

# Core Manager

0.0.1

Generated by Doxygen 1.9.1



<b>1 CoreManager</b>	<b>1</b>
<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.Console Namespace Reference . . . . .	11
5.3 CorePlugin.Core Namespace Reference . . . . .	11
5.4 CorePlugin.Core.Interface Namespace Reference . . . . .	12
5.5 CorePlugin.Cross Namespace Reference . . . . .	12
5.6 CorePlugin.Cross.Events Namespace Reference . . . . .	12
5.7 CorePlugin.Cross.Events.Interface Namespace Reference . . . . .	12
5.8 CorePlugin.Cross.SceneData Namespace Reference . . . . .	12
5.9 CorePlugin.Cross.SceneData.Interface Namespace Reference . . . . .	12
5.10 CorePlugin.CustomAttributes Namespace Reference . . . . .	13
5.11 CorePlugin.CustomAttributes.Editor Namespace Reference . . . . .	13
5.12 CorePlugin.CustomAttributes.Editor.Drawers Namespace Reference . . . . .	13
5.13 CorePlugin.CustomAttributes.Headers Namespace Reference . . . . .	13
5.14 CorePlugin.CustomAttributes.Validation Namespace Reference . . . . .	13
5.15 CorePlugin.CustomAttributes.Validation.Base Namespace Reference . . . . .	14
5.16 CorePlugin.Editor Namespace Reference . . . . .	14
5.17 CorePlugin.Extensions Namespace Reference . . . . .	14
5.18 CorePlugin.FileSystem Namespace Reference . . . . .	14
5.19 CorePlugin.Interface Namespace Reference . . . . .	15
5.20 CorePlugin.Logger Namespace Reference . . . . .	15
5.21 CorePlugin.ReferenceDistribution Namespace Reference . . . . .	15
5.22 CorePlugin.ReferenceDistribution.Interface Namespace Reference . . . . .	15
5.23 CorePlugin.Samples Namespace Reference . . . . .	15
5.24 CorePlugin.Samples.Scripts Namespace Reference . . . . .	15
5.25 CorePlugin.Samples.Scripts.Demo Namespace Reference . . . . .	15
5.26 CorePlugin.Samples.Scripts.EventTypes Namespace Reference . . . . .	15
5.27 CorePlugin.Samples.Scripts.Managers Namespace Reference . . . . .	15
5.28 CorePlugin.Samples.Scripts.Model Namespace Reference . . . . .	15
5.29 CorePlugin.Serializable Namespace Reference . . . . .	15
5.30 CorePlugin.Serializable.Interface Namespace Reference . . . . .	16
5.31 CorePlugin.Singletons Namespace Reference . . . . .	16
5.32 CorePlugin.UISystem Namespace Reference . . . . .	16

5.33 CorePlugin.UISystem.UI Namespace Reference	16
<b>6 Class Documentation</b>	<b>17</b>
6.1 BaseCore Class Reference	17
6.1.1 Detailed Description	17
6.1.2 Member Function Documentation	17
6.1.2.1 InitializeElements()	18
6.2 ClassValidationAttribute Class Reference	18
6.2.1 Detailed Description	18
6.3 ConsoleLogToggle Class Reference	18
6.3.1 Detailed Description	18
6.4 ConsoleMessage Class Reference	19
6.4.1 Detailed Description	19
6.4.2 Member Function Documentation	19
6.4.2.1 ClearHighlight()	19
6.4.2.2 HighlightText()	19
6.4.2.3 Initialize()	19
6.4.2.4 SetActive()	20
6.4.2.5 SubscribeOnButtonClick()	20
6.5 ConsoleTextSettings Class Reference	21
6.5.1 Detailed Description	21
6.6 CoreManager Class Reference	21
6.6.1 Detailed Description	21
6.7 CustomInspectorCoreManager Class Reference	21
6.7.1 Detailed Description	22
6.8 CustomLogger Class Reference	22
6.8.1 Detailed Description	22
6.9 EventInitializer Class Reference	22
6.9.1 Detailed Description	23
6.9.2 Member Function Documentation	23
6.9.2.1 AddHandler()	23
6.9.2.2 InitializeSubscriptions()	23
6.9.2.3 InvokeBase()	24
6.9.2.4 RemoveHandler()	24
6.9.2.5 Subscribe()	24
6.9.2.6 Unsubscribe()	24
6.10 EventTypes Class Reference	25
6.10.1 Detailed Description	25
6.11 FieldValidationAttribute Class Reference	25
6.11.1 Detailed Description	25
6.12 FileSystem Class Reference	26
6.12.1 Detailed Description	26

6.12.2 Constructor & Destructor Documentation	26
6.12.2.1 FileSystem() [1/3]	26
6.12.2.2 FileSystem() [2/3]	26
6.12.2.3 FileSystem() [3/3]	27
6.12.3 Member Function Documentation	27
6.12.3.1 Load< T >() [1/2]	27
6.12.3.2 Load< T >() [2/2]	28
6.12.3.3 Save< T >() [1/2]	28
6.12.3.4 Save< T >() [2/2]	29
6.13 HasComponentAttribute Class Reference	29
6.13.1 Detailed Description	30
6.14 ICore Interface Reference	30
6.14.1 Detailed Description	30
6.14.2 Member Function Documentation	30
6.14.2.1 InitializeElements()	30
6.15 IDepClone< out out T > Interface Template Reference	31
6.15.1 Detailed Description	31
6.15.2 Member Function Documentation	31
6.15.2.1 DeepClone() [1/2]	31
6.15.2.2 DeepClone() [2/2]	31
6.16 IDistributingReference Interface Reference	32
6.16.1 Detailed Description	32
6.17 IEventHandler Interface Reference	32
6.17.1 Detailed Description	32
6.17.2 Member Function Documentation	33
6.17.2.1 InvokeEvents()	33
6.17.2.2 Subscribe()	33
6.17.2.3 Unsubscribe()	33
6.18 IEventSubscriber Interface Reference	33
6.18.1 Detailed Description	34
6.18.2 Member Function Documentation	34
6.18.2.1 GetSubscribers()	34
6.19 ISceneData Interface Reference	34
6.19.1 Detailed Description	35
6.20 IUnique Interface Reference	35
6.20.1 Detailed Description	35
6.21 Named< TName, TKey, TValue > Class Template Reference	35
6.21.1 Detailed Description	35
6.22 NotNullAttribute Class Reference	36
6.22.1 Detailed Description	36
6.23 OneAndOnlyAttribute Class Reference	36
6.23.1 Detailed Description	37

6.24 PrefabHeaderAttribute Class Reference . . . . .	37
6.24.1 Detailed Description . . . . .	37
6.25 PrefabRequiredAttribute Class Reference . . . . .	37
6.25.1 Detailed Description . . . . .	37
6.26 ReferenceDistributor Class Reference . . . . .	38
6.26.1 Detailed Description . . . . .	38
6.26.2 Member Function Documentation . . . . .	38
6.26.2.1 AskReference< T >() . . . . .	38
6.26.2.2 GetReference< T >() . . . . .	39
6.26.2.3 GetReferences< T >() . . . . .	39
6.26.2.4 Initialize() . . . . .	40
6.27 ReferencesHeaderAttribute Class Reference . . . . .	40
6.27.1 Detailed Description . . . . .	40
6.28 RuntimeConsole Class Reference . . . . .	40
6.28.1 Detailed Description . . . . .	40
6.29 SceneDataHandler Class Reference . . . . .	41
6.29.1 Detailed Description . . . . .	41
6.29.2 Member Function Documentation . . . . .	41
6.29.2.1 AddData< T >() . . . . .	41
6.29.2.2 GetData< T >() . . . . .	42
6.29.2.3 RemoveData< T >() . . . . .	42
6.30 SettingsHeaderAttribute Class Reference . . . . .	43
6.30.1 Detailed Description . . . . .	43
6.31 Singleton< T > Class Template Reference . . . . .	43
6.31.1 Detailed Description . . . . .	43
6.32 StateHeaderAttribute Class Reference . . . . .	43
6.32.1 Detailed Description . . . . .	44
6.33 SubPagesController Class Reference . . . . .	44
6.33.1 Detailed Description . . . . .	44
6.33.2 Member Function Documentation . . . . .	44
6.33.2.1 AddPage() [1/2] . . . . .	44
6.33.2.2 AddPage() [2/2] . . . . .	45
6.33.2.3 DelayedOpenPage() [1/2] . . . . .	45
6.33.2.4 DelayedOpenPage() [2/2] . . . . .	45
6.33.2.5 HideAllTables() . . . . .	46
6.33.2.6 OpenPage() . . . . .	46
6.34 SymbolDefiner Class Reference . . . . .	46
6.34.1 Detailed Description . . . . .	47
6.34.2 Member Function Documentation . . . . .	47
6.34.2.1 OnEnable() . . . . .	47
6.34.2.2 ShowSymbolsButtons() . . . . .	47
6.35 UIManager Class Reference . . . . .	47

6.35.1 Detailed Description	48
6.35.2 Member Function Documentation	48
6.35.2.1 InitializeElements()	48
6.36 UIPage Class Reference	48
6.36.1 Detailed Description	48
6.36.2 Member Function Documentation	48
6.36.2.1 Initialize()	49
6.37 UIStateTools Class Reference	49
6.37.1 Detailed Description	49
6.37.2 Member Function Documentation	49
6.37.2.1 ChangeCursorState()	49
6.37.2.2 ChangeGroupState() [1/2]	50
6.37.2.3 ChangeGroupState() [2/2]	50
6.38 Unique Class Reference	50
6.38.1 Detailed Description	51
6.39 UnityEditorExtension Class Reference	51
6.39.1 Detailed Description	51
6.39.2 Member Function Documentation	51
6.39.2.1 HelpBox() [1/2]	51
6.39.2.2 HelpBox() [2/2]	52
6.39.2.3 IconName()	52
6.40 UnityExtensions Class Reference	52
6.40.1 Detailed Description	53
6.40.2 Member Function Documentation	53
6.40.2.1 Clear< T >()	53
6.40.2.2 IsNotNullAndEqual()	54
6.40.2.3 IsNotNullAndNotEqual()	54
6.40.2.4 TryFindObjectOfType< T >()	54
6.40.2.5 TryFindObjectOfType< T >() [1/2]	55
6.40.2.6 TryFindObjectOfType< T >() [2/2]	55
6.41 ValidationAttribute Class Reference	56
6.41.1 Detailed Description	56
6.42 ValidationAttributeEditor Class Reference	56
6.42.1 Detailed Description	57
<b>Index</b>	<b>59</b>





## Chapter 1

# CoreManager

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



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



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

ConsoleLogToggle . . . . .	18
ConsoleMessage . . . . .	19
ConsoleTextSettings . . . . .	21
CoreManager . . . . .	21
CustomLogger . . . . .	22
EventInitializer . . . . .	22
EventTypes . . . . .	25
FileSystem . . . . .	26
ICore . . . . .	30
BaseCore . . . . .	17
UIManager . . . . .	47
IDeepClone< out out T > . . . . .	31
IDistributingReference . . . . .	32
IEventHandler . . . . .	32
IEventSubscriber . . . . .	33
ISceneData . . . . .	34
IUnique . . . . .	35
Unique . . . . .	50
Named< TName, TKey, TValue > . . . . .	35
Named< string, Object > . . . . .	35
PrefabHeaderAttribute . . . . .	37
ReferenceDistributor . . . . .	38
ReferencesHeaderAttribute . . . . .	40
RuntimeConsole . . . . .	40
SettingsHeaderAttribute . . . . .	43
Singleton< T > . . . . .	43
Singleton< SceneDataHandler > . . . . .	43
SceneDataHandler . . . . .	41
StateHeaderAttribute . . . . .	43
SubPagesController . . . . .	44
SymbolDefiner . . . . .	46
UIPage . . . . .	48
UIStateTools . . . . .	49
UnityEditorExtension . . . . .	51

UnityExtensions . . . . .	52
ValidationAttribute . . . . .	56
ClassValidationAttribute . . . . .	18
OneAndOnlyAttribute . . . . .	36
FieldValidationAttribute . . . . .	25
HasComponentAttribute . . . . .	29
NotNullAttribute . . . . .	36
PrefabRequiredAttribute . . . . .	37
ValidationAttributeEditor . . . . .	56
CustomInspectorCoreManager . . . . .	21

## 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> . . . . .	17
<a href="#">ClassValidationAttribute</a>	
Base attribute for class validation . . . . .	18
<a href="#">ConsoleLogToggle</a>	
Log toggle for <a href="#">CorePlugin.Console.RuntimeConsole</a> . . . . .	18
<a href="#">ConsoleMessage</a>	
Log message for <a href="#">CorePlugin.Console.RuntimeConsole</a> . . . . .	19
<a href="#">ConsoleTextSettings</a>	
Settings class for <a href="#">CorePlugin.Console.RuntimeConsole</a> . . . . .	21
<a href="#">21</a>	
<a href="#">21</a>	
<a href="#">CustomLogger</a>	
Custom logger solution for logs . . . . .	22
<a href="#">22</a>	
<a href="#">EventTypes</a>	
Declare your delegate type to convert method group to delegate . . . . .	25
<a href="#">FieldValidationAttribute</a>	
Base attribute for field validation . . . . .	25
<a href="#">26</a>	

<a href="#">HasComponentAttribute</a>	Attribute validating whether Object in field or all objects in the list have desired component . . . . .	29
<a href="#">ICore</a>	<a href="#">Interface</a> for Manager declaration . . . . .	30
<a href="#">IDeepClone&lt; out out T &gt;</a>	<a href="#">Interface</a> for deep cloning . . . . .	31
<a href="#">IDistributingReference</a>	Flag interface for MonoBehaviour classes what need to referenced from another class . . . . .	32
<a href="#">IEventHandler</a>	<a href="#">Interface</a> for event handler . . . . .	32
<a href="#">IEventSubscriber</a>	<a href="#">Interface</a> for subscribers . . . . .	33
34		
<a href="#">IUnique</a>	<a href="#">Interface</a> 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 . . . . .	35
<a href="#">Named&lt; TName, TKey, TValue &gt;</a>	Replacement for dictionary in Unity Inspector . . . . .	35
<a href="#">NotNullAttribute</a>	Attribute validating whether field or element in the list equals null . . . . .	36
<a href="#">OneAndOnlyAttribute</a>	Attribute validating whether there is only one copy of this class in the scene . . . . .	36
<a href="#">PrefabHeaderAttribute</a>	Replacement for Header("Prefabs") . . . . .	37
<a href="#">PrefabRequiredAttribute</a>	Attribute validating whether the object or all items in list are prefabs . . . . .	37
<a href="#">ReferenceDistributor</a>	Class responsible for reference distribution inside one scene . . . . .	38
<a href="#">ReferencesHeaderAttribute</a>	Replacement for Header("References") . . . . .	40
<a href="#">RuntimeConsole</a>	Runtime console class . . . . .	40



<a href="#">SettingsHeaderAttribute</a>	
Replacement for Header("Settings") . . . . .	43
<a href="#">Singleton&lt; T &gt;</a>	
Base for all singletons. Strongly recommended to use singletons as little as possible . . . . .	43
<a href="#">StateHeaderAttribute</a>	
Replacement for Header("State") . . . . .	43
<a href="#">SubPagesController</a>	
Class designated for <a href="#">UI</a> subpages . . . . .	44
46	
<a href="#">UIManager</a>	
Base implementation of <a href="#">UIManager</a> . . . . .	47
<a href="#">UIPage</a>	
<a href="#">UI</a> page implementation . . . . .	48
<a href="#">UIStateTools</a>	
UI state tool for canvas groups . . . . .	49
<a href="#">Unique</a>	
Base class for classes that need to be saved in Json file . . . . .	50
<a href="#">UnityEditorExtension</a>	
<a href="#">Extensions</a> for Unity <a href="#">Editor</a> classes . . . . .	51
<a href="#">UnityExtensions</a>	
Extension class for default Unity classes . . . . .	52
<a href="#">ValidationAttribute</a>	
<a href="#">Base</a> attribute for validation . . . . .	56

56



## Chapter 5

# Namespace Documentation

### 5.1 CorePlugin Namespace Reference

### 5.2 CorePlugin.Console Namespace Reference

#### Classes

- class [ConsoleLogToggle](#)  
*Log toggle for [CorePlugin.Console.RuntimeConsole](#)*
- class [ConsoleMessage](#)  
*Log message for [CorePlugin.Console.RuntimeConsole](#)*
- class [ConsoleTextSettings](#)  
*Settings class for [CorePlugin.Console.RuntimeConsole](#)*
- class [RuntimeConsole](#)  
*Runtime console class*

### 5.3 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](#)*

## 5.4 CorePlugin.Core.Interface Namespace Reference

### Classes

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

## 5.5 CorePlugin.Cross Namespace Reference

## 5.6 CorePlugin.Cross.Events Namespace Reference

### Classes

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

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

### Classes

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

## 5.8 CorePlugin.Cross.SceneData Namespace Reference

### Classes

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

## 5.9 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](#)

## 5.10 CorePlugin.CustomAttributes Namespace Reference

## 5.11 CorePlugin.CustomAttributes.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*

*[CorePlugin.CustomAttributes.Validation.Base.ValidationAttribute](#), [CorePlugin.CustomAttributes.Validation.Base.FieldValidationAttribute](#)  
[CorePlugin.CustomAttributes.Validation.Base.ClassValidationAttribute](#)*

## 5.12 CorePlugin.CustomAttributes.Editor.Drawers Namespace Reference

## 5.13 CorePlugin.CustomAttributes.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")*

## 5.14 CorePlugin.CustomAttributes.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.*

## 5.15 CorePlugin.CustomAttributes.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*

## 5.16 CorePlugin.Editor Namespace Reference

### Classes

- class [CustomInspectorCoreManager](#)  
*Custom [Editor](#) CoreManager class.*  
*See also*  
[CorePlugin.Core.CoreManager](#)
- class [SymbolDefiner](#)  
*Class for "Scripting Define Symbols" defining from CoreManager Inspector.*  
*See also*  
[CorePlugin.Core.CoreManager](#)

## 5.17 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*

## 5.18 CorePlugin.FileSystem Namespace Reference

### Classes

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

## 5.19 CorePlugin.Interface Namespace Reference

### Classes

- interface [IDeepClone](#)  
*Interface for deep cloning*

## 5.20 CorePlugin.Logger Namespace Reference

### Classes

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

## 5.21 CorePlugin.ReferenceDistribution Namespace Reference

### Classes

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

## 5.22 CorePlugin.ReferenceDistribution.Interface Namespace Reference

### Classes

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

## 5.23 CorePlugin.Samples Namespace Reference

## 5.24 CorePlugin.Samples.Scripts Namespace Reference

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

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

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

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

## 5.29 CorePlugin.Serializable Namespace Reference

### Classes

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

## 5.30 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*

## 5.31 CorePlugin.Singletons Namespace Reference

### Classes

- class [Singleton](#)

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

## 5.32 CorePlugin.UISystem Namespace Reference

### Classes

- class [UIManager](#)

*Base implementation of [UIManager](#).*

## 5.33 CorePlugin.UISystem.UI Namespace Reference

### Classes

- class [SubPagesController](#)

*Class designated for [UI](#) subpages.*

- class [UIPage](#)

*[UI](#) page implementation*



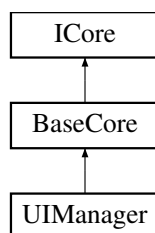
## Chapter 6

# 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](#).

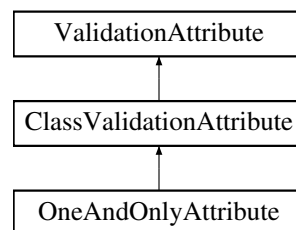
The documentation for this class was generated from the following file:

- 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 ConsoleLogToggle Class Reference

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

Inherits MonoBehaviour.

### 6.3.1 Detailed Description

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

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

- ConsoleLogToggle.cs

## 6.4 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*

### 6.4.1 Detailed Description

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

### 6.4.2 Member Function Documentation

#### 6.4.2.1 ClearHighlight()

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

Clear highlight marks

#### 6.4.2.2 HighlightText()

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

Sets highlight marks

#### 6.4.2.3 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>	
------------------------------------	--

**6.4.2.4 SetActive()**

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

Setting active message in console

**Parameters**

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

References `UIStateTools.ChangeGroupState()`.

**6.4.2.5 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](#)

## 6.5 ConsoleTextSettings Class Reference

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

### 6.5.1 Detailed Description

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

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

- [ConsoleTextSettings.cs](#)

## 6.6 CoreManager Class Reference

Manager for initialization of sub manager in the scene.

See also

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

Inherits [MonoBehaviour](#).

### 6.6.1 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](#)

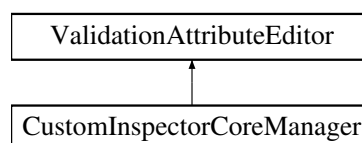
## 6.7 CustomInspectorCoreManager Class Reference

Custom [Editor](#) [CoreManager](#) class.

See also

[CorePlugin.Core.CoreManager](#)

Inheritance diagram for [CustomInspectorCoreManager](#):



### 6.7.1 Detailed Description

Custom [Editor](#) CoreManager class.

See also

[CorePlugin.Core.CoreManager](#)

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

- CustomInspectorCoreManager.cs

## 6.8 CustomLogger Class Reference

Custom logger solution for logs.

### 6.8.1 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:

- Logger.cs

## 6.9 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.*

### 6.9.1 Detailed Description

Class responsible for reference initialization

See also

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

## 6.9.2 Member Function Documentation

### 6.9.2.1 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\(\)](#).

### 6.9.2.2 InitializeSubscriptions()

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

Initialising cross subscriptions for all handlers in the scene.

### 6.9.2.3 InvokeBase()

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

Invoking event on handlers.

### 6.9.2.4 RemoveHandler()

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

Removing event handler after scene has Awoken from list

#### Parameters

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

### 6.9.2.5 Subscribe()

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

Subscribing event subscriber after scene has Awoken to event handlers.

#### Parameters

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

### 6.9.2.6 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

## 6.10 EventTypes Class Reference

Declare your delegate type to convert method group to delegate.

Inherited by CustomEventTypes.

### 6.10.1 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](#)

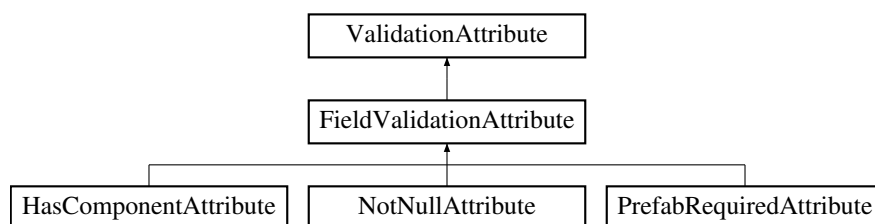
The documentation for this class was generated from the following file:

- EventTypes.cs

## 6.11 FieldValidationAttribute Class Reference

[Base](#) attribute for field validation.

Inheritance diagram for FieldValidationAttribute:



### 6.11.1 Detailed Description

[Base](#) attribute for field validation.

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

- FieldValidationAttribute.cs

## 6.12 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.*

### 6.12.1 Detailed Description

Class for saving Json file to disk.

See also

[CorePlugin.Serializable.Unique](#)

### 6.12.2 Constructor & Destructor Documentation

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

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

Default constructor.

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

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

Overloaded constructor which changes default path.

## Parameters

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

## 6.12.2.3 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>	

## 6.12.3 Member Function Documentation

## 6.12.3.1 Load&lt; T &gt;() [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**

**6.12.3.2 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***

**6.12.3.3 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***

**6.12.3.4 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**

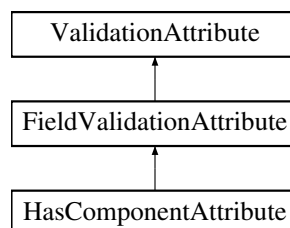
The documentation for this class was generated from the following file:

- FileSystem.cs

**6.13 HasComponentAttribute Class Reference**

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

Inheritance diagram for HasComponentAttribute:



### 6.13.1 Detailed Description

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

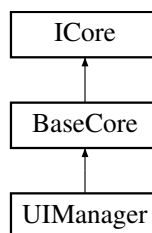
The documentation for this class was generated from the following file:

- HasComponentAttribute.cs

## 6.14 ICore Interface Reference

[Interface](#) for Manager declaration.

Inheritance diagram for ICore:



### Public Member Functions

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

### 6.14.1 Detailed Description

[Interface](#) for Manager declaration.

### 6.14.2 Member Function Documentation

#### 6.14.2.1 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

## 6.15 IDepClone< out out T > Interface Template Reference

[Interface](#) for deep cloning

Inherits IDepClone.

### Public Member Functions

- new T [DeepClone](#) ()  
*Clones instance of an object*
- object [DeepClone](#) ()  
*Clones instance of an object*

### 6.15.1 Detailed Description

[Interface](#) for deep cloning

### 6.15.2 Member Function Documentation

#### 6.15.2.1 DeepClone() [1/2]

```
new T DeepClone ( )
```

Clones instance of an object

**Returns**

#### 6.15.2.2 DeepClone() [2/2]

```
object DeepClone ( )
```

Clones instance of an object

**Returns**

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

- IDepClone.cs

## 6.16 IDistributingReference Interface Reference

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

### 6.16.1 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](#)

## 6.17 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](#) (IEnumerable< Delegate > subscribers)  
*Subscribing delegates to event*
- void [Unsubscribe](#) (IEnumerable< Delegate > unsubscribers)  
*Unsubscribing delegates to event*

### 6.17.1 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](#)



## 6.17.2 Member Function Documentation

### 6.17.2.1 InvokeEvents()

```
void InvokeEvents ( )
```

Invoking events that need to be invoked on scene initialization.

Referenced by EventInitializer.AddHandler().

### 6.17.2.2 Subscribe()

```
void Subscribe (
    IEnumerable< Delegate > subscribers )
```

Subscribing delegates to event

#### Parameters

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

Referenced by EventInitializer.AddHandler().

### 6.17.2.3 Unsubscribe()

```
void Unsubscribe (
    IEnumerable< Delegate > unsubscribers )
```

Unsubscribing delegates to event

#### Parameters

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

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

- IEventHandler.cs

## 6.18 IEventSubscriber Interface Reference

[Interface](#) for subscribers.

Inherited by SceneLoader.

## Public Member Functions

- `IEnumerable< Delegate > GetSubscribers ()`  
*Returns IEnumerable with all methods which need to be subscribed.*

### 6.18.1 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](#)

### 6.18.2 Member Function Documentation

#### 6.18.2.1 [GetSubscribers\(\)](#)

```
IEnumerable<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](#)

## 6.19 ISceneData Interface Reference

Flag interface for classes or structs to pass between scenes

See also

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

Inherited by [DataTransfer](#).

### 6.19.1 Detailed Description

Flag interface for classes or structs to pass between scenes

See also

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

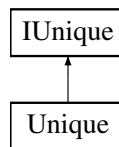
The documentation for this interface was generated from the following file:

- ISceneData.cs

## 6.20 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:



### 6.20.1 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

## 6.21 Named< TName, TKey, TValue > Class Template Reference

Replacement for dictionary in Unity Inspector

### 6.21.1 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>	

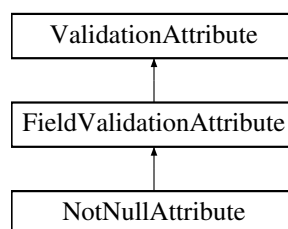
The documentation for this class was generated from the following file:

- Named.cs

## 6.22 NotNullAttribute Class Reference

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

Inheritance diagram for NotNullAttribute:



### 6.22.1 Detailed Description

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

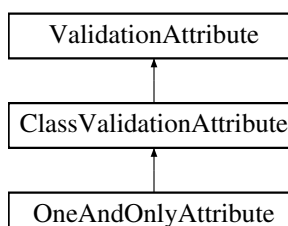
The documentation for this class was generated from the following file:

- NotNullAttribute.cs

## 6.23 OneAndOnlyAttribute Class Reference

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

Inheritance diagram for OneAndOnlyAttribute:



### 6.23.1 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

## 6.24 PrefabHeaderAttribute Class Reference

Replacement for Header("Prefabs")

Inherits HeaderAttribute.

### 6.24.1 Detailed Description

Replacement for Header("Prefabs")

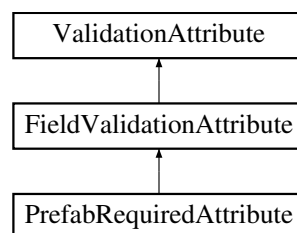
The documentation for this class was generated from the following file:

- PrefabHeaderAttribute.cs

## 6.25 PrefabRequiredAttribute Class Reference

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

Inheritance diagram for PrefabRequiredAttribute:



### 6.25.1 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

## 6.26 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 > ()  
*Getting reference by type from list*
- static bool [AskReference](#)< T > (ref T reference)  
*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*
- static IEnumerable< T > [GetReferences](#)< T > ()  
*Getting references by type from list*

### 6.26.1 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.

### 6.26.2 Member Function Documentation

#### 6.26.2.1 [AskReference](#)< T >()

```
static bool AskReference< T > (
    ref T reference ) [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 >().

**6.26.2.2 GetReference< T >()**

```
static T GetReference< T > ( ) [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 >().

**6.26.2.3 GetReferences< T >()**

```
static IEnumerable<T> GetReferences< T > ( ) [inline], [static]
```

Getting references by type from list

## Template Parameters

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

Returns

Type Constraints

***T : MonoBehaviour***

***T : IDistributingReference***

#### 6.26.2.4 Initialize()

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

Initializing distribution references

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

- ReferenceDistributor.cs

## 6.27 ReferencesHeaderAttribute Class Reference

Replacement for Header("References")

Inherits HeaderAttribute.

### 6.27.1 Detailed Description

Replacement for Header("References")

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

- ReferencesHeaderAttribute.cs

## 6.28 RuntimeConsole Class Reference

Runtime console class

Inherits MonoBehaviour.

### 6.28.1 Detailed Description

Runtime console class

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

- RuntimeConsole.cs



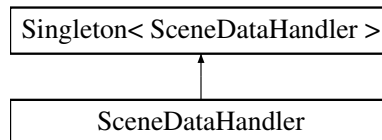
## 6.29 SceneDataHandler Class Reference

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*

### 6.29.1 Detailed Description

Singleton for passing data between scenes

See also

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

### 6.29.2 Member Function Documentation

#### 6.29.2.1 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()*****6.29.2.2    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()*****6.29.2.3    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

## 6.30 SettingsHeaderAttribute Class Reference

Replacement for Header("Settings")

Inherits HeaderAttribute.

### 6.30.1 Detailed Description

Replacement for Header("Settings")

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

- SettingsHeaderAttribute.cs

## 6.31 Singleton< T > Class Template Reference

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

Inherits MonoBehaviour.

### 6.31.1 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

## 6.32 StateHeaderAttribute Class Reference

Replacement for Header("State")

Inherits HeaderAttribute.

### 6.32.1 Detailed Description

Replacement for Header("State")

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

- StateHeaderAttribute.cs

## 6.33 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 UnityEngine.UIAction 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*

### 6.33.1 Detailed Description

Class designated for [UI](#) subpages.

### 6.33.2 Member Function Documentation

#### 6.33.2.1 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().

**6.33.2.2 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().

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

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

Enables page interaction after delay

## Parameters

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

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

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

Enables page interaction after delay

**Parameters**

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

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

**6.33.2.5 HideAllTables()**

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

Hides all pages in this subpage controller

References `UIStateTools.ChangeGroupState()`.

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

**6.33.2.6 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`

**6.34 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*

### 6.34.1 Detailed Description

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

See also

[CorePlugin.Core.CoreManager](#)

### 6.34.2 Member Function Documentation

#### 6.34.2.1 OnEnable()

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

Check for whether the are already defined symbols OnEnable

#### 6.34.2.2 ShowSymbolsButtons()

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

Shows buttons in Inspector.

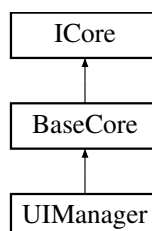
The documentation for this class was generated from the following file:

- SymbolDefiner.cs

## 6.35 UIManager Class Reference

Base implementation of [UIManager](#).

Inheritance diagram for UIManager:



## Public Member Functions

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

### 6.35.1 Detailed Description

Base implementation of [UIManager](#).

### 6.35.2 Member Function Documentation

#### 6.35.2.1 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](#)

## 6.36 UIPage Class Reference

[UI](#) page implementation

Inherits [MonoBehaviour](#).

## Public Member Functions

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

### 6.36.1 Detailed Description

[UI](#) page implementation

### 6.36.2 Member Function Documentation



### 6.36.2.1 Initialize()

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

Initializes elements for this [UIPage](#)

Returns

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

- [UIPage.cs](#)

## 6.37 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*

### 6.37.1 Detailed Description

UI state tool for canvas groups

### 6.37.2 Member Function Documentation

#### 6.37.2.1 ChangeCursorState()

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

Changing state of mouse cursor

Parameters

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

### 6.37.2.2 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(), and ConsoleMessage.SetActive().

### 6.37.2.3 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>	

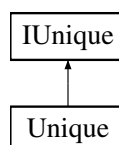
The documentation for this class was generated from the following file:

- UIStateTools.cs

## 6.38 Unique Class Reference

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

Inheritance diagram for Unique:



### 6.38.1 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

## 6.39 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*

### 6.39.1 Detailed Description

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

### 6.39.2 Member Function Documentation

#### 6.39.2.1 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>	

### 6.39.2.2 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().

### 6.39.2.3 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

## 6.40 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 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*

### 6.40.1 Detailed Description

Extension class for default Unity classes

### 6.40.2 Member Function Documentation

#### 6.40.2.1 `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***

#### 6.40.2.2 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

#### 6.40.2.3 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

#### 6.40.2.4 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

**6.40.2.5 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

**6.40.2.6 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

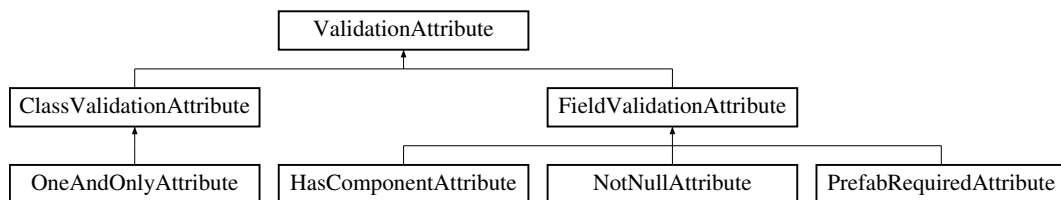
The documentation for this class was generated from the following file:

- UnityExtensions.cs

## 6.41 ValidationAttribute Class Reference

[Base](#) attribute for validation

Inheritance diagram for ValidationAttribute:



### 6.41.1 Detailed Description

[Base](#) attribute for validation

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

- ValidationAttribute.cs

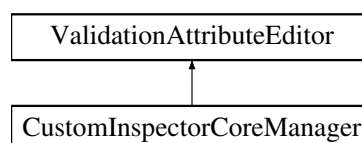
## 6.42 ValidationAttributeEditor Class Reference

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

## See also

[CorePlugin.CustomAttributes.Validation.Base.ValidationAttribute](#), [CorePlugin.CustomAttributes.Validation.Base.FieldValidationAttribute](#), [CorePlugin.CustomAttributes.Validation.Base.ClassValidationAttribute](#)

Inheritance diagram for ValidationAttributeEditor:





### 6.42.1 Detailed Description

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

See also

[CorePlugin.CustomAttributes.Validation.Base.ValidationAttribute](#), [CorePlugin.CustomAttributes.Validation.Base.FieldValidationAttribute](#),  
[CorePlugin.CustomAttributes.Validation.Base.ClassValidationAttribute](#)

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

- ValidationAttributeEditor.cs



# Index

- AddData< T >
  - SceneDataHandler, [41](#)
- AddHandler
  - EventInitializer, [23](#)
- AddPage
  - SubPagesController, [44](#), [45](#)
- AskReference< T >
  - ReferenceDistributor, [38](#)
- BaseCore, [17](#)
  - InitializeElements, [17](#)
- ChangeCursorState
  - UIStateTools, [49](#)
- ChangeGroupState
  - UIStateTools, [50](#)
- ClassValidationAttribute, [18](#)
- Clear< T >
  - UnityExtensions, [53](#)
- ClearHighlight
  - ConsoleMessage, [19](#)
- ConsoleLogToggle, [18](#)
- ConsoleMessage, [19](#)
  - ClearHighlight, [19](#)
  - HighlightText, [19](#)
  - Initialize, [19](#)
  - SetActive, [20](#)
  - SubscribeOnButtonClick, [20](#)
- ConsoleTextSettings, [21](#)
- CoreManager, [21](#)
- CorePlugin, [11](#)
- CorePlugin.Console, [11](#)
- CorePlugin.Core, [11](#)
- CorePlugin.Core.Interface, [12](#)
- CorePlugin.Cross, [12](#)
- CorePlugin.Cross.Events, [12](#)
- CorePlugin.Cross.Events.Interface, [12](#)
- CorePlugin.Cross.SceneData, [12](#)
- CorePlugin.Cross.SceneData.Interface, [12](#)
- CorePlugin.CustomAttributes, [13](#)
- CorePlugin.CustomAttributes.Editor, [13](#)
- CorePlugin.CustomAttributes.Editor.Drawers, [13](#)
- CorePlugin.CustomAttributes.Headers, [13](#)
- CorePlugin.CustomAttributes.Validation, [13](#)
- CorePlugin.CustomAttributes.Validation.Base, [14](#)
- CorePlugin.Editor, [14](#)
- CorePlugin.Extensions, [14](#)
- CorePlugin.FileSystem, [14](#)
- CorePlugin.Interface, [15](#)
- CorePlugin.Logger, [15](#)
- CorePlugin.ReferenceDistribution, [15](#)
- CorePlugin.ReferenceDistribution.Interface, [15](#)
- CorePlugin.Samples, [15](#)
- CorePlugin.Samples.Scripts, [15](#)
- CorePlugin.Samples.Scripts.Demo, [15](#)
- CorePlugin.Samples.Scripts.EventTypes, [15](#)
- CorePlugin.Samples.Scripts.Managers, [15](#)
- CorePlugin.Samples.Scripts.Model, [15](#)
- CorePlugin.Serializable, [15](#)
- CorePlugin.Serializable.Interface, [16](#)
- CorePlugin.Singletons, [16](#)
- CorePlugin.UISystem, [16](#)
- CorePlugin.UISystem.UI, [16](#)
- CustomInspectorCoreManager, [21](#)
- CustomLogger, [22](#)
- DeepClone
  - IDeepClone< out out T >, [31](#)
- DelayedOpenPage
  - SubPagesController, [45](#)
- EventInitializer, [22](#)
  - AddHandler, [23](#)
  - InitializeSubscriptions, [23](#)
  - InvokeBase, [24](#)
  - RemoveHandler, [24](#)
  - Subscribe, [24](#)
  - Unsubscribe, [24](#)
- EventTypes, [25](#)
- FieldValidationAttribute, [25](#)
- FileSystem, [26](#)
  - FileSystem, [26](#), [27](#)
  - Load< T >, [27](#)
  - Save< T >, [28](#), [29](#)
- GetData< T >
  - SceneDataHandler, [42](#)
- GetReference< T >
  - ReferenceDistributor, [39](#)
- GetReferences< T >
  - ReferenceDistributor, [39](#)
- GetSubscribers
  - IEventSubscriber, [34](#)
- HasComponentAttribute, [29](#)
- HelpBox
  - UnityEditorExtension, [51](#)
- HideAllTables
  - SubPagesController, [46](#)
- HighlightText

- ConsoleMessage, 19
- IconName
  - UnityEditorExtension, 52
- ICore, 30
  - InitializeElements, 30
- IDeepClone< out out T >, 31
  - DeepClone, 31
- IDistributingReference, 32
- IEventHandler, 32
  - InvokeEvents, 33
  - Subscribe, 33
  - Unsubscribe, 33
- IEventSubscriber, 33
  - GetSubscribers, 34
- Initialize
  - ConsoleMessage, 19
  - ReferenceDistributor, 40
  - UIPage, 48
- InitializeElements
  - BaseCore, 17
  - ICore, 30
  - UIManager, 48
- InitializeSubscriptions
  - EventInitializer, 23
- InvokeBase
  - EventInitializer, 24
- InvokeEvents
  - IEventHandler, 33
- ISceneData, 34
- IsNotNullAndEqual
  - UnityExtensions, 53
- IsNotNullAndNotEqual
  - UnityExtensions, 54
- IUnique, 35
- Load< T >
  - FileSystem, 27
- Named< TName, TKey, TValue >, 35
- NotNullAttribute, 36
- OneAndOnlyAttribute, 36
- OnEnable
  - SymbolDefiner, 47
- OpenPage
  - SubPagesController, 46
- PrefabHeaderAttribute, 37
- PrefabRequiredAttribute, 37
- ReferenceDistributor, 38
  - AskReference< T >, 38
  - GetReference< T >, 39
  - GetReferences< T >, 39
  - Initialize, 40
- ReferencesHeaderAttribute, 40
- RemoveData< T >
  - SceneDataHandler, 42
- RemoveHandler
  - EventInitializer, 24
  - RuntimeConsole, 40
- Save< T >
  - FileSystem, 28, 29
- SceneDataHandler, 41
  - AddData< T >, 41
  - GetData< T >, 42
  - RemoveData< T >, 42
- SetActive
  - ConsoleMessage, 20
- SettingsHeaderAttribute, 43
- ShowSymbolsButtons
  - SymbolDefiner, 47
- Singleton< T >, 43
- StateHeaderAttribute, 43
- SubPagesController, 44
  - AddPage, 44, 45
  - DelayedOpenPage, 45
  - HideAllTables, 46
  - OpenPage, 46
- Subscribe
  - EventInitializer, 24
  - IEventHandler, 33
- SubscribeOnButtonClick
  - ConsoleMessage, 20
- SymbolDefiner, 46
  - OnEnable, 47
  - ShowSymbolsButtons, 47
- TryFindObjectOfType< T >
  - UnityExtensions, 54
- TryFindObjectsOfType< T >
  - UnityExtensions, 55
- UIManager, 47
  - InitializeElements, 48
- UIPage, 48
  - Initialize, 48
- UIStateTools, 49
  - ChangeCursorState, 49
  - ChangeGroupState, 50
- Unique, 50
- UnityEditorExtension, 51
  - HelpBox, 51
  - IconName, 52
- UnityExtensions, 52
  - Clear< T >, 53
  - IsNotNullAndEqual, 53
  - IsNotNullAndNotEqual, 54
  - TryFindObjectOfType< T >, 54
  - TryFindObjectsOfType< T >, 55
- Unsubscribe
  - EventInitializer, 24
  - IEventHandler, 33
- ValidationAttribute, 56
- ValidationAttributeEditor, 56