 Detailed Description

Replacement for Header("Prefabs")

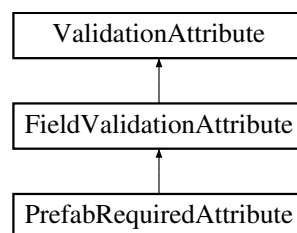
The documentation for this class was generated from the following file:

- PrefabHeaderAttribute.cs

## 6.31 PrefabRequiredAttribute Class Reference

Attribute validating whether the object or all items in list are prefabs.

Inheritance diagram for PrefabRequiredAttribute:



### 6.31.1 Detailed Description

Attribute validating whether the object or all items in list are prefabs.

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

- PrefabRequiredAttribute.cs

## 6.32 ReferenceDistributor Class Reference

Class responsible for reference distribution inside one scene.

Inherits MonoBehaviour.

### Public Member Functions

- void [Initialize](#) ()  
*Initializing distribution references*

### Static Public Member Functions

- static T [GetReference](#)< T > ([CallerMemberName] string callerName="")  
*Getting reference by type from list*
- static T [GetInterfaceReference](#)< T > ([CallerMemberName] string callerName="")  
*Getting reference by type from list*
- static IEnumerable< T > [GetInterfaceReferences](#)< T > ([CallerMemberName] string callerName="")  
*Getting reference by type from list*
- static IEnumerable< T > [GetReferences](#)< T > ([CallerMemberName] string callerName="")  
*Getting references by type from list*
- static bool [AskReference](#)< T > (ref T reference, [CallerMemberName] string callerName="")  
*Finding reference if passed parameter is null. Use this if you need reference not in Start() and/or reference should be received in some event*

### 6.32.1 Detailed Description

Class responsible for reference distribution inside one scene.

Strongly recommended to use [CorePlugin.Cross.Events](#) and [CorePlugin.Cross.Events.Interface](#) instead of direct reference serialization.

### 6.32.2 Member Function Documentation

#### 6.32.2.1 AskReference< T >()

```
static bool AskReference< T > (
    ref T reference,
    [CallerMemberName] string callerName = "" ) [inline], [static]
```

Finding reference if passed parameter is null. Use this if you need reference not in Start() and/or reference should be received in some event

## Parameters

<i>reference</i>	
------------------	--

## Template Parameters

<i>T</i>	
----------	--

## Returns

## Type Constraints

***T : MonoBehaviour***  
***T : IDistributingReference***

References ReferenceDistributor.GetReference< T >().

**6.32.2.2 GetInterfaceReference< T >()**

```
static T GetInterfaceReference< T > (
    [CallerMemberName] string callerName = "" ) [inline], [static]
```

Getting reference by type from list

## Template Parameters

<i>T</i>	
----------	--

## Returns

## Type Constraints

***T : IDistributingReference***

**6.32.2.3 GetInterfaceReferences< T >()**

```
static IEnumerable<T> GetInterfaceReferences< T > (
    [CallerMemberName] string callerName = "" ) [inline], [static]
```

Getting reference by type from list



## Template Parameters

<i>T</i>	
----------	--

## Returns

## Type Constraints

***T : IDistributingReference***

**6.32.2.4 GetReference< T >()**

```
static T GetReference< T > (  
    [CallerMemberName] string callerName = "" ) [inline], [static]
```

Getting reference by type from list

## Template Parameters

<i>T</i>	
----------	--

## Returns

## Type Constraints

***T : MonoBehaviour***  
***T : IDistributingReference***

Referenced by ReferenceDistributor.AskReference< T >().

**6.32.2.5 GetReferences< T >()**

```
static IEnumerable<T> GetReferences< T > (  
    [CallerMemberName] string callerName = "" ) [inline], [static]
```

Getting references by type from list

## Template Parameters

<i>T</i>	
----------	--

Returns

Type Constraints

***T : MonoBehaviour***

***T : IDistributingReference***

#### 6.32.2.6 Initialize()

```
void Initialize ( ) [inline]
```

Initializing distribution references

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

- ReferenceDistributor.cs

### 6.33 ReferencesHeaderAttribute Class Reference

Replacement for Header("References")

Inherits HeaderAttribute.

#### 6.33.1 Detailed Description

Replacement for Header("References")

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

- ReferencesHeaderAttribute.cs

### 6.34 RequireInterfaceAttribute Class Reference

Attribute that require implementation of the provided interface.

Inherits PropertyAttribute.

#### Public Member Functions

- [RequireInterfaceAttribute](#) (Type type)

*Requiring implementation of the T:CorePlugin.Attributes.Validation.RequireInterfaceAttribute interface.*

### 6.34.1 Detailed Description

Attribute that require implementation of the provided interface.

### 6.34.2 Constructor & Destructor Documentation

#### 6.34.2.1 RequireInterfaceAttribute()

```
RequireInterfaceAttribute (
    Type type ) [inline]
```

Requiring implementation of the T:CorePlugin.Attributes.Validation.RequireInterfaceAttribute interface.

Parameters

<i>type</i>	Interface type.
-------------	-----------------

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

- RequireInterfaceAttribute.cs

## 6.35 RequireInterfaceDrawer Class Reference

Drawer for the RequireInterface attribute.

Inherits PropertyDrawer.

### Public Member Functions

- override void [OnGUI](#) (Rect position, SerializedProperty property, GUIContent label)  
*Overrides GUI drawing for the attribute.*

### 6.35.1 Detailed Description

Drawer for the RequireInterface attribute.

### 6.35.2 Member Function Documentation

#### 6.35.2.1 OnGUI()

```
override void OnGUI (
    Rect position,
    SerializedProperty property,
    GUIContent label ) [inline]
```

Overrides GUI drawing for the attribute.

## Parameters

<i>position</i>	Position.
<i>property</i>	Property.
<i>label</i>	Label.

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

- `RequireInterfaceDrawer.cs`

## 6.36 RuntimeConsole Class Reference

Main class for [RuntimeConsole](#)

Inherits `MonoBehaviour`.

### Public Member Functions

- [RuntimeConsole Initialize](#) (Action onMinimized, [ConsoleIcons](#) icons)  
*Initializing [RuntimeConsole](#)*
- void [SetActive](#) (bool state)  
*Hides or Show console*

#### 6.36.1 Detailed Description

Main class for [RuntimeConsole](#)

#### 6.36.2 Member Function Documentation

##### 6.36.2.1 Initialize()

```
RuntimeConsole Initialize (
    Action onMinimized,
    ConsoleIcons icons ) [inline]
```

Initializing [RuntimeConsole](#)

## Parameters

<i>onMinimized</i>	Action what will executed on console minimized
<i>icons</i>	

Returns

### 6.36.2.2 SetActive()

```
void SetActive (
    bool state ) [inline]
```

Hides or Show console

Parameters

<i>state</i>	
--------------	--

References `UIStateTools.ChangeGroupState()`.

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

- `RuntimeConsole.cs`

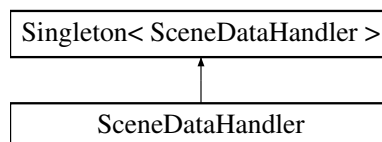
## 6.37 SceneDataHandler Class Reference

Singleton for passing data between scenes

See also

[CorePlugin.Cross.SceneData.Interface.ISceneData](#)

Inheritance diagram for SceneDataHandler:



### Public Member Functions

- void [AddData< T >](#) (T data)  
*Adding data to dictionary by passed Type*
- bool [GetData< T >](#) (out T data)  
*Getting data from dictionary by passed Type*
- void [RemoveData< T >](#) ()  
*Removing data from dictionary by passed Type*

### 6.37.1 Detailed Description

Singleton for passing data between scenes

See also

[CorePlugin.Cross.SceneData.Interface.ISceneData](#)

### 6.37.2 Member Function Documentation

#### 6.37.2.1 AddData< T >()

```
void AddData< T > (  
    T data ) [inline]
```

Adding data to dictionary by passed Type

Parameters

<i>data</i>	
-------------	--

Template Parameters

<i>T</i>	
----------	--

Type Constraints

***T : ISceneData***  
***T : new()***

#### 6.37.2.2 GetData< T >()

```
bool GetData< T > (  
    out T data ) [inline]
```

Getting data from dictionary by passed Type

Parameters

<i>data</i>	
-------------	--

## Template Parameters

<i>T</i>	
----------	--

## Type Constraints

***T : ISceneData******T : new()*****6.37.2.3 RemoveData< T >()**

```
void RemoveData< T > ( ) [inline]
```

Removing data from dictionary by passed Type

## Template Parameters

<i>T</i>	
----------	--

## Type Constraints

***T : ISceneData******T : new()***

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

- SceneDataHandler.cs

## 6.38 SceneLoader Class Reference

Scene Loader at run-time

### Static Public Member Functions

- static void [LoadSceneAsync](#) ([SceneLoaderAsset](#) asset, [LoadSceneOptions](#) loadSceneOptions)  
*Loads async [SceneLoaderAsset](#) with [LoadSceneOptions](#)*
- static void [LoadSceneAsync](#) ([SceneLoaderAsset](#) asset)  
*Loads async [SceneLoaderAsset](#) with default options*

### 6.38.1 Detailed Description

Scene Loader at run-time

## 6.38.2 Member Function Documentation

### 6.38.2.1 LoadSceneAsync() [1/2]

```
static void LoadSceneAsync (
    SceneLoaderAsset asset ) [inline], [static]
```

Loads async [SceneLoaderAsset](#) with default options

#### Parameters

<i>asset</i>	
--------------	--

References [SceneLoader.LoadSceneAsync\(\)](#).

### 6.38.2.2 LoadSceneAsync() [2/2]

```
static void LoadSceneAsync (
    SceneLoaderAsset asset,
    LoadSceneOptions loadSceneOptions ) [inline], [static]
```

Loads async [SceneLoaderAsset](#) with [LoadSceneOptions](#)

#### Parameters

<i>asset</i>	
<i>loadSceneOptions</i>	

Referenced by [SceneLoader.LoadSceneAsync\(\)](#).

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

- [SceneLoader.cs](#)

## 6.39 SceneLoaderAsset Class Reference

Class represents SceneAsset for

#### See also

[CorePlugin.SceneManagement.SceneLoader](#)



### 6.39.1 Detailed Description

Class represents SceneAsset for

See also

[CorePlugin.SceneManagement.SceneLoader](#)

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

- SceneLoaderAsset.cs

## 6.40 SceneLoaderExtensions Class Reference

[Extensions](#) for

See also

[CorePlugin.SceneManagement.SceneLoader](#)

### Static Public Member Functions

- static IEnumerator [SceneUnloadOperation](#) (this Scene scene, UnloadSceneOptions mode, Action< AsyncOperation > onSceneReadyToSwitch, Action< float > onProgressChanged=null)  
*Unloads current Scene*
- static bool [Until](#) (Action< float > onProgressChanged, AsyncOperation sceneOperation)  
*Awaits Until sceneOperation ready to switch*
- static IEnumerator [SceneLoadOperation](#) (this [SceneLoaderAsset](#) sceneAsset, LoadSceneMode mode, Action< AsyncOperation > onSceneReadyToSwitch, Action< float > onProgressChanged=null)  
*Loads [SceneLoaderAsset](#)*
- static IEnumerator [SceneUnloadOperation](#) (this string name, UnloadSceneOptions mode, Action< AsyncOperation > onSceneReadyToSwitch, Action< float > onProgressChanged=null)  
*Unloads current Scene*

### 6.40.1 Detailed Description

[Extensions](#) for

See also

[CorePlugin.SceneManagement.SceneLoader](#)

### 6.40.2 Member Function Documentation

#### 6.40.2.1 SceneLoadOperation()

```
static IEnumerator SceneLoadOperation (
    this SceneLoaderAsset sceneAsset,
    LoadSceneMode mode,
    Action< AsyncOperation > onSceneReadyToSwitch,
    Action< float > onProgressChanged = null ) [inline], [static]
```

Loads [SceneLoaderAsset](#)

**Parameters**

<i>sceneAsset</i>	
<i>mode</i>	
<i>onSceneReadyToSwitch</i>	
<i>onProgressChanged</i>	

**Returns**

References SceneLoaderExtensions.Until().

**6.40.2.2 SceneUnloadOperation() [1/2]**

```
static IEnumerator SceneUnloadOperation (
    this Scene scene,
    UnloadSceneOptions mode,
    Action< AsyncOperation > onSceneReadyToSwitch,
    Action< float > onProgressChanged = null ) [inline], [static]
```

Unloads current Scene

**Parameters**

<i>scene</i>	
<i>mode</i>	
<i>onSceneReadyToSwitch</i>	
<i>onProgressChanged</i>	

**Returns**

References SceneLoaderExtensions.Until().

**6.40.2.3 SceneUnloadOperation() [2/2]**

```
static IEnumerator SceneUnloadOperation (
    this string name,
    UnloadSceneOptions mode,
    Action< AsyncOperation > onSceneReadyToSwitch,
    Action< float > onProgressChanged = null ) [inline], [static]
```

Unloads current Scene

## Parameters

<i>name</i>	
<i>mode</i>	
<i>onSceneReadyToSwitch</i>	
<i>onProgressChanged</i>	

## Returns

References SceneLoaderExtensions.Until().

### 6.40.2.4 Until()

```
static bool Until (
    Action< float > onProgressChanged,
    AsyncOperation sceneOperation ) [inline], [static]
```

Awaits Until sceneOperation ready to switch

## Parameters

<i>onProgressChanged</i>	
<i>sceneOperation</i>	

## Returns

Referenced by SceneLoaderExtensions.SceneLoadOperation(), and SceneLoaderExtensions.SceneUnloadOperation().

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

- SceneLoaderExtensions.cs

## 6.41 SceneLoaderSettings Class Reference

Scene Loader Settings

Inherits ScriptableObject.

### 6.41.1 Detailed Description

Scene Loader Settings

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

- SceneLoaderSettings.cs

## 6.42 SceneLoaderSettingsEditor Class Reference

[Editor](#) for

See also

[CorePlugin.SceneManagement.SceneLoaderSettings](#)

validation

Inherits Editor.

### 6.42.1 Detailed Description

[Editor](#) for

See also

[CorePlugin.SceneManagement.SceneLoaderSettings](#)

validation

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

- SceneLoaderSettingsEditor.cs

## 6.43 SceneLoaderSettingsValidator Class Reference

Validator class for

See also

[CorePlugin.SceneManagement.SceneLoaderSettings](#)

### 6.43.1 Detailed Description

Validator class for

See also

[CorePlugin.SceneManagement.SceneLoaderSettings](#)

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

- SceneLoaderSettingsValidator.cs

## 6.44 SettingsHeaderAttribute Class Reference

Replacement for Header("Settings")

Inherits HeaderAttribute.

### 6.44.1 Detailed Description

Replacement for Header("Settings")

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

- SettingsHeaderAttribute.cs

## 6.45 Singleton< T > Class Template Reference

Base for all singletons. Strongly recommended to use singletons as little as possible.

Inherits MonoBehaviour.

### 6.45.1 Detailed Description

Base for all singletons. Strongly recommended to use singletons as little as possible.

Template Parameters

<i>T</i>	
----------	--

Type Constraints

***T* : *MonoBehaviour***

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

- Singleton.cs

## 6.46 StateHeaderAttribute Class Reference

Replacement for Header("State")

Inherits HeaderAttribute.

### 6.46.1 Detailed Description

Replacement for Header("State")

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

- StateHeaderAttribute.cs

## 6.47 StaticObjectSingleton< T > Class Template Reference

Base for static objects singletons. Strongly recommended to use singletons as little as possible.

Inherits MonoBehaviour.

### 6.47.1 Detailed Description

Base for static objects singletons. Strongly recommended to use singletons as little as possible.

Template Parameters

<i>T</i>	
----------	--

Type Constraints

***T* : *MonoBehaviour***

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

- StaticObjectSingleton.cs

## 6.48 SubPagesController Class Reference

Class designated for [UI](#) subpages.

Inherits MonoBehaviour.

## Public Member Functions

- void [OpenPage](#) ([UIPage](#) page)  
*Showing canvas group sent thought parameter and disabling all others.*
- void [AddPage](#) ([UIPage](#) page, out Action openPage)  
*Adding new [UIPage](#) to this subpages controller*
- void [AddPage](#) ([UIPage](#) page, out UnityAction openPage)  
*Adding new [UIPage](#) to this subpages controller*
- void [DelayedOpenPage](#) ([UIPage](#) page)  
*Enables page interaction after delay*
- void [DelayedOpenPage](#) ([UIPage](#) page, float delay)  
*Enables page interaction after delay*
- void [HideAllTables](#) ()  
*Hides all pages in this subpage controller*

### 6.48.1 Detailed Description

Class designated for [UI](#) subpages.

### 6.48.2 Member Function Documentation

#### 6.48.2.1 [AddPage\(\)](#) [1/2]

```
void AddPage (
    UIPage page,
    out Action openPage ) [inline]
```

Adding new [UIPage](#) to this subpages controller

##### Parameters

<i>page</i>	
<i>openPage</i>	

References [SubPagesController.OpenPage\(\)](#).

Referenced by [UIManager.InitializeElements\(\)](#).

#### 6.48.2.2 [AddPage\(\)](#) [2/2]

```
void AddPage (
    UIPage page,
    out UnityAction openPage ) [inline]
```

Adding new [UIPage](#) to this subpages controller

**Parameters**

<i>page</i>	
<i>openPage</i>	

References SubPagesController.OpenPage().

**6.48.2.3 DelayedOpenPage() [1/2]**

```
void DelayedOpenPage (
    UIWindow page ) [inline]
```

Enables page interaction after delay

**Parameters**

<i>page</i>	
-------------	--

**6.48.2.4 DelayedOpenPage() [2/2]**

```
void DelayedOpenPage (
    UIWindow page,
    float delay ) [inline]
```

Enables page interaction after delay

**Parameters**

<i>page</i>	
<i>delay</i>	

References UIStateTools.ChangeGroupState(), and SubPagesController.HideAllTables().

**6.48.2.5 HideAllTables()**

```
void HideAllTables ( ) [inline]
```

Hides all pages in this subpage controller

References UIStateTools.ChangeGroupState().

Referenced by SubPagesController.DelayedOpenPage(), and SubPagesController.OpenPage().



### 6.48.2.6 OpenPage()

```
void OpenPage (
    UIPage page ) [inline]
```

Showing canvas group sent thought parameter and disabling all others.

#### Parameters

<i>page</i>	
-------------	--

References [UIStateTools.ChangeGroupState\(\)](#), and [SubPagesController.HideAllTables\(\)](#).

Referenced by [SubPagesController.AddPage\(\)](#).

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

- [SubPagesController.cs](#)

## 6.49 SymbolDefiner Class Reference

Class for "Scripting Define Symbols" defining from CoreManager Inspector.

See also

[CorePlugin.Core.CoreManager](#)

### Public Member Functions

- void [ShowSymbolsButtons](#) ()  
*Shows buttons in Inspector.*
- void [OnEnable](#) ()  
*Check for whether the are already defined symbols OnEnable*

### 6.49.1 Detailed Description

Class for "Scripting Define Symbols" defining from CoreManager Inspector.

See also

[CorePlugin.Core.CoreManager](#)

### 6.49.2 Member Function Documentation

### 6.49.2.1 OnEnable()

```
void OnEnable ( ) [inline]
```

Check for whether the are already defined symbols OnEnable

### 6.49.2.2 ShowSymbolsButtons()

```
void ShowSymbolsButtons ( ) [inline]
```

Shows buttons in Inspector.

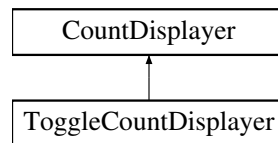
The documentation for this class was generated from the following file:

- SymbolDefiner.cs

## 6.50 ToggleCountDisplayer Class Reference

Log toggle for [CorePlugin.Console.RuntimeConsole](#)

Inheritance diagram for ToggleCountDisplayer:



### Public Member Functions

- override [CountDisplayer Initialize](#) ([ConsoleIcons](#) icons)  
*Initialize [CountDisplayer](#) with icons*
- override [CountDisplayer SetInteractionAction](#) (Action< LogType, bool > onInteractWithDisplayer)  
*Setting action when interaction with [CountDisplayer](#) happens*

### 6.50.1 Detailed Description

Log toggle for [CorePlugin.Console.RuntimeConsole](#)

### 6.50.2 Member Function Documentation

#### 6.50.2.1 Initialize()

```
override CountDisplayer Initialize (
    ConsoleIcons icons ) [inline], [virtual]
```

Initialize [CountDisplayer](#) with icons

## Parameters

<i>icons</i>	
--------------	--

## Returns

Reimplemented from [CountDisplayer](#).

References [CountDisplayer.Initialize\(\)](#).

### 6.50.2.2 SetInteractionAction()

```
override CountDisplayer SetInteractionAction (
    Action< LogType, bool > onInteractWithDisplayer ) [inline], [virtual]
```

Setting action when interaction with [CountDisplayer](#) happens

## Parameters

<i>onInteractWithDisplayer</i>	
--------------------------------	--

## Returns

Implements [CountDisplayer](#).

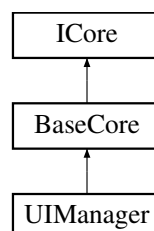
The documentation for this class was generated from the following file:

- [ToggleCountDisplayer.cs](#)

## 6.51 UIManager Class Reference

Base implementation of [UIManager](#).

Inheritance diagram for UIManager:



## Public Member Functions

- override void [InitializeElements](#) ()  
*Initializing all managers elements.*

### 6.51.1 Detailed Description

Base implementation of [UIManager](#).

### 6.51.2 Member Function Documentation

#### 6.51.2.1 InitializeElements()

```
override void InitializeElements ( ) [inline], [virtual]
```

Initializing all managers elements.

Reimplemented from [BaseCore](#).

References [SubPagesController.AddPage\(\)](#).

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

- [UIManager.cs](#)

## 6.52 UIPage Class Reference

[UI](#) page implementation

Inherits [MonoBehaviour](#).

## Public Member Functions

- [UIPage Initialize](#) ()  
*Initializes elements for this [UIPage](#)*

### 6.52.1 Detailed Description

[UI](#) page implementation

### 6.52.2 Member Function Documentation

### 6.52.2.1 Initialize()

```
UIPage Initialize ( ) [inline]
```

Initializes elements for this [UIPage](#)

Returns

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

- [UIPage.cs](#)

## 6.53 UIStateTools Class Reference

UI state tool for canvas groups

### Static Public Member Functions

- static void [ChangeGroupState](#) (CanvasGroup group, bool isVisible)  
*Changing canvas visibility and interactivity*
- static void [ChangeCursorState](#) (bool state)  
*Changing state of mouse cursor*
- static IEnumerator [ChangeGroupState](#) (CanvasGroup group, bool isVisible, float delay)  
*Changing canvas visibility and interactivity after delay*

### 6.53.1 Detailed Description

UI state tool for canvas groups

### 6.53.2 Member Function Documentation

#### 6.53.2.1 ChangeCursorState()

```
static void ChangeCursorState (
    bool state ) [inline], [static]
```

Changing state of mouse cursor

Parameters

<i>state</i>	
--------------	--

### 6.53.2.2 ChangeGroupState() [1/2]

```
static void ChangeGroupState (
    CanvasGroup group,
    bool isVisible ) [inline], [static]
```

Changing canvas visibility and interactivity

#### Parameters

<i>group</i>	
<i>isVisible</i>	

Referenced by SubPagesController.DelayedOpenPage(), SubPagesController.HideAllTables(), SubPagesController.OpenPage(), ConsoleMessage.SetActive(), MinimizedConsole.SetActive(), and RuntimeConsole.SetActive().

### 6.53.2.3 ChangeGroupState() [2/2]

```
static IEnumerator ChangeGroupState (
    CanvasGroup group,
    bool isVisible,
    float delay ) [inline], [static]
```

Changing canvas visibility and interactivity after delay

#### Parameters

<i>group</i>	
<i>isVisible</i>	
<i>delay</i>	

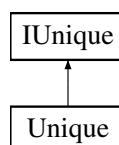
The documentation for this class was generated from the following file:

- UIStateTools.cs

## 6.54 Unique Class Reference

Base class for classes that need to be saved in Json file.

Inheritance diagram for Unique:



### 6.54.1 Detailed Description

Base class for classes that need to be saved in Json file.

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

- Unique.cs

## 6.55 UnityEditorExtension Class Reference

[Extensions](#) for Unity [Editor](#) classes

### Static Public Member Functions

- static void [HelpBox](#) (string message, MessageType type)  
*Override for default Inspector HelpBox with RTF text*
- static void [HelpBox](#) (string message, MessageType type, GUIStyle style)  
*Override for default Inspector HelpBox with style*
- static string [IconName](#) (MessageType type)  
*Getting Icon Name from Unity Inspector*

### 6.55.1 Detailed Description

[Extensions](#) for Unity [Editor](#) classes

### 6.55.2 Member Function Documentation

#### 6.55.2.1 HelpBox() [1/2]

```
static void HelpBox (
    string message,
    MessageType type ) [inline], [static]
```

Override for default Inspector HelpBox with RTF text

#### Parameters

<i>message</i>	
<i>type</i>	

### 6.55.2.2 HelpBox() [2/2]

```
static void HelpBox (  
    string message,  
    MessageType type,  
    GUIStyle style ) [inline], [static]
```

Override for default Inspector HelpBox with style

#### Parameters

<i>message</i>	
<i>type</i>	
<i>style</i>	

References UnityEditorExtension.IconName().

### 6.55.2.3 IconName()

```
static string IconName (  
    MessageType type ) [inline], [static]
```

Getting Icon Name from Unity Inspector

#### Parameters

<i>type</i>	
-------------	--

#### Returns

Referenced by UnityEditorExtension.HelpBox().

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

- UnityEditorExtension.cs

## 6.56 UnityExtensions Class Reference

Extension class for default Unity classes



## Static Public Member Functions

- static bool [IsNotNullAndNotEqual](#) (this [IUnique](#) lci, [IUnique](#) rci)  
*Checks whether the left item is null and doesn't equal right item*
- static List< T > [RemoveRange< T >](#) (this List< T > list, IEnumerable< T > enumerable)  
*Removing range of items from list*
- static void [SnapToLatest](#) (this ScrollRect scrollRect, bool reverseOrder)  
*Scrolls ScrollRect to last item*
- static void [CopyToClipboard](#) (this string str)  
*Puts the string into the Clipboard.*
- static bool [IsNotNullAndEqual](#) (this [IUnique](#) lci, [IUnique](#) rci)  
*Checks whether the left item is null and equals right item*
- static void [Clear< T >](#) (ref List< T > list)  
*Clearing list and destroying its items*
- static bool [TryFindObjectOfType< T >](#) (out T result)  
*Trying to find object on scene that was inherited from T*
- static bool [TryFindObjectsOfType< T >](#) (out IEnumerable< T > result)  
*Trying to find objects on scene that were inherited from T*
- static bool [TryFindObjectsOfType< T >](#) (out IList< T > result)  
*Trying to find objects on scene that were inherited from T and return a list of said objects*

### 6.56.1 Detailed Description

Extension class for default Unity classes

### 6.56.2 Member Function Documentation

#### 6.56.2.1 [Clear< T >\(\)](#)

```
static void Clear< T > (
    ref List< T > list ) [inline], [static]
```

Clearing list and destroying its items

##### Parameters

<i>list</i>	
-------------	--

##### Template Parameters

<i>T</i>	
----------	--

##### Type Constraints

***T* : [MonoBehaviour](#)**

### 6.56.2.2 CopyToClipboard()

```
static void CopyToClipboard (  
    this string str ) [inline], [static]
```

Puts the string into the Clipboard.

### 6.56.2.3 IsNotNullAndEqual()

```
static bool IsNotNullAndEqual (  
    this IUnique lci,  
    IUnique rci ) [inline], [static]
```

Checks whether the left item is null and equals right item

#### Parameters

<i>lci</i>	Left compare item
<i>rci</i>	Right compare item

#### Returns

### 6.56.2.4 IsNotNullAndNotEqual()

```
static bool IsNotNullAndNotEqual (  
    this IUnique lci,  
    IUnique rci ) [inline], [static]
```

Checks whether the left item is null and doesn't equal right item

#### Parameters

<i>lci</i>	Left compare item
<i>rci</i>	Right compare item

#### Returns

### 6.56.2.5 RemoveRange< T >()

```
static List<T> RemoveRange< T > (
    this List< T > list,
    IEnumerable< T > enumerable ) [inline], [static]
```

Removing range of items from list

#### Parameters

<i>list</i>	
<i>enumerable</i>	

#### Template Parameters

<i>T</i>	
----------	--

#### Returns

### 6.56.2.6 SnapToLatest()

```
static void SnapToLatest (
    this ScrollRect scrollRect,
    bool reverseOrder ) [inline], [static]
```

Scrolls ScrollRect to last item

#### Parameters

<i>scrollRect</i>	
<i>reverseOrder</i>	

### 6.56.2.7 TryFindObjectOfType< T >()

```
static bool TryFindObjectOfType< T > (
    out T result ) [inline], [static]
```

Trying to find object on scene that was inherited from T

#### Parameters

<i>result</i>	
---------------	--

## Template Parameters

<i>T</i>	
----------	--

## Returns

**6.56.2.8 TryToFindObjectsOfType< T >() [1/2]**

```
static bool TryToFindObjectsOfType< T > (
    out IEnumerable< T > result )    [inline], [static]
```

Trying to find objects on scene that were inherited from T

## Parameters

<i>result</i>	
---------------	--

## Template Parameters

<i>T</i>	
----------	--

## Returns

**6.56.2.9 TryToFindObjectsOfType< T >() [2/2]**

```
static bool TryToFindObjectsOfType< T > (
    out IList< T > result )    [inline], [static]
```

Trying to find objects on scene that were inherited from T and return a list of said objects

## Parameters

<i>result</i>	
---------------	--

## Template Parameters

<i>T</i>	
----------	--

## Returns

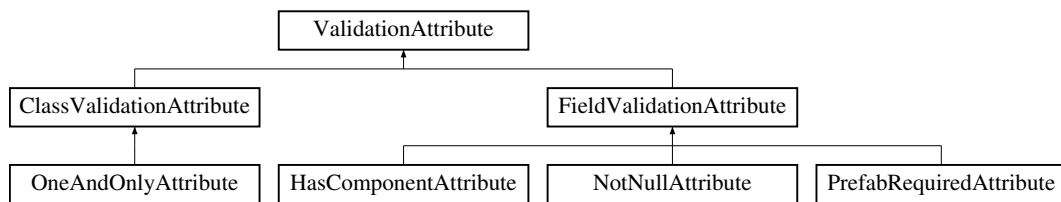
The documentation for this class was generated from the following file:

- UnityExtensions.cs

## 6.57 ValidationAttribute Class Reference

[Base](#) attribute for validation

Inheritance diagram for ValidationAttribute:



### 6.57.1 Detailed Description

[Base](#) attribute for validation

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

- ValidationAttribute.cs

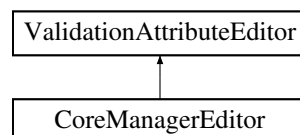
## 6.58 ValidationAttributeEditor Class Reference

Custom editor for Unity.Object class. If you want to create your own custom editor inherit from this class.

## See also

ValidationAttribute, FieldValidationAttribute, ClassValidationAttribute

Inheritance diagram for ValidationAttributeEditor:



### 6.58.1 Detailed Description

Custom editor for Unity.Object class. If you want to create your own custom editor inherit from this class.

## See also

ValidationAttribute, FieldValidationAttribute, ClassValidationAttribute

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

- ValidationAttributeEditor.cs



# Index

- AddData< T >
  - SceneDataHandler, [52](#)
- AddHandler
  - EventInitializer, [28](#)
- AddPage
  - SubPagesController, [61](#)
- AskReference< T >
  - ReferenceDistributor, [45](#)
- BaseCore, [19](#)
  - InitializeElements, [19](#)
- ChangeCursorState
  - UIStateTools, [67](#)
- ChangeGroupState
  - UIStateTools, [68](#)
- ClassValidationAttribute, [20](#)
- Clear< T >
  - UnityExtensions, [71](#)
- ClearHighlight
  - ConsoleMessage, [22](#)
- ConsoleIcons, [20](#)
- ConsoleInitializer, [21](#)
- ConsoleMessage, [21](#)
  - ClearHighlight, [22](#)
  - HighlightText, [22](#)
  - Initialize, [22](#)
  - SetActive, [23](#)
  - SubscribeOnButtonClick, [23](#)
- ConsoleTextSettings, [23](#)
- CopyToClipboard
  - UnityExtensions, [72](#)
- CoreManager, [24](#)
- CoreManagerEditor, [24](#)
- CorePlugin, [11](#)
- CorePlugin.Attributes, [11](#)
- CorePlugin.Attributes.Editor, [11](#)
- CorePlugin.Attributes.Editor.Drawers, [11](#)
- CorePlugin.Attributes.Headers, [11](#)
- CorePlugin.Attributes.Validation, [12](#)
- CorePlugin.Attributes.Validation.Base, [12](#)
- CorePlugin.Console, [12](#)
- CorePlugin.Console.ConsoleElements, [13](#)
- CorePlugin.Core, [13](#)
- CorePlugin.Core.Interface, [13](#)
- CorePlugin.Cross, [13](#)
- CorePlugin.Cross.Events, [13](#)
- CorePlugin.Cross.Events.Interface, [14](#)
- CorePlugin.Cross.SceneData, [14](#)
- CorePlugin.Cross.SceneData.Interface, [14](#)
- CorePlugin.Dispatchers, [14](#)
- CorePlugin.Editor, [14](#)
- CorePlugin.Extensions, [15](#)
- CorePlugin.FileSystem, [15](#)
- CorePlugin.Logger, [15](#)
- CorePlugin.ReferenceDistribution, [15](#)
- CorePlugin.ReferenceDistribution.Interface, [15](#)
- CorePlugin.Samples, [16](#)
- CorePlugin.Samples.Scripts, [16](#)
- CorePlugin.Samples.Scripts.Demo, [16](#)
- CorePlugin.Samples.Scripts.EventTypes, [16](#)
- CorePlugin.Samples.Scripts.Managers, [16](#)
- CorePlugin.Samples.Scripts.Model, [16](#)
- CorePlugin.SceneManagement, [16](#)
- CorePlugin.Serializable, [16](#)
- CorePlugin.Serializable.Interface, [16](#)
- CorePlugin.Singletons, [17](#)
- CorePlugin.UISystem, [17](#)
- CorePlugin.UISystem.UI, [17](#)
- CountDisplay, [25](#)
  - Initialize, [25](#)
  - OnLogCountChanged, [26](#)
  - SetInteractionAction, [26](#)
- DebugLogger, [27](#)
- DelayedOpenPage
  - SubPagesController, [62](#)
- EventInitializer, [27](#)
  - AddHandler, [28](#)
  - InitializeSubscriptions, [28](#)
  - InvokeBase, [28](#)
  - RemoveHandler, [28](#)
  - Subscribe, [29](#)
  - Unsubscribe, [29](#)
- EventTypes, [29](#)
- FieldValidationAttribute, [30](#)
- FileSystem, [30](#)
  - FileSystem, [31](#)
  - Load< T >, [32](#)
  - Save< T >, [33](#)
- GetData< T >
  - SceneDataHandler, [52](#)
- GetInterfaceReference< T >
  - ReferenceDistributor, [46](#)
- GetInterfaceReferences< T >
  - ReferenceDistributor, [46](#)
- GetReference< T >

- ReferenceDistributor, 47
- GetReferences< T >
  - ReferenceDistributor, 47
- GetSubscribers
  - IEventSubscriber, 38
- HasComponentAttribute, 34
- HelpBox
  - UnityEditorExtension, 69
- HideAllTables
  - SubPagesController, 62
- HighlightText
  - ConsoleMessage, 22
- IconName
  - UnityEditorExtension, 70
- ICore, 34
  - InitializeElements, 35
- IDistributingReference, 35
- EventHandler, 35
  - InvokeEvents, 36
  - Subscribe, 36
  - Unsubscribe, 37
- IEventSubscriber, 37
  - GetSubscribers, 38
- Initialize
  - ConsoleMessage, 22
  - CountDisplay, 25
  - MinimizeCountDisplay, 40
  - MinimizedConsole, 41
  - ReferenceDistributor, 48
  - RuntimeConsole, 50
  - ToggleCountDisplay, 64
  - UIPage, 66
- InitializeElements
  - BaseCore, 19
  - ICore, 35
  - UIManager, 66
- InitializeSubscriptions
  - EventInitializer, 28
- InvokeBase
  - EventInitializer, 28
- InvokeEvents
  - EventHandler, 36
- ISceneData, 38
- IsNullAndEqual
  - UnityExtensions, 72
- IsNullAndNotEqual
  - UnityExtensions, 72
- ItemHider, 38
- IUnique, 39
- Load< T >
  - FileSystem, 32
- LoadSceneAsync
  - SceneLoader, 54
- LoadSceneOptions, 39
- MinimizeCountDisplay, 40
- Initialize, 40
  - SetInteractionAction, 40
- MinimizedConsole, 41
  - Initialize, 41
  - SetActive, 42
- MovableInsideScreen, 42
- Named< TName, TKey, TValue >, 42
- NotNullAttribute, 43
- OneAndOnlyAttribute, 43
- OnEnable
  - SymbolDefiner, 63
- OnGUI
  - RequireInterfaceDrawer, 49
- OnLogCountChanged
  - CountDisplay, 26
- OpenPage
  - SubPagesController, 62
- PrefabHeaderAttribute, 44
- PrefabRequiredAttribute, 44
- ReferenceDistributor, 45
  - AskReference< T >, 45
  - GetInterfaceReference< T >, 46
  - GetInterfaceReferences< T >, 46
  - GetReference< T >, 47
  - GetReferences< T >, 47
  - Initialize, 48
- ReferencesHeaderAttribute, 48
- RemoveData< T >
  - SceneDataHandler, 53
- RemoveHandler
  - EventInitializer, 28
- RemoveRange< T >
  - UnityExtensions, 72
- RequireInterfaceAttribute, 48
  - RequireInterfaceAttribute, 49
- RequireInterfaceDrawer, 49
  - OnGUI, 49
- RuntimeConsole, 50
  - Initialize, 50
  - SetActive, 51
- Save< T >
  - FileSystem, 33
- SceneDataHandler, 51
  - AddData< T >, 52
  - GetData< T >, 52
  - RemoveData< T >, 53
- SceneLoader, 53
  - LoadSceneAsync, 54
- SceneLoaderAsset, 54
- SceneLoaderExtensions, 55
  - SceneLoadOperation, 55
  - SceneUnloadOperation, 56
  - Until, 57
- SceneLoaderSettings, 57



- SceneLoaderSettingsEditor, [58](#)
- SceneLoaderSettingsValidator, [58](#)
- SceneLoadOperation
  - SceneLoaderExtensions, [55](#)
- SceneUnloadOperation
  - SceneLoaderExtensions, [56](#)
- SetActive
  - ConsoleMessage, [23](#)
  - MinimizedConsole, [42](#)
  - RuntimeConsole, [51](#)
- SetInteractionAction
  - CountDisplay, [26](#)
  - MinimizeCountDisplay, [40](#)
  - ToggleCountDisplay, [65](#)
- SettingsHeaderAttribute, [59](#)
- ShowSymbolsButtons
  - SymbolDefiner, [64](#)
- Singleton< T >, [59](#)
- SnapToLatest
  - UnityExtensions, [73](#)
- StateHeaderAttribute, [60](#)
- StaticObjectSingleton< T >, [60](#)
- SubPagesController, [60](#)
  - AddPage, [61](#)
  - DelayedOpenPage, [62](#)
  - HideAllTables, [62](#)
  - OpenPage, [62](#)
- Subscribe
  - EventInitializer, [29](#)
  - IEventHandler, [36](#)
- SubscribeOnButtonClick
  - ConsoleMessage, [23](#)
- SymbolDefiner, [63](#)
  - OnEnable, [63](#)
  - ShowSymbolsButtons, [64](#)
- ToggleCountDisplay, [64](#)
  - Initialize, [64](#)
  - SetInteractionAction, [65](#)
- TryFindObjectOfType< T >
  - UnityExtensions, [73](#)
- TryFindObjectsOfType< T >
  - UnityExtensions, [74](#)
- UIManager, [65](#)
  - InitializeElements, [66](#)
- UIPage, [66](#)
  - Initialize, [66](#)
- UIStateTools, [67](#)
  - ChangeCursorState, [67](#)
  - ChangeGroupState, [68](#)
- Unique, [68](#)
- UnityEditorExtension, [69](#)
  - HelpBox, [69](#)
  - IconName, [70](#)
- UnityExtensions, [70](#)
  - Clear< T >, [71](#)
  - CopyToClipboard, [72](#)
  - IsNotNullAndEqual, [72](#)
  - IsNotNullAndNotEqual, [72](#)
  - RemoveRange< T >, [72](#)
  - SnapToLatest, [73](#)
  - TryFindObjectOfType< T >, [73](#)
  - TryFindObjectsOfType< T >, [74](#)
- Unsubscribe
  - EventInitializer, [29](#)
  - IEventHandler, [37](#)
- Until
  - SceneLoaderExtensions, [57](#)
- ValidationAttribute, [75](#)
- ValidationAttributeEditor, [75](#)