Core Manager

2.0.2

Generated by Doxygen 1.9.1

1 CoreManager 1

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
1.4.3 v2.0.1 . . . 2
1.4.4 v2.0.2 . . . 2

2 Namespace Index 3

2.1 Namespace List . . . 3

3 Hierarchical Index 5

3.1 Class Hierarchy . . . 5

4 Class Index 7

4.1 Class List . . . 7

5 Namespace Documentation 11

5.1 CorePlugin Namespace Reference . . . 11
5.2 CorePlugin.Attributes Namespace Reference . . . 11
5.3 CorePlugin.Attributes.Base Namespace Reference . . . 11
5.4 CorePlugin.Attributes.Editor Namespace Reference . . . 11
5.5 CorePlugin.Attributes.EditorAddons Namespace Reference . . . 12
5.6 CorePlugin.Attributes.EditorAddons.SelectAttributes Namespace Reference . . . 12
5.7 CorePlugin.Attributes.Headers Namespace Reference . . . 12
5.8 CorePlugin.Attributes.Validation Namespace Reference . . . 12
5.9 CorePlugin.Core Namespace Reference . . . 13
5.10 CorePlugin.Core.Interface Namespace Reference . . . 13
5.11 CorePlugin.Cross Namespace Reference . . . 13
5.12 CorePlugin.Cross.Events Namespace Reference . . . 13
5.13 CorePlugin.Cross.Events.Interface Namespace Reference . . . 13
5.14 CorePlugin.Cross.SceneData Namespace Reference . . . 14
5.15 CorePlugin.Cross.SceneData.Interface Namespace Reference . . . 14
5.16 CorePlugin.Dispatchers Namespace Reference . . . 14
5.17 CorePlugin.Editor Namespace Reference . . . 14
5.18 CorePlugin.Editor.Drawers Namespace Reference . . . 14
5.19 CorePlugin.Editor.Drawers.Decorators Namespace Reference . . . 14
5.20 CorePlugin.Editor.Drawers.SelectDrawers Namespace Reference . . . 14
5.21 CorePlugin.Editor.EditorState Namespace Reference . . . 14
5.22 CorePlugin.Editor.Extensions Namespace Reference . . . 14
5.23 CorePlugin.Editor.Helpers Namespace Reference . . . 15
5.24 CorePlugin.Editor.MenuItemHelpers Namespace Reference . . . 15

5.25 CorePlugin.Editor.Windows Namespace Reference . . . 15
5.26 CorePlugin.Extensions Namespace Reference . . . 15
5.27 CorePlugin.FileSystem Namespace Reference . . . 15
5.28 CorePlugin.Logger Namespace Reference . . . 15
5.29 CorePlugin.ReferenceDistribution Namespace Reference . . . 15
5.30 CorePlugin.ReferenceDistribution.Interface Namespace Reference . . . 16
5.31 CorePlugin.Samples Namespace Reference . . . 16
5.32 CorePlugin.Samples.Scripts Namespace Reference . . . 16
5.33 CorePlugin.Samples.Scripts.Demo Namespace Reference . . . 16
5.34 CorePlugin.Samples.Scripts.EventTypes Namespace Reference . . . 16
5.35 CorePlugin.Samples.Scripts.Managers Namespace Reference . . . 16
5.36 CorePlugin.Samples.Scripts.Model Namespace Reference . . . 16
5.37 CorePlugin.Serializable Namespace Reference . . . 16
5.38 CorePlugin.Serializable.Interface Namespace Reference . . . 16
5.39 CorePlugin.Singletons Namespace Reference . . . 16

6 Class Documentation **17**

6.1 BaseCore Class Reference . . . 17
6.1.1 Detailed Description . . . 17
6.1.2 Member Function Documentation . . . 17
6.1.2.1 InitializeElements() . . . 17
6.2 ClassValidationAttribute Class Reference . . . 18
6.2.1 Detailed Description . . . 18
6.3 CoreManager Class Reference . . . 18
6.3.1 Detailed Description . . . 18
6.4 CoreManagerElementAttribute Class Reference . . . 19
6.4.1 Detailed Description . . . 19
6.5 CoreManagerElementsFieldAttribute Class Reference . . . 19
6.5.1 Detailed Description . . . 19
6.6 DebugLogger Class Reference . . . 20
6.6.1 Detailed Description . . . 20
6.7 EditorButtonAttribute Class Reference . . . 20
6.7.1 Detailed Description . . . 20
6.7.2 Constructor & Destructor Documentation . . . 21
6.7.2.1 EditorButtonAttribute() [1/6] . . . 21
6.7.2.2 EditorButtonAttribute() [2/6] . . . 21
6.7.2.3 EditorButtonAttribute() [3/6] . . . 21
6.7.2.4 EditorButtonAttribute() [4/6] . . . 22
6.7.2.5 EditorButtonAttribute() [5/6] . . . 22
6.7.2.6 EditorButtonAttribute() [6/6] . . . 22
6.8 EventInitializer Class Reference . . . 23
6.8.1 Detailed Description . . . 23

6.8.2 Member Function Documentation . . . 23
6.8.2.1 AddHandler() . . . 23
6.8.2.2 InitializeSubscriptions() . . . 24
6.8.2.3 InvokeBase() . . . 24
6.8.2.4 RemoveHandler() . . . 24
6.8.2.5 Subscribe() . . . 24
6.8.2.6 Unsubscribe() . . . 25
6.9 EventTypes Class Reference . . . 25
6.9.1 Detailed Description . . . 25
6.10 FieldValidationAttribute Class Reference . . . 26
6.10.1 Detailed Description . . . 26
6.11 FileSystem Class Reference . . . 26
6.11.1 Detailed Description . . . 27
6.11.2 Constructor & Destructor Documentation . . . 27
6.11.2.1 FileSystem() [1/3] . . . 27
6.11.2.2 FileSystem() [2/3] . . . 27
6.11.2.3 FileSystem() [3/3] . . . 27
6.11.3 Member Function Documentation . . . 28
6.11.3.1 Load< T >() [1/2] . . . 28
6.11.3.2 Load< T >() [2/2] . . . 28
6.11.3.3 Save< T >() [1/2] . . . 29
6.11.3.4 Save< T >() [2/2] . . . 29
6.12 HasComponentAttribute Class Reference . . . 30
6.12.1 Detailed Description . . . 30
6.13 ICore Interface Reference . . . 30
6.13.1 Detailed Description . . . 31
6.13.2 Member Function Documentation . . . 31
6.13.2.1 InitializeElements() . . . 31
6.14 IDistributingReference Interface Reference . . . 31
6.14.1 Detailed Description . . . 31
6.15 IEventHandler Interface Reference . . . 31
6.15.1 Detailed Description . . . 32
6.15.2 Member Function Documentation . . . 32
6.15.2.1 InvokeEvents() . . . 32
6.15.2.2 Subscribe() . . . 32
6.15.2.3 Unsubscribe() . . . 33
6.16 IEventSubscriber Interface Reference . . . 33
6.16.1 Detailed Description . . . 33
6.16.2 Member Function Documentation . . . 34
6.16.2.1 GetSubscribers() . . . 34
6.17 ISceneData Interface Reference . . . 34
6.17.1 Detailed Description . . . 34

6.18 IUnique Interface Reference . . . 35
6.18.1 Detailed Description . . . 35
6.19 Named< TName, TKey, TValue > Class Template Reference . . . 35
6.19.1 Detailed Description . . . 35
6.20 NotNullAttribute Class Reference . . . 36
6.20.1 Detailed Description . . . 36
6.21 OneAndOnlyAttribute Class Reference . . . 36
6.21.1 Detailed Description . . . 36
6.22 PrefabHeaderAttribute Class Reference . . . 37
6.22.1 Detailed Description . . . 37
6.23 PrefabRequiredAttribute Class Reference . . . 37
6.23.1 Detailed Description . . . 37
6.24 ReferenceDistributor Class Reference . . . 37
6.24.1 Detailed Description . . . 38
6.24.2 Member Function Documentation . . . 38
6.24.2.1 AskReference< T >() . . . 38
6.24.2.2 GetInterfaceReference< T >() . . . 39
6.24.2.3 GetInterfaceReferences< T >() . . . 39
6.24.2.4 GetReference< T >() . . . 40
6.24.2.5 GetReferences< T >() . . . 40
6.24.2.6 Initialize() . . . 40
6.25 ReferencesHeaderAttribute Class Reference . . . 41
6.25.1 Detailed Description . . . 41
6.26 SceneDataHandler Class Reference . . . 41
6.26.1 Detailed Description . . . 42
6.26.2 Member Function Documentation . . . 42
6.26.2.1 AddData< T >() . . . 42
6.26.2.2 GetData< T >() . . . 42
6.26.2.3 RemoveData< T >() [1/2] . . . 43
6.26.2.4 RemoveData< T >() [2/2] . . . 43
6.27 SelectImplementationAttribute Class Reference . . . 44
6.27.1 Detailed Description . . . 44
6.28 SelectTypeAttribute Class Reference . . . 44
6.28.1 Detailed Description . . . 44
6.29 SettingsHeaderAttribute Class Reference . . . 44
6.29.1 Detailed Description . . . 44
6.30 Singleton< T > Class Template Reference . . . 44
6.30.1 Detailed Description . . . 44
6.31 StateHeaderAttribute Class Reference . . . 45
6.31.1 Detailed Description . . . 45
6.32 StaticObjectSingleton< T > Class Template Reference . . . 45
6.32.1 Detailed Description . . . 45

6.33 UIStateTools Class Reference . . . 46
6.33.1 Detailed Description . . . 46
6.33.2 Member Function Documentation . . . 46
6.33.2.1 ChangeCursorState() . . . 46
6.33.2.2 ChangeGroupState() [1/2] . . . 46
6.33.2.3 ChangeGroupState() [2/2] . . . 47
6.34 Unique Class Reference . . . 47
6.34.1 Detailed Description . . . 47
6.35 UnityEditorExtension Class Reference . . . 48
6.35.1 Detailed Description . . . 48
6.35.2 Member Function Documentation . . . 48
6.35.2.1 HelpBox() [1/2] . . . 48
6.35.2.2 HelpBox() [2/2] . . . 48
6.35.2.3 IconName() . . . 49
6.35.2.4 PrettyEditorObjectName() . . . 49
6.36 UnityExtensions Class Reference . . . 50
6.36.1 Detailed Description . . . 50
6.36.2 Member Function Documentation . . . 50
6.36.2.1 Clear< T >() . . . 50
6.36.2.2 CopyToClipboard() . . . 51
6.36.2.3 IsNotNullAndEqual() . . . 51
6.36.2.4 IsNotNullAndNotEqual() . . . 51
6.36.2.5 PrettyObjectName() . . . 52
6.36.2.6 RemoveRange< T >() . . . 52
6.36.2.7 TryToFindObjectOfType< T >() . . . 53
6.36.2.8 TryToFindObjectsOfType< T >() [1/2] . . . 53
6.36.2.9 TryToFindObjectsOfType< T >() [2/2] . . . 54
6.37 ValidationAttribute Class Reference . . . 54
6.37.1 Detailed Description . . . 54
6.38 ValidationAttributeEditor Class Reference . . . 55
6.38.1 Detailed Description . . . 55

Index **57**

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.

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.

1.4.3 v2.0.1

1. Striped part of functions is now available on GitHub repo.
   (a) `Scene Loader` - allows to asynchronously load scene through an intermediate scene and allows to serialize SceneAssets through Inspector (use SceneLoaderAsset).
   (b) `Runtime console` - console with Unity logs for debug and/or release builds. Allows display console logs like in Unity Editor. Strips from release build if other not predetermined.
   (c) `Editor Symbol Definer` - allows defining Scripting Define Symbols in the project thought attribute or button in Inspector.

1.4.4 v2.0.2

1. Fixed build issue with Validation Attributes
2. Removed redundant prefabs and scripts
3. Added:
   (a) SelectImplementation Attribute
   (b) SelectType Attribute

Chapter 2

Namespace Index

2.1 Namespace List

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

CorePlugin . . . 11
CorePlugin.Attributes . . . 11
CorePlugin.Attributes.Base . . . 11
CorePlugin.Attributes.Editor . . . 11
CorePlugin.Attributes.EditorAddons . . . 12
CorePlugin.Attributes.EditorAddons.SelectAttributes . . . 12
CorePlugin.Attributes.Headers . . . 12
CorePlugin.Attributes.Validation . . . 12
CorePlugin.Core . . . 13
CorePlugin.Core.Interface . . . 13
CorePlugin.Cross . . . 13
CorePlugin.Cross.Events . . . 13
CorePlugin.Cross.Events.Interface . . . 13
CorePlugin.Cross.SceneData . . . 14
CorePlugin.Cross.SceneData.Interface . . . 14
CorePlugin.Dispatchers . . . 14
CorePlugin.Editor . . . 14
CorePlugin.Editor.Drawers . . . 14
CorePlugin.Editor.Drawers.Decorators . . . 14
CorePlugin.Editor.Drawers.SelectDrawers . . . 14
CorePlugin.Editor.EditorState . . . 14
CorePlugin.Editor.Extensions . . . 14
CorePlugin.Editor.Helpers . . . 15
CorePlugin.Editor.MenuItemHelpers . . . 15
CorePlugin.Editor.Windows . . . 15
CorePlugin.Extensions . . . 15
CorePlugin.FileSystem . . . 15
CorePlugin.Logger . . . 15
CorePlugin.ReferenceDistribution . . . 15
CorePlugin.ReferenceDistribution.Interface . . . 16
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.Serializable 16
CorePlugin.Serializable.Interface 16
CorePlugin.Singletons 16

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

CoreManager . . . 18
CoreManagerElementAttribute . . . 19
CoreManagerElementsFieldAttribute . . . 19
DebugLogger . . . 20
EditorButtonAttribute . . . 20
EventInitializer . . . 23
EventTypes . . . 25
FileSystem . . . 26
ICore . . . 30
    BaseCore . . . 17
IDistributingReference . . . 31
IEventHandler . . . 31
IEventSubscriber . . . 33
ISceneData . . . 34
IUnique . . . 35
    Unique . . . 47
Named< TName, TKey, TValue > . . . 35
Named< Object, string > . . . 35
Named< string, Object > . . . 35
PrefabHeaderAttribute . . . 37
ReferenceDistributor . . . 37
ReferencesHeaderAttribute . . . 41
SelectImplementationAttribute . . . 44
SelectTypeAttribute . . . 44
SettingsHeaderAttribute . . . 44
Singleton< T > . . . 44
StateHeaderAttribute . . . 45
StaticObjectSingleton< T > . . . 45
StaticObjectSingleton< MainThreadDispatcher > . . . 45
StaticObjectSingleton< SceneDataHandler > . . . 45
    SceneDataHandler . . . 41
StaticObjectSingleton< StaticCoroutineDispatcher > . . . 45
UIStateTools . . . 46
UnityEditorExtension . . . 48

UnityExtensions 50
ValidationAttribute 54
ClassValidationAttribute 18
OneAndOnlyAttribute 36
FieldValidationAttribute 26
HasComponentAttribute 30
NotNullAttribute 36
PrefabRequiredAttribute 37
ValidationAttributeEditor 55

Chapter 4

Class Index

4.1 Class List

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

BaseCore
Base implementation of IManager 17
ClassValidationAttribute
Base attribute for class validation 18

18

19

19

DebugLogger
Custom logger solution for logs 20

EditorButtonAttribute
Displays Button in Inspector 20

23

EventTypes
Declare your delegate type to convert method group to delegate 25

FieldValidationAttribute
Base attribute for field validation 26

26

HasComponentAttribute
Attribute validating whether Object in field or all objects in the list have desired component 30

ICore
Interface for Manager declaration 30

IDistributingReference
Flag interface for MonoBehaviour classes what need to referenced from another class 31

IEventHandler
Interface for event handler 31

IEventSubscriber
Interface for subscribers 33

34

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 35

Named< TName, TKey, TValue >
Replacement for dictionary in Unity Inspector 35

NotNullAttribute
Attribute validating whether field or element in the list equals null 36

OneAndOnlyAttribute
Attribute validating whether there is only one copy of this class in the scene 36

PrefabHeaderAttribute
Replacement for Header("Prefabs") 37

PrefabRequiredAttribute
Attribute validating whether the object or all items in list are prefabs 37

ReferenceDistributor
Class responsible for reference distribution inside one scene 37

ReferencesHeaderAttribute
Replacement for Header("References") 41

41

SelectImplementationAttribute
Attribute for Implementation selection in Inspector. Use in pair with [SerializeReference] Attribute . 44

SelectTypeAttribute
Attribute for Type selection in Inspector. Use in pair with [SerializeReference] Attribute 44

SettingsHeaderAttribute
Replacement for Header("Settings") 44

Singleton< T >
Base for all singletons. Strongly recommended to use singletons as little as possible 44

StateHeaderAttribute
Replacement for Header("State") 45

StaticObjectSingleton< T >
Base for static objects singletons. Strongly recommended to use singletons as little as possible . . 45

UIStateTools
UI state tool for canvas groups 46

Unique
Base class for classes that need to be saved in Json file 47

UnityEditorExtension
Extensions for Unity Editor classes 48

UnityExtensions
Extension class for default Unity classes 50

ValidationAttribute
Base attribute for validation 54

55

Chapter 5

Namespace Documentation

5.1 CorePlugin Namespace Reference

5.2 CorePlugin.Attributes Namespace Reference

5.3 CorePlugin.Attributes.Base Namespace Reference

Classes

- class ClassValidationAttribute

  Base attribute for class validation.
- class FieldValidationAttribute

  Base attribute for field validation.
- class ValidationAttribute

  Base attribute for validation

5.4 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

  CorePlugin.Attributes.Base.ValidationAttribute, CorePlugin.Attributes.Base.FieldValidationAttribute, CorePlugin.Attributes.Base.Cla

5.5 CorePlugin.Attributes.EditorAddons Namespace Reference

Classes

- class CoreManagerElementAttribute

  Provide component marked by this attribute to
  See also

  Editor.Windows.CoreSelectorWindow
- class CoreManagerElementsFieldAttribute

  Provide data from object field marked by this attribute to
  See also

  Editor.Windows.CoreSelectorWindow
- class EditorButtonAttribute

  Displays Button in Inspector

5.6 CorePlugin.Attributes.EditorAddons.SelectAttributes Namespace Reference

Classes

- class SelectImplementationAttribute

  Attribute for Implementation selection in Inspector. Use in pair with [SerializeReference] Attribute.
- class SelectTypeAttribute

  Attribute for Type selection in Inspector. Use in pair with [SerializeReference] Attribute.

5.7 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")

5.8 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.

5.9 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

5.10 CorePlugin.Core.Interface Namespace Reference

Classes

- interface ICore

  Interface for Manager declaration.

5.11 CorePlugin.Cross Namespace Reference

5.12 CorePlugin.Cross.Events Namespace Reference

Classes

- class EventTypes

  Declare your delegate type to convert method group to delegate.

5.13 CorePlugin.Cross.Events.Interface Namespace Reference

Classes

- interface IEventHandler

  Interface for event handler.
- interface IEventSubscriber

  Interface for subscribers.

5.14 CorePlugin.Cross.SceneData Namespace Reference

Classes

- class SceneDataHandler

  Singleton for passing data between scenes
  See also

  CorePlugin.Cross.SceneData.Interface.ISceneData

5.15 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

5.16 CorePlugin.Dispatchers Namespace Reference

5.17 CorePlugin.Editor Namespace Reference

5.18 CorePlugin.Editor.Drawers Namespace Reference

5.19 CorePlugin.Editor.Drawers.Decorators Namespace Reference

5.20 CorePlugin.Editor.Drawers.SelectDrawers Namespace Reference

5.21 CorePlugin.Editor.EditorState Namespace Reference

5.22 CorePlugin.Editor.Extensions Namespace Reference

Classes

- class UnityEditorExtension

  Extensions for Unity Editor classes

5.23 CorePlugin.Editor.Helpers Namespace Reference

5.24 CorePlugin.Editor.MenuItemHelpers Namespace Reference

5.25 CorePlugin.Editor.Windows Namespace Reference

5.26 CorePlugin.Extensions Namespace Reference

Classes

- class Named

  Replacement for dictionary in Unity Inspector
- class UIStateTools

  UI state tool for canvas groups
- class UnityExtensions

  Extension class for default Unity classes

5.27 CorePlugin.FileSystem Namespace Reference

Classes

- class FileSystem

  Class for saving Json file to disk.
  See also

  CorePlugin.Serializable.Unique

5.28 CorePlugin.Logger Namespace Reference

Classes

- class DebugLogger

  Custom logger solution for logs.

5.29 CorePlugin.ReferenceDistribution Namespace Reference

Classes

- class ReferenceDistributor

  Class responsible for reference distribution inside one scene.

5.30 CorePlugin.ReferenceDistribution.Interface Namespace Reference

Classes

- interface IDistributingReference

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

5.31 CorePlugin.Samples Namespace Reference

5.32 CorePlugin.Samples.Scripts Namespace Reference

5.33 CorePlugin.Samples.Scripts.Demo Namespace Reference

5.34 CorePlugin.Samples.Scripts.EventTypes Namespace Reference

5.35 CorePlugin.Samples.Scripts.Managers Namespace Reference

5.36 CorePlugin.Samples.Scripts.Model Namespace Reference

5.37 CorePlugin.Serializable Namespace Reference

Classes

- class Unique

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

5.38 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

5.39 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.

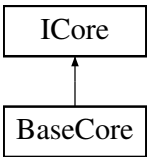
Chapter 6

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.

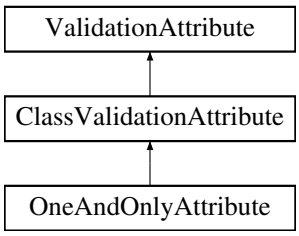
The documentation for this class was generated from the following file:

- 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 CoreManager Class Reference

Manager for initialization of sub manager in the scene.

See also

CorePlugin.ReferenceDistribution.ReferenceDistributor, CorePlugin.Core.Interface.ICore

Inherits MonoBehaviour.

6.3.1 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

6.4 CoreManagerElementAttribute Class Reference

Provide component marked by this attribute to

See also

Editor.Windows.CoreSelectorWindow

Inherits DisplayNameAttribute.

6.4.1 Detailed Description

Provide component marked by this attribute to

See also

Editor.Windows.CoreSelectorWindow

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

- CoreManagerElementAttribute.cs

6.5 CoreManagerElementsFieldAttribute Class Reference

Provide data from object field marked by this attribute to

See also

Editor.Windows.CoreSelectorWindow

Inherits Attribute.

6.5.1 Detailed Description

Provide data from object field marked by this attribute to

See also

Editor.Windows.CoreSelectorWindow

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

- CoreManagerElementAttribute.cs

6.6 DebugLogger Class Reference

Custom logger solution for logs.

6.6.1 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". It's fully stripped from Release Builds.

See also

CorePlugin.Core.CoreManager

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

- DebugLogger.cs

6.7 EditorButtonAttribute Class Reference

Displays Button in Inspector

Inherits DisplayNameAttribute.

Public Member Functions

- EditorButtonAttribute (string displayName, params object[ ] invokeParams)
  Provides Editor button
- EditorButtonAttribute (string displayName, int captureGroup, int priority, params object[ ] invokeParams)
  Provides Editor button
- EditorButtonAttribute (params object[ ] invokeParams)
  Provides Editor button
- EditorButtonAttribute (int captureGroup, params object[ ] invokeParams)
  Provides Editor button
- EditorButtonAttribute (string displayName, int captureGroup, params object[ ] invokeParams)
  Provides Editor button
- EditorButtonAttribute (int captureGroup, int priority, params object[ ] invokeParams)
  Provides Editor button

6.7.1 Detailed Description

Displays Button in Inspector

6.7.2 Constructor & Destructor Documentation

6.7.2.1 EditorButtonAttribute() [1/6]

```
EditorButtonAttribute (
            string displayName,
            params object[] invokeParams )  [inline]
```

Provides Editor button

Parameters

displayName	
invokeParams	

6.7.2.2 EditorButtonAttribute() [2/6]

```
EditorButtonAttribute (
            string displayName,
            int captureGroup,
            int priority,
            params object[] invokeParams )  [inline]
```

Provides Editor button

Parameters

displayName	
captureGroup	
priority	
invokeParams	

6.7.2.3 EditorButtonAttribute() [3/6]

```
EditorButtonAttribute (
            params object[] invokeParams )  [inline]
```

Provides Editor button

Parameters

invokeParams	

6.7.2.4 EditorButtonAttribute() [4/6]

```
EditorButtonAttribute (
            int captureGroup,
            params object[] invokeParams )  [inline]
```

Provides Editor button

Parameters

captureGroup	
invokeParams	

6.7.2.5 EditorButtonAttribute() [5/6]

```
EditorButtonAttribute (
            string displayName,
            int captureGroup,
            params object[] invokeParams )  [inline]
```

Provides Editor button

Parameters

displayName	
captureGroup	
invokeParams	

6.7.2.6 EditorButtonAttribute() [6/6]

```
EditorButtonAttribute (
            int captureGroup,
            int priority,
            params object[] invokeParams )  [inline]
```

Provides Editor button

Parameters

captureGroup	
priority	
invokeParams	

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

- EditorButtonAttribute.cs

6.8 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.

6.8.1 Detailed Description

Class responsible for reference initialization

See also

CorePlugin.Cross.Events.Interface.IEventHandler, CorePlugin.Cross.Events.Interface.IEventSubscriber

6.8.2 Member Function Documentation

6.8.2.1 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

handler	
subscriptionsNeeded	If false invokeNeeded will not be called
invokeNeeded	

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

6.8.2.2 InitializeSubscriptions()

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

Initialising cross subscriptions for all handlers in the scene.

6.8.2.3 InvokeBase()

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

Invoking event on handlers.

6.8.2.4 RemoveHandler()

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

Removing event handler after scene has Awoken from list

Parameters

handler	

6.8.2.5 Subscribe()

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

Subscribing event subscriber after scene has Awoken to event handlers.

Parameters

subscriber	

6.8.2.6 Unsubscribe()

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

Unsubscribing event subscriber after scene has Awoken from event handlers.

Parameters

subscriber	

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

- EventInitializer.cs

6.9 EventTypes Class Reference

Declare your delegate type to convert method group to delegate.

Inherited by CustomEventTypes.

6.9.1 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

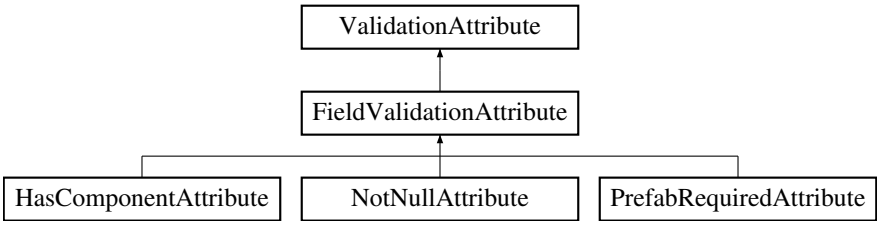
The documentation for this class was generated from the following file:

- EventTypes.cs

6.10 FieldValidationAttribute Class Reference

Base attribute for field validation.

Inheritance diagram for FieldValidationAttribute:



6.10.1 Detailed Description

Base attribute for field validation.

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

- FieldValidationAttribute.cs

6.11 FileSystem Class Reference

Class for saving Json file to disk.

See also

CorePlugin.Serializable.Unique

Inherits IDisposable.

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.

6.11.1 Detailed Description

Class for saving Json file to disk.

See also

CorePlugin.Serializable.Unique

6.11.2 Constructor & Destructor Documentation

6.11.2.1 FileSystem() [1/3]

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

Default constructor.

6.11.2.2 FileSystem() [2/3]

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

Overloaded constructor which changes default path.

Parameters

path	

6.11.2.3 FileSystem() [3/3]

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

Overloaded constructor which changes default path and file extension.

Parameters

path	
extension	

6.11.3 Member Function Documentation

6.11.3.1 Load< T >() [1/2]

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

Loads file.

Parameters

onLoaded	
onError	
context	Required if errors should be shown on Object

Template Parameters

T	

Type Constraints

T* : *Unique

6.11.3.2 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

fileName	
onLoaded	
onError	
context	Required if errors should be shown on Object

Template Parameters

T	

Type Constraints

T **:** ***Unique***

6.11.3.3 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

data	
onError	
context	Required if errors should be shown on Object

Template Parameters

T	

Type Constraints

T **:** ***Unique***

6.11.3.4 Save< T >() [2/2]

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

Saves your class to file.

Parameters

data	
fileName	
onError	
context	Required if errors should be shown on Object

Template Parameters

T	

Type Constraints

T* : *Unique

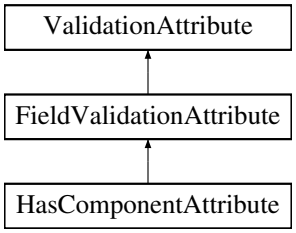
The documentation for this class was generated from the following file:

- FileSystem.cs

6.12 HasComponentAttribute Class Reference

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

Inheritance diagram for HasComponentAttribute:



6.12.1 Detailed Description

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

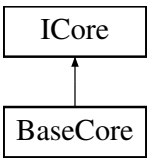
The documentation for this class was generated from the following file:

- HasComponentAttribute.cs

6.13 ICore Interface Reference

Interface for Manager declaration.

Inheritance diagram for ICore:



Public Member Functions

- void InitializeElements ()

  Initializing all managers elements.

6.13.1 Detailed Description

Interface for Manager declaration.

6.13.2 Member Function Documentation

6.13.2.1 InitializeElements()

```
void InitializeElements ( )
```

Initializing all managers elements.

Implemented in BaseCore.

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

- ICore.cs

6.14 IDistributingReference Interface Reference

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

6.14.1 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

6.15 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

6.15.1 Detailed Description

Interface for event handler.

```
public void Subscribe(params Delegate[ ] subscriber)
{
EventExtensions.Subscribe(ref event, subscribers);
}
```

See also

CorePlugin.Cross.Events.EventTypes

6.15.2 Member Function Documentation

6.15.2.1 InvokeEvents()

```
void InvokeEvents ( )
```

Invoking events that need to be invoked on scene initialization.

Referenced by EventInitializer.AddHandler().

6.15.2.2 Subscribe()

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

Subscribing delegates to event

Parameters

subscribers	

Referenced by EventInitializer.AddHandler().

6.15.2.3 Unsubscribe()

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

Unsubscribing delegates to event

Parameters

unsubscribers	

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

- IEventHandler.cs

6.16 IEventSubscriber Interface Reference

Interface for subscribers.

Inherited by SceneSwitcher.

Public Member Functions

- Delegate[] GetSubscribers ()

  Returns IEnumerable with all methods which need to be subscribed.

6.16.1 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

6.16.2 Member Function Documentation

6.16.2.1 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

6.17 ISceneData Interface Reference

Flag interface for classes or structs to pass between scenes

See also

CorePlugin.Cross.SceneData.SceneDataHandler

Inherited by DataTransfer.

6.17.1 Detailed Description

Flag interface for classes or structs to pass between scenes

See also

CorePlugin.Cross.SceneData.SceneDataHandler

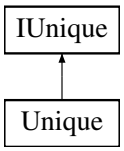
The documentation for this interface was generated from the following file:

- ISceneData.cs

6.18 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:



6.18.1 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

6.19 Named< TName, TKey, TValue > Class Template Reference

Replacement for dictionary in Unity Inspector

6.19.1 Detailed Description

Replacement for dictionary in Unity Inspector

More complex list for dictionary

Template Parameters

TKey	
TValue	

Template Parameters

TName	
TKey	
TValue	

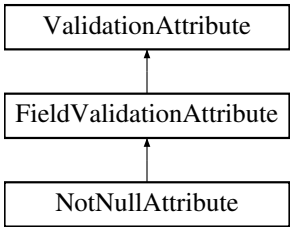
The documentation for this class was generated from the following file:

- Named.cs

6.20 NotNullAttribute Class Reference

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

Inheritance diagram for NotNullAttribute:



6.20.1 Detailed Description

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

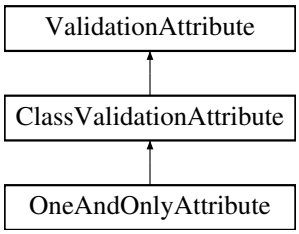
The documentation for this class was generated from the following file:

- NotNullAttribute.cs

6.21 OneAndOnlyAttribute Class Reference

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

Inheritance diagram for OneAndOnlyAttribute:



6.21.1 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

6.22 PrefabHeaderAttribute Class Reference

Replacement for Header("Prefabs")

Inherits HeaderAttribute.

6.22.1 Detailed Description

Replacement for Header("Prefabs")

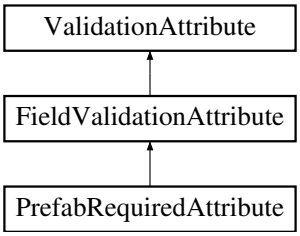
The documentation for this class was generated from the following file:

- PrefabHeaderAttribute.cs

6.23 PrefabRequiredAttribute Class Reference

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

Inheritance diagram for PrefabRequiredAttribute:



6.23.1 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

6.24 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 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
- 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 T GetReference< T > ([CallerMemberName] string callerName="")

  Getting reference by type from list
- static IEnumerable< T > GetReferences< T > ([CallerMemberName] string callerName="")

  Getting references by type from list

6.24.1 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.

6.24.2 Member Function Documentation

6.24.2.1 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

reference	
callerName	

Template Parameters

T	

Returns

Type Constraints

T* : *MonoBehaviour

T* : *IDistributingReference

References ReferenceDistributor.GetReference< T >().

6.24.2.2 GetInterfaceReference< T >()

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

Getting reference by type from list

Template Parameters

T	

Returns

Type Constraints

T* : *IDistributingReference

6.24.2.3 GetInterfaceReferences< T >()

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

Getting reference by type from list

Template Parameters

T	

Returns

Type Constraints

T* : *IDistributingReference

6.24.2.4 GetReference< T >()

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

Getting reference by type from list

Template Parameters

T	

Returns

Type Constraints

T* : *MonoBehaviour
T* : *IDistributingReference

Referenced by ReferenceDistributor.AskReference< T >().

6.24.2.5 GetReferences< T >()

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

Getting references by type from list

Template Parameters

T	

Returns

Type Constraints

T* : *MonoBehaviour
T* : *IDistributingReference

6.24.2.6 Initialize()

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

Initializing distribution references

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

- ReferenceDistributor.cs

6.25 ReferencesHeaderAttribute Class Reference

Replacement for Header("References")

Inherits HeaderAttribute.

6.25.1 Detailed Description

Replacement for Header("References")

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

- ReferencesHeaderAttribute.cs

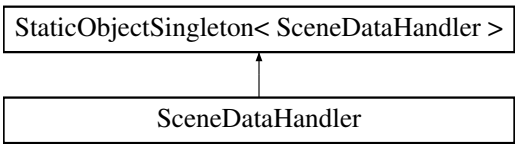
6.26 SceneDataHandler Class Reference

Singleton for passing data between scenes

See also

CorePlugin.Cross.SceneData.Interface.ISceneData

Inheritance diagram for SceneDataHandler:



Public Member Functions

- void RemoveData< T > (T data)

  Removing data from dictionary by passed Type

Static Public Member Functions

- static void AddData< T > (T data)

  Adding data to dictionary by passed Type
- static bool GetData< T > (out T data)

  Getting data from dictionary by passed Type
- static void RemoveData< T > ()

  Removing data from dictionary by passed Type

6.26.1 Detailed Description

Singleton for passing data between scenes

See also

CorePlugin.Cross.SceneData.Interface.ISceneData

6.26.2 Member Function Documentation

6.26.2.1 AddData< T >()

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

Adding data to dictionary by passed Type

Parameters

data	

Template Parameters

T	

Type Constraints

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

6.26.2.2 GetData< T >()

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

Getting data from dictionary by passed Type

Parameters

data	

Template Parameters

T	

Type Constraints

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

6.26.2.3 RemoveData< T >() [1/2]

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

Removing data from dictionary by passed Type

Template Parameters

T	

Type Constraints

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

6.26.2.4 RemoveData< T >() [2/2]

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

Removing data from dictionary by passed Type

Template Parameters

T	

Type Constraints

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

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

- SceneDataHandler.cs

6.27 SelectImplementationAttribute Class Reference

Attribute for Implementation selection in Inspector. Use in pair with [SerializeReference] Attribute.

Inherits SelectAttributeBase.

6.27.1 Detailed Description

Attribute for Implementation selection in Inspector. Use in pair with [SerializeReference] Attribute.

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

- SelectImplementationAttribute.cs

6.28 SelectTypeAttribute Class Reference

Attribute for Type selection in Inspector. Use in pair with [SerializeReference] Attribute.

Inherits SelectAttributeBase.

6.28.1 Detailed Description

Attribute for Type selection in Inspector. Use in pair with [SerializeReference] Attribute.

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

- SelectTypeAttribute.cs

6.29 SettingsHeaderAttribute Class Reference

Replacement for Header("Settings")

Inherits HeaderAttribute.

6.29.1 Detailed Description

Replacement for Header("Settings")

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

- SettingsHeaderAttribute.cs

6.30 Singleton< T > Class Template Reference

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

Inherits MonoBehaviour.

6.30.1 Detailed Description

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

Template Parameters

T	

Type Constraints

T* : *MonoBehaviour

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

- Singleton.cs

6.31 StateHeaderAttribute Class Reference

Replacement for Header("State")

Inherits HeaderAttribute.

6.31.1 Detailed Description

Replacement for Header("State")

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

- StateHeaderAttribute.cs

6.32 StaticObjectSingleton< T > Class Template Reference

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

Inherits MonoBehaviour.

6.32.1 Detailed Description

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

Template Parameters

T	

Type Constraints

T* : *MonoBehaviour

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

- StaticObjectSingleton.cs

6.33 UIStateTools Class Reference

UI state tool for canvas groups

Static Public Member Functions

- static void ChangeGroupState (this 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

6.33.1 Detailed Description

UI state tool for canvas groups

6.33.2 Member Function Documentation

6.33.2.1 ChangeCursorState()

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

Changing state of mouse cursor

Parameters

state	

6.33.2.2 ChangeGroupState() [1/2]

```
static IEnumerator ChangeGroupState (
            CanvasGroup group,
```

```
            bool isVisible,
            float delay )  [inline], [static]
```

Changing canvas visibility and interactivity after delay

Parameters

group	
isVisible	
delay	

6.33.2.3 ChangeGroupState() [2/2]

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

Changing canvas visibility and interactivity

Parameters

group	
isVisible	

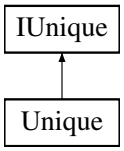
The documentation for this class was generated from the following file:

- UIStateTools.cs

6.34 Unique Class Reference

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

Inheritance diagram for Unique:



6.34.1 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

6.35 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 string PrettyEditorObjectName (this Object input, params string[] remove)

  Gets object name and converts into CamelCase. If input has attribute DisplayNameAttribute then displays name from attribute
- 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

6.35.1 Detailed Description

Extensions for Unity Editor classes

6.35.2 Member Function Documentation

6.35.2.1 HelpBox() [1/2]

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

Override for default Inspector HelpBox with RTF text

Parameters

message	
type	

6.35.2.2 HelpBox() [2/2]

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

Override for default Inspector HelpBox with style

Parameters

message	
type	
style	

References UnityEditorExtension.IconName().

6.35.2.3 IconName()

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

Getting Icon Name from Unity Inspector

Parameters

type	

Returns

Referenced by UnityEditorExtension.HelpBox().

6.35.2.4 PrettyEditorObjectName()

```
static string PrettyEditorObjectName (
            this Object input,
            params string[] remove )  [inline], [static]
```

Gets object name and converts into CamelCase. If input has attribute DisplayNameAttribute then displays name from attribute

Parameters

input	
remove	

Returns

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

- UnityEditorExtension.cs

6.36 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 string PrettyObjectName (this Object input, params string[] remove)

  Gets object name and converts into CamelCase
- static List< T > RemoveRange< T > (this List< T > list, IEnumerable< T > enumerable)

  Removing range of items from list
- 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 TryToFindObjectOfType< T > (out T result)

  Trying to find object on scene that was inherited from T
- static bool TryToFindObjectsOfType< T > (out IEnumerable< T > result)

  Trying to find objects on scene that were inherited from T
- static bool TryToFindObjectsOfType< T > (out IList< T > result)

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

6.36.1 Detailed Description

Extension class for default Unity classes

6.36.2 Member Function Documentation

6.36.2.1 Clear< T >()

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

Clearing list and destroying its items

Parameters

list	

Template Parameters

T	

Type Constraints

T* : *MonoBehaviour

6.36.2.2 CopyToClipboard()

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

Puts the string into the Clipboard.

6.36.2.3 IsNotNullAndEqual()

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

Checks whether the left item is null and equals right item

Parameters

lci	Left compare item
rci	Right compare item

Returns

6.36.2.4 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

lci	Left compare item
rci	Right compare item

Returns

6.36.2.5 PrettyObjectName()

```
static string PrettyObjectName (
            this Object input,
            params string[] remove )  [inline], [static]
```

Gets object name and converts into CamelCase

Parameters

input	
remove	

Returns

6.36.2.6 RemoveRange< T >()

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

Removing range of items from list

Parameters

list	
enumerable	

Template Parameters

T	

Returns

6.36.2.7 TryToFindObjectOfType< T >()

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

Trying to find object on scene that was inherited from T

Parameters

result	

Template Parameters

T	

Returns

6.36.2.8 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

result	

Template Parameters

T	

Returns

6.36.2.9 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

result	

Template Parameters

T	

Returns

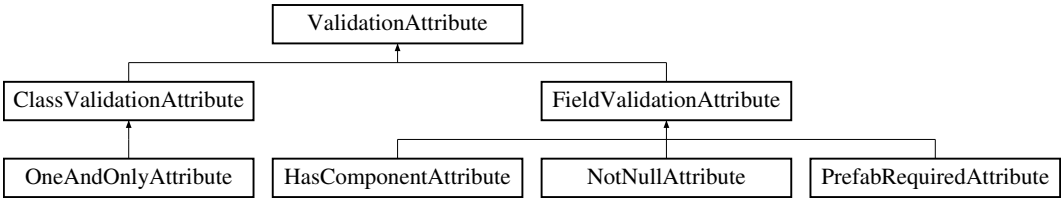
The documentation for this class was generated from the following file:

- UnityExtensions.cs

6.37 ValidationAttribute Class Reference

Base attribute for validation

Inheritance diagram for ValidationAttribute:



6.37.1 Detailed Description

Base attribute for validation

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

- ValidationAttribute.cs

6.38 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.Attributes.Base.ValidationAttribute, CorePlugin.Attributes.Base.FieldValidationAttribute, CorePlugin.Attributes.Base.

Inherits Editor.

Inherited by CoreManagerEditor.

6.38.1 Detailed Description

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

See also

CorePlugin.Attributes.Base.ValidationAttribute, CorePlugin.Attributes.Base.FieldValidationAttribute, CorePlugin.Attributes.Base.

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

- ValidationAttributeEditor.cs

Index

AddData< T >
    SceneDataHandler, 42
AddHandler
    EventInitializer, 23
AskReference< T >
    ReferenceDistributor, 38

BaseCore, 17
    InitializeElements, 17

ChangeCursorState
    UIStateTools, 46
ChangeGroupState
    UIStateTools, 46, 47
ClassValidationAttribute, 18
Clear< T >
    UnityExtensions, 50
CopyToClipboard
    UnityExtensions, 51
CoreManager, 18
CoreManagerElementAttribute, 19
CoreManagerElementsFieldAttribute, 19
CorePlugin, 11
CorePlugin.Attributes, 11
CorePlugin.Attributes.Base, 11
CorePlugin.Attributes.Editor, 11
CorePlugin.Attributes.EditorAddons, 12
CorePlugin.Attributes.EditorAddons.SelectAttributes, 12
CorePlugin.Attributes.Headers, 12
CorePlugin.Attributes.Validation, 12
CorePlugin.Core, 13
CorePlugin.Core.Interface, 13
CorePlugin.Cross, 13
CorePlugin.Cross.Events, 13
CorePlugin.Cross.Events.Interface, 13
CorePlugin.Cross.SceneData, 14
CorePlugin.Cross.SceneData.Interface, 14
CorePlugin.Dispatchers, 14
CorePlugin.Editor, 14
CorePlugin.Editor.Drawers, 14
CorePlugin.Editor.Drawers.Decorators, 14
CorePlugin.Editor.Drawers.SelectDrawers, 14
CorePlugin.Editor.EditorState, 14
CorePlugin.Editor.Extensions, 14
CorePlugin.Editor.Helpers, 15
CorePlugin.Editor.MenuItemHelpers, 15
CorePlugin.Editor.Windows, 15
CorePlugin.Extensions, 15
CorePlugin.FileSystem, 15
CorePlugin.Logger, 15
CorePlugin.ReferenceDistribution, 15
CorePlugin.ReferenceDistribution.Interface, 16
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.Serializable, 16
CorePlugin.Serializable.Interface, 16
CorePlugin.Singletons, 16

DebugLogger, 20

EditorButtonAttribute, 20
    EditorButtonAttribute, 21, 22
EventInitializer, 23
    AddHandler, 23
    InitializeSubscriptions, 24
    InvokeBase, 24
    RemoveHandler, 24
    Subscribe, 24
    Unsubscribe, 25
EventTypes, 25

FieldValidationAttribute, 26
FileSystem, 26
    FileSystem, 27
    Load< T >, 28
    Save< T >, 29

GetData< T >
    SceneDataHandler, 42
GetInterfaceReference< T >
    ReferenceDistributor, 39
GetInterfaceReferences< T >
    ReferenceDistributor, 39
GetReference< T >
    ReferenceDistributor, 39
GetReferences< T >
    ReferenceDistributor, 40
GetSubscribers
    IEventSubscriber, 34

HasComponentAttribute, 30
HelpBox
    UnityEditorExtension, 48

IconName
    UnityEditorExtension, 49
ICore, 30

InitializeElements, 31
IDistributingReference, 31
IEventHandler, 31
InvokeEvents, 32
Subscribe, 32
Unsubscribe, 33
IEventSubscriber, 33
GetSubscribers, 34
Initialize
ReferenceDistributor, 40
InitializeElements
BaseCore, 17
ICore, 31
InitializeSubscriptions
EventInitializer, 24
InvokeBase
EventInitializer, 24
InvokeEvents
IEventHandler, 32
ISceneData, 34
IsNotNullAndEqual
UnityExtensions, 51
IsNotNullAndNotEqual
UnityExtensions, 51
IUnique, 35

Load< T >
FileSystem, 28

Named< TName, TKey, TValue >, 35
NotNullAttribute, 36

OneAndOnlyAttribute, 36

PrefabHeaderAttribute, 37
PrefabRequiredAttribute, 37
PrettyEditorObjectName
UnityEditorExtension, 49
PrettyObjectName
UnityExtensions, 52

ReferenceDistributor, 37
AskReference< T >, 38
GetInterfaceReference< T >, 39
GetInterfaceReferences< T >, 39
GetReference< T >, 39
GetReferences< T >, 40
Initialize, 40
ReferencesHeaderAttribute, 41
RemoveData< T >
SceneDataHandler, 43
RemoveHandler
EventInitializer, 24
RemoveRange< T >
UnityExtensions, 52

Save< T >
FileSystem, 29
SceneDataHandler, 41
AddData< T >, 42
GetData< T >, 42
RemoveData< T >, 43
SelectImplementationAttribute, 44
SelectTypeAttribute, 44
SettingsHeaderAttribute, 44
Singleton< T >, 44
StateHeaderAttribute, 45
StaticObjectSingleton< T >, 45
Subscribe
EventInitializer, 24
IEventHandler, 32

TryToFindObjectOfType< T >
UnityExtensions, 53
TryToFindObjectsOfType< T >
UnityExtensions, 53

UIStateTools, 46
ChangeCursorState, 46
ChangeGroupState, 46, 47
Unique, 47
UnityEditorExtension, 48
HelpBox, 48
IconName, 49
PrettyEditorObjectName, 49
UnityExtensions, 50
Clear< T >, 50
CopyToClipboard, 51
IsNotNullAndEqual, 51
IsNotNullAndNotEqual, 51
PrettyObjectName, 52
RemoveRange< T >, 52
TryToFindObjectOfType< T >, 53
TryToFindObjectsOfType< T >, 53
Unsubscribe
EventInitializer, 25
IEventHandler, 33

ValidationAttribute, 54
ValidationAttributeEditor, 55