Advanced Scene Manager

Generated by Doxygen 1.10.0

1 Hierarchical Index	1
1.1 Class Hierarchy	. 1
2 Class Index	5
2.1 Class List	. 5
3 Namespace Documentation	9
3.1 AdvancedSceneManager Namespace Reference	. 9
3.2 AdvancedSceneManager.Callbacks Namespace Reference	. 9
3.3 AdvancedSceneManager.Core Namespace Reference	. 10
3.3.1 Enumeration Type Documentation	. 10
3.3.1.1 Phase	. 10
3.4 AdvancedSceneManager.DependencyInjection Namespace Reference	. 11
3.5 AdvancedSceneManager.DependencyInjection.Editor Namespace Reference	. 11
3.6 AdvancedSceneManager.Models Namespace Reference	. 11
3.6.1 Enumeration Type Documentation	. 12
3.6.1.1 InputBindingInteractionType	. 12
3.7 AdvancedSceneManager.Models.Enums Namespace Reference	. 12
3.7.1 Enumeration Type Documentation	. 13
3.7.1.1 CollectionLoadingThreadPriority	. 13
3.7.1.2 CollectionStartupOption	. 13
3.7.1.3 EditorPersistentOption	. 14
3.7.1.4 LoadingScreenUsage	. 14
3.7.1.5 SceneImportOption	. 14
3.7.1.6 SceneState	. 14
3.8 AdvancedSceneManager.Models.Helpers Namespace Reference	. 15
3.9 AdvancedSceneManager.Models.Internal Namespace Reference	. 15
3.10 AdvancedSceneManager.Models.Utility Namespace Reference	. 15
3.11 AdvancedSceneManager.Utility Namespace Reference	. 16
3.12 AdvancedSceneManager.Utility.CrossSceneReferences Namespace Reference	. 17
3.12.1 Class Documentation	. 17
3.12.1.1 class AdvancedSceneManager::Utility::CrossSceneReferences::SceneReference	
3.12.2 Enumeration Type Documentation	. 17
3.12.2.1 SceneStatus	
3.13 Lazy Namespace Reference	. 18
3.14 Lazy.Utility Namespace Reference	
4 Class Documentation	19
4.1 ActionUtility	
4.1.1 Detailed Description	
4.1.2 Member Function Documentation	
4.1.2.1 Trylnvoke()	

4.2 App	20
4.2.1 Detailed Description	21
4.2.2 Member Function Documentation	21
4.2.2.1 CancelQuit()	21
4.2.2.2 Exit()	21
4.2.2.3 Quit()	21
4.3 ASMFilePathAttribute	22
4.3.1 Detailed Description	22
4.4 ASMModel	22
4.4.1 Detailed Description	23
4.4.2 Member Function Documentation	23
4.4.2.1 Save()	23
4.4.2.2 SaveNow() [1/2]	23
4.4.2.3 SaveNow() [2/2]	23
4.5 ASMModelExtensions	23
4.5.1 Detailed Description	24
4.5.2 Member Function Documentation	24
4.5.2.1 CloseAll()	24
4.5.2.2 IndexOf< T $>$ ()	24
4.5.2.3 OpenAdditive() [1/2]	25
4.5.2.4 OpenAdditive() [2/2]	25
4.5.2.5 OpenAll()	25
4.6 ASMSceneHelper	25
4.6.1 Detailed Description	27
4.6.2 Member Function Documentation	27
4.6.2.1 Close()	27
4.6.2.2 FinishPreload()	27
4.6.2.3 Open()	27
4.6.2.4 Preload()	27
4.7 ASMScriptableSingleton < T >	28
4.7.1 Detailed Description	28
4.7.2 Member Function Documentation	28
4.7.2.1 Save()	28
4.7.2.2 SaveNow()	28
4.8 ASMSettings	29
4.8.1 Detailed Description	30
4.8.2 Property Documentation	30
4.8.2.1 allowExcludingCollectionsFromBuild	30
4.8.2.2 enableCrossSceneReferences	31
4.9 Assets	31
4.9.1 Detailed Description	31
4.9.2 Property Documentation	31

4.9.2.1 assetPath	31
4.10 AssetSearchUtility	32
4.10.1 Detailed Description	32
4.11 AssetsProxy	32
4.11.1 Detailed Description	33
4.12 Async< T >	33
4.12.1 Detailed Description	33
4.13 BuildOption	33
4.13.1 Detailed Description	34
4.14 Callback	34
4.14.1 Detailed Description	35
4.14.2 Member Enumeration Documentation	35
4.14.2.1 When	35
4.14.3 Member Function Documentation	35
4.14.3.1 Before() [1/2]	35
4.14.3.2 Before() [2/2]	36
4.14.4 Property Documentation	36
4.14.4.1 scene	36
4.15 CallbackUtility	36
4.15.1 Detailed Description	36
4.16 CanvasSortOrderUtility	36
4.16.1 Detailed Description	37
4.16.2 Member Function Documentation	37
4.16.2.1 MakeSure()	37
4.17 CoroutineUtility	37
4.17.1 Detailed Description	38
4.17.2 Member Function Documentation	38
4.17.2.1 StartCoroutineGlobal()	38
4.17.2.2 StopAllCoroutines()	38
4.17.2.3 Timer()	39
4.18 CrossSceneReference	39
4.18.1 Detailed Description	39
4.19 CrossSceneReferenceUtility	39
4.19.1 Detailed Description	40
4.19.2 Member Function Documentation	40
4.19.2.1 CanSceneBeSaved()	40
4.20 DefaultScenes	40
4.20.1 Detailed Description	41
4.20.2 Member Function Documentation	41
4.20.2.1 Enumerate()	41
4.20.3 Property Documentation	41
4.20.3.1 fadeScene	41

4.20.3.2 iconBounceScene	 . 42
4.20.3.3 inGameToolbarScene	 . 42
4.20.3.4 pauseScene	 . 42
4.20.3.5 pressAnyKeyScene	 . 42
4.20.3.6 progressBarScene	 . 42
4.20.3.7 quoteScene	 . 42
4.20.3.8 splashASMScene	 . 43
4.20.3.9 splashFadeScene	 . 43
4.21 DependencyInjectionUtility	 . 43
4.21.1 Detailed Description	 . 43
4.22 ProfileDependent $<$ T $>$.Dict	 . 43
4.22.1 Detailed Description	 . 43
4.23 DictionaryUtility	 . 44
4.23.1 Detailed Description	 . 44
4.24 DynamicCollection	 . 44
4.24.1 Detailed Description	 . 45
4.25 CoroutineUtility.Events	 . 45
4.25.1 Detailed Description	 . 46
4.25.2 Member Function Documentation	 . 46
4.25.2.1 CoroutineFrameEndEvent()	 . 46
4.25.2.2 CoroutineFrameStartEvent()	 . 46
4.25.3 Member Data Documentation	 . 47
4.25.3.1 onSubroutineStart	 . 47
4.26 FallbackSceneUtility	 . 47
4.26.1 Detailed Description	 . 47
4.27 CallbackUtility.FluentInvokeAPI< T >	 . 47
4.27.1 Detailed Description	 . 48
4.27.2 Member Function Documentation	 . 48
4.27.2.1 WithCallback()	 . 48
4.28 GlobalCoroutine	 . 48
4.28.1 Detailed Description	 . 49
4.29 GuidReference	 . 49
4.29.1 Detailed Description	 . 49
4.30 GuidReferenceUtility	 . 50
4.30.1 Detailed Description	 . 50
4.30.2 Member Function Documentation	 . 50
4.30.2.1 GenerateID()	 . 50
4.30.2.2 GetOrAddPersistent()	 . 50
4.31 IApp	 . 51
4.31.1 Detailed Description	 . 51
4.32 IAssetsProvider	 . 51
4.32.1 Detailed Description	. 51

4.33 ICollectionClose	51
4.33.1 Detailed Description	51
4.34 ICollectionCloseAsync	52
4.34.1 Detailed Description	52
4.35 ICollectionExtraDataCallbacks	52
4.35.1 Detailed Description	53
4.36 ICollectionExtraDataCallbacksAsync	53
4.36.1 Detailed Description	53
4.37 ICollectionOpen	53
4.37.1 Detailed Description	54
4.38 ICollectionOpenAsync	54
4.38.1 Detailed Description	54
4.39 Scene.lMethods.lEvent	54
4.39.1 Detailed Description	55
4.40 Scene.lMethods_Target.lEvent	55
4.40.1 Detailed Description	55
4.41 SceneCollection.IMethods.IEvent	56
4.41.1 Detailed Description	56
4.42 SceneCollection.IMethods_Target.IEvent	56
4.42.1 Detailed Description	56
4.43 FadeLoadingScreen	57
4.43.1 Detailed Description	57
4.44 DependencyInjectionUtility.IInjectable	57
4.44.1 Detailed Description	57
4.45 ILockable	57
4.45.1 Detailed Description	58
4.46 Scene.lMethods	58
4.46.1 Detailed Description	58
4.46.2 Member Function Documentation	59
4.46.2.1 Close()	59
4.46.2.2 FinishPreload()	59
4.46.2.3 Open()	59
4.46.2.4 Preload()	59
4.47 SceneCollection.IMethods	59
4.47.1 Detailed Description	60
4.47.2 Member Function Documentation	60
4.47.2.1 Close()	60
4.47.2.2 Open()	60
4.47.2.3 OpenAdditive()	61
4.47.2.4 ToggleOpen()	61
4.48 Scene.lMethods_Target	61
4.48.1 Detailed Description	62

4.48.2 Member Function Documentation	62
4.48.2.1 Close()	62
4.48.2.2 FinishPreload()	62
4.48.2.3 Open()	63
4.48.2.4 Preload()	63
4.49 SceneCollection.IMethods_Target	63
4.49.1 Detailed Description	63
4.50 InitializeInEditorAttribute	64
4.50.1 Detailed Description	64
4.51 InitializeInEditorMethodAttribute	64
4.51.1 Detailed Description	64
4.52 InputBinding	64
4.52.1 Detailed Description	64
4.53 InputButton	65
4.53.1 Detailed Description	65
4.53.2 Member Data Documentation	65
4.53.2.1 path	65
4.54 IProfileManager	65
4.54.1 Detailed Description	66
4.55 IProjectSettings	66
4.55.1 Detailed Description	66
4.56 IQueueable	66
4.56.1 Detailed Description	67
4.56.2 Member Function Documentation	67
4.56.2.1 OnTurn()	67
4.57 IRuntime	67
4.57.1 Detailed Description	69
4.58 SceneCallbacks	69
4.58.1 Detailed Description	69
4.59 ISceneClose	70
4.59.1 Detailed Description	70
4.60 ISceneCloseAsync	70
4.60.1 Detailed Description	70
4.61 ISceneCollection	71
4.61.1 Detailed Description	71
4.62 ISceneManager	71
4.62.1 Detailed Description	73
4.63 SceneOpen	73
4.63.1 Detailed Description	73
4.64 ISceneOpenAsync	73
4.64.1 Detailed Description	74
4.65 LerpUtility	74

4.65.1 Detailed Description	. 74
4.65.2 Member Function Documentation	. 74
4.65.2.1 Lerp()	. 74
4.66 LoadingScreen	. 75
4.66.1 Detailed Description	. 76
4.66.2 Member Function Documentation	. 76
4.66.2.1 OnClose()	. 76
4.66.2.2 OnOpen()	. 76
4.67 LoadingScreenBase	. 76
4.67.1 Detailed Description	. 77
4.67.2 Member Function Documentation	. 77
4.67.2.1 HasPressedAnyKey()	. 77
4.67.3 Member Data Documentation	. 78
4.67.3.1 canvas	. 78
4.67.4 Property Documentation	. 78
4.67.4.1 isOpen	. 78
4.68 LoadingScreenUtility	. 78
4.68.1 Detailed Description	. 79
4.68.2 Member Function Documentation	. 79
4.68.2.1 CloseLoadingScreen()	. 79
4.68.2.2 DoAction()	. 79
4.69 MainThreadUtility	. 80
4.69.1 Detailed Description	. 80
4.69.2 Member Function Documentation	. 80
4.69.2.1 Invoke() [1/2]	. 80
4.69.2.2 Invoke() [2/2]	. 81
4.69.2.3 Invoke < T > () [1/2]	. 81
4.69.2.4 Invoke < T >() [2/2]	. 81
4.70 ObjectReference	. 81
4.70.1 Detailed Description	. 82
4.71 ParallelASMCallbacks	. 82
4.71.1 Detailed Description	. 82
4.72 Profile	. 82
4.72.1 Detailed Description	. 84
4.72.2 Property Documentation	. 84
4.72.2.1 removedCollections	. 84
4.72.2.2 scenes	. 84
4.72.2.3 specialScenes	. 85
4.72.2.4 startupCollections	. 85
4.73 ProfileDependent $<$ T $>$. 85
4.73.1 Detailed Description	. 86
4.73.2 Member Function Documentation	86

4.73.2.1 DoAction()	 . 86
4.73.2.2 DoAction< T2 >()	 . 86
4.73.2.3 GetModel()	 . 86
4.74 ProfileDependentCollection	 . 86
4.74.1 Detailed Description	 . 87
4.74.2 Member Function Documentation	 . 87
4.74.2.1 Close()	 . 87
4.74.2.2 Open()	 . 88
4.74.2.3 OpenAdditive()	 . 88
4.74.2.4 ToggleOpen()	 . 88
4.75 ProfileDependentScene	 . 89
4.75.1 Detailed Description	 . 90
4.75.2 Member Function Documentation	 . 90
4.75.2.1 Close()	 . 90
4.75.2.2 FinishPreload()	 . 90
4.75.2.3 Open()	 . 90
4.75.2.4 Preload()	 . 91
4.76 QueueUtility< T $>$. 91
4.76.1 Detailed Description	 . 91
4.77 ResolvedCrossReference	 . 92
4.77.1 Detailed Description	 . 92
4.77.2 Member Data Documentation	 . 92
4.77.2.1 reference [1/2]	 . 92
4.77.2.2 reference [2/2]	 . 92
4.78 ResolvedReference	 . 92
4.78.1 Detailed Description	 . 92
4.79 Runtime	 . 93
4.79.1 Detailed Description	 . 95
4.79.2 Member Function Documentation	 . 95
4.79.2.1 Close()	 . 95
4.79.2.2 Open()	 . 96
4.79.2.3 OpenAdditive() [1/2]	 . 96
4.79.2.4 OpenAdditive() [2/2]	 . 96
4.79.2.5 SetActive()	 . 96
4.79.2.6 Track() [1/2]	 . 97
4.79.2.7 Track() [2/2]	 . 97
4.79.2.8 Untrack() [1/2]	 . 97
4.79.2.9 Untrack() [2/2]	 . 97
4.79.2.10 UntrackCollections()	 . 98
4.79.2.11 UntrackScenes()	 . 98
4.79.3 Member Data Documentation	 . 98
4.79.3.1 onAllScenesClosed	 . 98

4.79.4 Property Documentation
4.79.4.1 activeScene
4.79.4.2 dontDestroyOnLoad
4.80 Scene
4.80.1 Detailed Description
4.80.2 Member Function Documentation
4.80.2.1 Close()
4.80.2.2 EvalOpenAsPersistent()
4.80.2.3 FindObject< T >()
4.80.2.4 FindObjects < T >()
4.80.2.5 FinishPreload()
4.80.2.6 GetRootGameObjects()
4.80.2.7 Open()
4.80.2.8 Preload()
4.80.2.9 SetSceneLoader< T >()
4.80.3 Property Documentation
4.80.3.1 hasSceneAsset
4.80.3.2 internalScene
4.80.3.3 isImported
4.80.3.4 isLoadingScreen
4.80.3.5 isSpecial
4.80.3.6 isSplashScreen
4.80.3.7 keepOpenWhenCollectionsClose
4.80.3.8 keepOpenWhenNewCollectionWouldReopen
4.80.3.9 openOnPlayMode
4.80.3.10 openOnStartup
4.81 SceneCollection
4.81.1 Detailed Description
4.81.2 Member Function Documentation
4.81.2.1 Close()
4.81.2.2 Open()
4.81.2.3 OpenAdditive()
4.81.2.4 ToggleOpen()
4.81.3 Property Documentation
4.81.3.1 allScenes
4.81.3.2 isStartupCollection
4.81.3.3 userData
4.82 SceneCollectionTemplate
4.82.1 Detailed Description
4.83 SceneLoadArgs
4.83.1 Detailed Description
4.83.2 Member Function Documentation

4.83.2.1 GetOpenedScene()	15
4.83.2.2 SetCompleted() [1/2]	15
4.83.2.3 SetCompleted() [2/2]	15
4.84 SceneLoader	16
4.84.1 Detailed Description	16
4.84.2 Property Documentation	17
4.84.2.1 isGlobal	17
4.84.2.2 Key	17
4.85 SceneLoaderArgsBase	17
4.85.1 Detailed Description	17
4.86 SceneManager	18
4.86.1 Detailed Description	18
4.86.2 Member Function Documentation	18
4.86.2.1 OnInitialized()	18
4.86.3 Property Documentation	18
4.86.3.1 isInitialized	18
4.87 SceneOperation	19
4.87.1 Detailed Description	21
4.87.2 Member Function Documentation	21
4.87.2.1 Activate()	21
4.87.2.2 Cancel()	21
4.87.2.3 Close()	22
4.87.2.4 Focus()	22
4.87.2.5 Open()	22
4.87.2.6 Preload()	22
4.87.2.7 ReportProgress()	22
4.87.2.8 With()	23
4.87.3 Property Documentation	23
4.87.3.1 close	23
4.87.3.2 focus	23
4.87.3.3 keepWaiting	23
4.87.3.4 open	23
4.88 SceneUnloadArgs	24
4.88.1 Detailed Description	24
4.89 SceneUtility	24
4.89.1 Detailed Description	26
4.89.2 Member Function Documentation	
4.89.2.1 CreateDynamic()	26
4.89.2.2 Disable()	26
4.89.2.3 Enable()	26
4.89.2.4 EvaluateFinalSceneList() [1/2]	
4.89.2.5 EvaluateFinalSceneList() [2/2]	27

135

4.89.2.6 FindCollection()
4.89.2.7 SetEnabled()
4.90 ScriptableObjectUtility
4.90.1 Detailed Description
4.90.2 Member Function Documentation
4.90.2.1 Save()
4.91 SerializableDictionary < TKey, TValue >
4.91.1 Detailed Description
4.92 SerializableStringBoolDict
4.92.1 Detailed Description
4.93 SettingsProxy
4.93.1 Detailed Description
4.93.2 Property Documentation
4.93.2.1 profile
4.94 SpamCheck
4.94.1 Detailed Description
4.94.2 Property Documentation
4.94.2.1 Global
4.94.2.2 isEnabled
4.95 SplashScreen
4.95.1 Detailed Description
4.95.2 Member Function Documentation
4.95.2.1 OnClose()
4.95.2.2 OnOpen()
4.96 StandaloneCollection
4.96.1 Detailed Description
4.97 App.StartupProps
4.97.1 Detailed Description
4.97.2 Member Data Documentation
4.97.2.1 fadeColor
4.98 UIFadeExtensions
4.98.1 Detailed Description
4.99 UnityCompatibiltyHelper
4.99.1 Detailed Description

Index

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

ActionUtility
ASMFilePathAttribute
ASMModel
Profile
Scene
SceneCollection
SceneCollectionTemplate
ASMModelExtensions
ASMScriptableSingleton< T >
ASMScriptableSingleton < ASMSettings >
ASMSettings
Assets
AssetSearchUtility
Async< T >
Async< bool >
BuildOption
Callback
CallbackUtility
CanvasSortOrderUtility
CoroutineUtility
CrossSceneReference
CrossSceneReferenceUtility
DefaultScenes
DependencyInjectionUtility
DictionaryUtility
CoroutineUtility.Events
FallbackSceneUtility
CallbackUtility.FluentInvokeAPI< T >
GlobalCoroutine
GuidReference
GuidReferenceUtility
Scene.IMethods.IEvent
Scene
ProfileDependentScene

2 Hierarchical Index

Scene.lMethods_Target.lEvent	5
ASMSceneHelper	5
SceneCollection.IMethods.IEvent	6
SceneCollection	6
ProfileDependentCollection	6
SceneCollection.IMethods_Target.IEvent	6
ASMSceneHelper	5
IFadeLoadingScreen	7
DependencyInjectionUtility.IInjectable	7
IApp	1
App	0
IAssetsProvider	1
AssetsProxy	
IProfileManager	
IProjectSettings	
ASMSettings	
ISceneManager	
Runtime	
IRuntime	
ILockable	
Scene	
SceneCollection	
Scene.lMethods	7
Scene 9 ProfileDependentScene 8	
SceneCollection.IMethods	
SceneCollection	
ProfileDependentCollection	
Scene.IMethods Target	
ASMSceneHelper	
•	
_ 0	
Runtime 9 ASMSceneHelper 2	
InitializeInEditorAttribute	
InitializeInEditorMethodAttribute	
InputBinding	
InputButton	
iQueueable	6
SceneOperation	9
ISceneCallbacks	9
ICollectionClose	1
ICollectionCloseAsync	2
ICollectionExtraDataCallbacks	2
ICollectionExtraDataCallbacksAsync	-
ICollectionOpen	
ICollectionOpenAsync	
ICollectionExtraDataCallbacks	
ICollectionExtraDataCallbacksAsync	
ISceneClose	7
ISceneOpen	7
ISceneOpenAsync	-
ISceneCollection	
DynamicCollection	
=	

1.1 Class Hierarchy 3

SceneCollection	06
LerpUtility	74
LoadingScreenBase	76
LoadingScreen	75
SplashScreen	30
LoadingScreenUtility	78
•	80
,	81
	82
	85
'	85
ProfileDependentScene	89
${\sf ProfileDependent} {<} {\sf SceneCollection} {>} \dots \dots$	85
ProfileDependentCollection	86
QueueUtility < T >	91
ResolvedCrossReference	92
ResolvedReference	92
SceneLoader	16
SceneLoaderArgsBase	17
SceneLoadArgs	14
SceneUnloadArgs	24
SceneManager	
SceneReferenceCollection	
SceneUtility	
ScriptableObjectUtility	
SerializableDictionary < TKey, TValue >	
$Serializable Dictionary < Profile, T > \dots \dots$	
ProfileDependent< T >.Dict	
$Serializable Dictionary < string, Advanced Scene Manager. Models. Input Binding > \dots $	
$Serializable Dictionary < string, bool > \dots $	28
SerializableStringBoolDict	28
SettingsProxy	29
SpamCheck	29
App.StartupProps	32
UIFadeExtensions	33
UnityCompatibiltyHelper	34

4 Hierarchical Index

Chapter 2

Class Index

2.1 Class List

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

ActionUtility	
Contains utility functions for Action	19
App	
Manages startup and quit processes	20
ASMFilePathAttribute	
A FilePathAttribute that supports build	22
ASMModel	
A base class for Profile, SceneCollection and Scene	22
ASMModelExtensions	
Provides utility methods for working with SceneCollection	23
ASMSceneHelper	
Represents scene helper. Contains functions for opening / closing collections and scenes from	
UnityEngine.Events.UnityEvent	25
ASMScriptableSingleton< T >	
A ScriptableSingleton <t> that supports build</t>	28
ASMSettings	
Contains the core of ASM assets. Contains projectSettings and assets	29
Assets	
Manages all ASM assets	31
AssetSearchUtility	
Provides utility functions for searching ASM assets	32
AssetsProxy	
Provides access to the scenes, collections and profiles managed by ASM	32
Async< T >	
Represents a async operation that returns a value	33
BuildOption	
Represents an enabled state depending on build context (editor, dev build, non-dev build)	33
Callback	
Represents a callback that can be run on Phase change, or right before loading screen hide (or	
when it would, if it was enabled)	34
CallbackUtility	
An utility class that invokes callbacks (defined in interfaces based on ISceneCallbacks), and	
tracks performance and provides tools for optimizing and diagnosing bottlenecks in these call-	
backs	36
CanvasSortOrderUtility	
An utility class to manage sort order on canvases	36

6 Class Index

CoroutineUtility	
An utility class that helps with running coroutines detached from MonoBehaviour	37
CrossSceneReference	
A reference to a variable that references another object in some other scene	39
CrossSceneReferenceUtility An utility for saving and restoring cross-scene references	39
DefaultScenes	00
Provides access to the default ASM scenes	40
DependencyInjectionUtility	
Contains utility methods for depdency injection	43
ProfileDependent< T >.Dict	
A dictionary of type Profile, T	43
DictionaryUtility	
Contains utility functions for working with dictionaries	44
DynamicCollection	
Represents a collection that can take a path and then gather all scenes within, guaranteeing that they are all added to build, including non-imported and blacklisted scenes	44
CoroutineUtility.Events	44
Provides events for coroutine events	45
FallbackSceneUtility	
An utility class that manages the default scene, called 'AdvancedSceneManager'	47
CallbackUtility.FluentInvokeAPI <t></t>	
An helper class to facilitate a fluent api	47
GlobalCoroutine	
Represents a IEnumerator coroutine started using CoroutineUtility	48
GuidReference	
Represents a persistent reference to the GameObject that this is attached to, see also Guid ReferenceUtility	49
GuidReferenceUtility	73
An utility for referencing objects globally	50
IApp	
51	
lAssetsProvider Section 1997	
51	
CollectionClose	51
ICollectionCloseAsync	52
Callbacks for a ScriptableObject that has been set as extra data for a collection	52
ICollectionExtraDataCallbacksAsync	52
Callbacks for a ScriptableObject that has been set as extra data for a collection	53
CollectionOpen	53
ICollectionOpenAsync	54
Scene.IMethods.IEvent	54
Scene.IMethods_Target.IEvent	55
SceneCollection.IMethods.IEvent	56
SceneCollection.IMethods_Target.IEvent	56
FadeLoadingScreen	57
Used to pass arguments from LoadingScreenUtility.FadeIn(LoadingScreen, float, Color?) DependencyInjectionUtility.IInjectable	37
Base interface for all injectable services	57
Lockable	٠.
Specifies a object that can be locked, using LockUtility	57
Scene.IMethods	58
SceneCollection.IMethods	59
Scene.IMethods_Target	61
SceneCollection.IMethods_Target	63
InitializeInEditorAttribute	٠.
Initializes a class in editor on recompile	64

2.1 Class List 7

InitializeInEditorMethodAttribute	
Initializes a class in editor on recompile	64
InputBinding Represents a input binding for InputSystem. Available even when InputSystem is uninstalled .	64
InputButton	
Specifies a input binding for use with InputSystem	65
IProfileManager	
Manages the current profile	65
IProjectSettings	
66	
IQueueable	
Represents a queueable item	66
IRuntime	
67	
ISceneCallbacks	
Base interface for ISceneOpen, ISceneClose, ICollectionOpen, ICollectionClose. Does nothing	00
on its own, used by CallbackUtility	69
ISceneClose	70
Callback for when the scene that a MonoBehaviour is contained within is closed	70
ISceneColseAsync	70
Represents the core variables of what makes up a scene collection	71
ISceneManager	/
71	
ISceneOpen	
Callback for when the scene that a MonoBehaviour is contained within is opened	73
ISceneOpenAsync	73
LerpUtility	/ (
Provides some convinience functions for lerping	74
LoadingScreen	,-
A class that contains callbacks for loading screens	75
LoadingScreenBase	
A generic base class for loading screens. You probably want to inherit from LoadingScreen	
though	76
LoadingScreenUtility	
Manager for loading screens	78
MainThreadUtility	80
ObjectReference	
A reference to an object in a scene	81
ParallelASMCallbacks	
Specifies whatever the ASM callbacks should be run in parallel for any callbacks defined in this	
script	82
Profile	
A profile for ASM, contains settings and collections	82
ProfileDependent< T >	
Specifies a T that changes depending on active Profile	85
ProfileDependentCollection	
Represents a SceneCollection that changes depending on active Profile	86
ProfileDependentScene	
Represents a Scene that changes depending on active Profile	89
QueueUtility< T >	
A utility that provides queuing	91
ResolvedCrossReference	
Represents a resolved reference	92
ResolvedReference	00
Represents a resolved ObjectReference	92
Runtime Manages runtime functionality for Advanced Scope Manager such as open scopes and collection	01
Manages runtime functionality for Advanced Scene Manager such as open scenes and collection	93

8 Class Index

Scene	
Represents a scene	99
SceneCollection	
Represents a collection of scenes	106
SceneCollectionTemplate	
Represents a template for a SceneCollection	111
SceneLoadArgs	
Specifies arguments for SceneLoader.LoadScene(Models.Scene, SceneLoadArgs)	114
SceneLoader	440
Specifies a scene loader	116
SceneLoaderArgsBase	447
Base class for SceneLoadArgs and SceneUnloadArgs	117
SceneManager The control Advanced Course Managers ABL Breaking a course to the great increase the break in a control of the course of the cou	
The central Advanced Scene Manager API. Provides access to the most important things in ASI	VI I I8
SceneOperation	
A scene operation is a queueable operation that can open or close scenes. See also: Scene ←	
Action	119
SceneUnloadArgs	104
Specifies arguments for SceneLoader.UnloadScene(Models.Scene, SceneUnloadArgs)	124
SceneUtility	104
An utility class to perform actions on scenes	124
ScriptableObjectUtility Contains utility methods for ScriptableObject	107
Contains utility methods for ScriptableObject	127
Serializable Dictionary < TKey, TValue >	100
A serializable dictionary	128
A serializable dictionary of string and bool	128
SettingsProxy	120
Provides access to ASM settings	129
SpamCheck	129
Provides an easy way to check for spamming	129
SplashScreen	123
A class that contains callbacks for splash screens	130
StandaloneCollection	130
Represents a collection of standalone scenes. These scenes are guaranteed to be included in	
build (if the associated Profile is active)	132
App.StartupProps	102
An object that persists start properties across domain reload, which is needed when configurable	
enter play mode is set to reload domain on enter play mode	132
UlFadeExtensions	102
Provides extension methods for CanvasGroup	133
UnityCompatibilityHelper	.00
Contains helpers for dealing with multiple versions of unity	134

Chapter 3

Namespace Documentation

3.1 AdvancedSceneManager Namespace Reference

Classes

· class SceneManager

The central Advanced Scene Manager API. Provides access to the most important things in ASM.

3.2 AdvancedSceneManager.Callbacks Namespace Reference

Classes

· class ActionUtility

Contains utility functions for Action.

· class CallbackUtility

An utility class that invokes callbacks (defined in interfaces based on ISceneCallbacks), and tracks performance and provides tools for optimizing and diagnosing bottlenecks in these callbacks.

- interface ICollectionClose
- interface ICollectionCloseAsync
- interface ICollectionExtraDataCallbacks

Callbacks for a ScriptableObject that has been set as extra data for a collection.

• interface ICollectionExtraDataCallbacksAsync

Callbacks for a ScriptableObject that has been set as extra data for a collection.

- interface ICollectionOpen
- interface ICollectionOpenAsync
- interface ISceneCallbacks

Base interface for ISceneOpen, ISceneClose, ICollectionOpen, ICollectionClose. Does nothing on its own, used by CallbackUtility.

interface ISceneClose

Callback for when the scene that a MonoBehaviour is contained within is closed.

- interface ISceneCloseAsync
- interface ISceneOpen

Callback for when the scene that a MonoBehaviour is contained within is opened.

- interface ISceneOpenAsync
- · class LoadingScreen

A class that contains callbacks for loading screens.

· class LoadingScreenBase

A generic base class for loading screens. You probably want to inherit from LoadingScreen though.

· class ParallelASMCallbacks

Specifies whatever the ASM callbacks should be run in parallel for any callbacks defined in this script.

· class SplashScreen

A class that contains callbacks for splash screens.

3.3 AdvancedSceneManager.Core Namespace Reference

Classes

class App

Manages startup and quit processes.

class Callback

Represents a callback that can be run on Phase change, or right before loading screen hide (or when it would, if it was enabled).

· class Runtime

Manages runtime functionality for Advanced Scene Manager such as open scenes and collection.

class SceneLoadArgs

Specifies arguments for SceneLoader.LoadScene(Models.Scene, SceneLoadArgs).

· class SceneLoader

Specifies a scene loader.

class SceneLoaderArgsBase

Base class for SceneLoadArgs and SceneUnloadArgs.

class SceneOperation

A scene operation is a queueable operation that can open or close scenes. See also: SceneAction.

• class SceneUnloadArgs

Specifies arguments for SceneLoader.UnloadScene(Models.Scene, SceneUnloadArgs).

Enumerations

```
    enum Phase {
        CloseCallbacks , UnloadScenes , LoadScenes , OpenCallbacks ,
        CustomActions }
```

The phase that a SceneOperation is currently in.

3.3.1 Enumeration Type Documentation

3.3.1.1 Phase

enum Phase

The phase that a SceneOperation is currently in.

Enumerator

CloseCallbacks	The scene operation is currently executing close callbacks on the scenes that are being closed, if any.
UnloadScenes	The scene operation is currently unloading the scenes, if any.
LoadScenes	The scene operation is currently loading the scenes, if any.
OpenCallbacks	The scene operation is currently executing open callbacks on the scenes that are being opened, if any.
CustomActions	The scene operation is currently executing custom actions, added through SceneOperation.WithAction(SceneAction[]) or similar methods, if any.

3.4 AdvancedSceneManager.DependencyInjection Namespace Reference

Classes

· class DependencyInjectionUtility

Contains utility methods for depdency injection.

- interface IApp
- interface IAssetsProvider
- · interface IProfileManager

Manages the current profile.

- interface IProjectSettings
- interface IRuntime
- interface ISceneManager

3.5 AdvancedSceneManager.DependencyInjection.Editor Namespace Reference

3.6 AdvancedSceneManager.Models Namespace Reference

Classes

class ASMModel

A base class for Profile, SceneCollection and Scene.

class ASMModelExtensions

Provides utility methods for working with SceneCollection.

• class ASMSceneHelper

Represents scene helper. Contains functions for opening / closing collections and scenes from UnityEngine. Events. ← UnityEvent.

class ASMSettings

Contains the core of ASM assets. Contains projectSettings and assets

class DynamicCollection

Represents a collection that can take a path and then gather all scenes within, guaranteeing that they are all added to build, including non-imported and blacklisted scenes.

interface ILockable

Specifies a object that can be locked, using LockUtility.

· class InputBinding

Represents a input binding for InputSystem. Available even when InputSystem is uninstalled.

struct InputButton

Specifies a input binding for use with InputSystem.

• interface ISceneCollection

Represents the core variables of what makes up a scene collection.

class Profile

A profile for ASM, contains settings and collections.

· class Scene

Represents a scene.

· class SceneCollection

Represents a collection of scenes.

· class StandaloneCollection

Represents a collection of standalone scenes. These scenes are guaranteed to be included in build (if the associated Profile is active).

Enumerations

enum InputBindingInteractionType { Open , Hold , Toggle }

Specifies the interaction type to use for scene bindings.

3.6.1 Enumeration Type Documentation

3.6.1.1 InputBindingInteractionType

```
{\tt enum} \ {\tt InputBindingInteractionType}
```

Specifies the interaction type to use for scene bindings.

Enumerator

Open	Specifies that the scene or collection will be opened automatically, but not closed.
Hold	Specifies that the scene or collection will be opened automatically on button down, then closed on
	button up.
Toggle	Specifies that the scene or collection will be opened automatically on button down, then closed on
	next button down.

3.7 AdvancedSceneManager.Models.Enums Namespace Reference

Enumerations

```
    enum CollectionLoadingThreadPriority {
        Auto = -2 , Low = ThreadPriority.Low , BelowNormal = ThreadPriority.BelowNormal , Normal = Thread←
        Priority.Normal ,
        High = ThreadPriority.High }
```

Wrapper for ThreadPriority, adds CollectionLoadingThreadPriority.Auto.

enum CollectionStartupOption { Auto , Open , DoNotOpen }

Specifies what to do with a SceneCollection during startup.

• enum EditorPersistentOption { Never , WhenAnyOfTheFollowingScenesAreOpened , AnySceneOpened }

Specifies whatever a scene should be automatically opened outside of play-mode.

enum LoadingScreenUsage { DoNotUse , UseDefault , Override }

Specifies what loading screen to use, if any.

• enum SceneImportOption { Manual , SceneCreated }

Specifies how to scenes are imported.

enum SceneState {
 Unknown , NotOpen , Queued , Opening ,
 Preloading , Preloaded , Open }

Specifies that state of a scene.

3.7.1 Enumeration Type Documentation

3.7.1.1 CollectionLoadingThreadPriority

enum CollectionLoadingThreadPriority

Wrapper for ThreadPriority, adds CollectionLoadingThreadPriority.Auto.

ThreadPriority:

Enumerator

Auto	Automatically decide ThreadPriority based on if loading screen is open.
Low	Lowest thread priority.
BelowNormal	Below normal thread priority.
Normal	Normal thread priority.
High	Highest thread priority.

3.7.1.2 CollectionStartupOption

 $\verb"enum CollectionStartupOption"$

Specifies what to do with a SceneCollection during startup.

Enumerator

Auto	Specifies that ASM should automatically decide if a SceneCollection should be opened during startup. This means that if no collection in the list specifies either Open or OpenAsPersistent, then the first collection in the list that has Auto will be opened.
Open	Specifies that a SceneCollection will open during startup.
DoNotOpen	Specifies that a SceneCollection will not open during startup.

3.7.1.3 EditorPersistentOption

enum EditorPersistentOption

Specifies whatever a scene should be automatically opened outside of play-mode.

Enumerator

Never	Never automatically open scene.
WhenAnyOfTheFollowingScenesAreOpened	Automatically open scene when any specified scene is opened.
AnySceneOpened	Automatically open scene when any scene opens.

3.7.1.4 LoadingScreenUsage

enum LoadingScreenUsage

Specifies what loading screen to use, if any.

Enumerator

DoNotUse	Specifies no loading screen.
UseDefault	Specifies default loading screen, defined in profile settings.
Override	Specifies overriden loading screen, defined in SceneCollection.

3.7.1.5 SceneImportOption

enum SceneImportOption

Specifies how to scenes are imported.

Enumerator

Manual	User will manually import scenes.	
SceneCreated	Scenes will be automatically imported when created, otherwise manual.]

3.7.1.6 SceneState

enum SceneState

Specifies that state of a scene.

Enumerator

Unknown	The state of the scene is unknown. (An issue probably occured while checking state)	
NotOpen	The scene is not open.	
Queued	The scene is in queue to be opened.	

Enumerator

Opening	The scene is currently being opened. Mutually exclusive to Preloading.
Preloading	The scene is currently being preloaded. Mutually exclusive to Opening.
Preloaded	The scene is currently preloaded.
Open	The scene is open.

3.8 AdvancedSceneManager.Models.Helpers Namespace Reference

Classes

class AssetsProxy

Provides access to the scenes, collections and profiles managed by ASM.

· class DefaultScenes

Provides access to the default ASM scenes.

class SettingsProxy

Provides access to ASM settings.

3.9 AdvancedSceneManager.Models.Internal Namespace Reference

Classes

· class Assets

Manages all ASM assets.

3.10 AdvancedSceneManager.Models.Utility Namespace Reference

Classes

· class BuildOption

Represents an enabled state depending on build context (editor, dev build, non-dev build).

· class ProfileDependent

Specifies a T that changes depending on active Profile.

• class ProfileDependentCollection

Represents a SceneCollection that changes depending on active Profile.

· class ProfileDependentScene

Represents a Scene that changes depending on active Profile.

• class SceneCollectionTemplate

Represents a template for a SceneCollection.

3.11 AdvancedSceneManager.Utility Namespace Reference

Classes

· class ASMFilePathAttribute

A FilePathAttribute that supports build.

class ASMScriptableSingleton

A ScriptableSingleton<T> that supports build.

· class AssetSearchUtility

Provides utility functions for searching ASM assets.

class Async

Represents a async operation that returns a value.

· class CanvasSortOrderUtility

An utility class to manage sort order on canvases.

· class DictionaryUtility

Contains utility functions for working with dictionaries.

class FallbackSceneUtility

An utility class that manages the default scene, called 'AdvancedSceneManager'.

· class GuidReference

Represents a persistent reference to the GameObject that this is attached to, see also GuidReferenceUtility .

· class GuidReferenceUtility

An utility for referencing objects globally.

interface IFadeLoadingScreen

Used to pass arguments from LoadingScreenUtility.FadeIn(LoadingScreen, float, Color?)

class InitializeInEditorAttribute

Initializes a class in editor on recompile.

· class InitializeInEditorMethodAttribute

Initializes a class in editor on recompile.

• interface |Queueable

Represents a queueable item.

· class LerpUtility

Provides some convinience functions for lerping.

class LoadingScreenUtility

Manager for loading screens.

class QueueUtility

A utility that provides queuing.

class SceneUtility

An utility class to perform actions on scenes.

· class ScriptableObjectUtility

Contains utility methods for ScriptableObject.

· class SerializableDictionary

A serializable dictionary.

• class SerializableStringBoolDict

A serializable dictionary of string and bool.

class SpamCheck

Provides an easy way to check for spamming.

· class UIFadeExtensions

Provides extension methods for CanvasGroup.

class UnityCompatibiltyHelper

Contains helpers for dealing with multiple versions of unity.

3.12 AdvancedSceneManager.Utility.CrossSceneReferences Namespace Reference

Classes

class CrossSceneReference

A reference to a variable that references another object in some other scene.

· class CrossSceneReferenceUtility

An utility for saving and restoring cross-scene references.

· class ObjectReference

A reference to an object in a scene.

• struct ResolvedCrossReference

Represents a resolved reference.

• struct ResolvedReference

Represents a resolved ObjectReference.

• class SceneReferenceCollection

A collection of CrossSceneReference for a scene. More...

Enumerations

• enum ResolveStatus

Specifies the result of a resolve.

enum SceneStatus { Default , Restored , Cleared }

Specifies the state of a scene.

3.12.1 Class Documentation

3.12.1.1 class AdvancedSceneManager::Utility::CrossSceneReferences::SceneReferenceCollection

A collection of CrossSceneReference for a scene.

3.12.2 Enumeration Type Documentation

3.12.2.1 SceneStatus

enum SceneStatus

Specifies the state of a scene.

Enumerator

Default	Cross-scene reference utility has not done anything to this scene.
Restored	Cross-scene reference utility has restored references in this scene.
Cleared	Cross-scene reference utility has cleared references in this scene.

3.13 Lazy Namespace Reference

3.14 Lazy. Utility Namespace Reference

Classes

· class CoroutineUtility

An utility class that helps with running coroutines detached from MonoBehaviour.

· class GlobalCoroutine

Represents a IEnumerator coroutine started using CoroutineUtility.

class MainThreadUtility

Chapter 4

Class Documentation

4.1 ActionUtility

Contains utility functions for Action.

Static Public Member Functions

• static void LogInvoke (this Action action)

Tries to invoke the action, then logs error to the console if an error occurred.

• static void **Trylnvoke** (this Action action)

Tries to invoke the action, eats the exception.

• static bool TryInvoke (this Action action, [NotNullWhen(false)] out Exception exception)

Tries to invoke the action.

4.1.1 Detailed Description

Contains utility functions for Action.

4.1.2 Member Function Documentation

4.1.2.1 Trylnvoke()

Tries to invoke the action.

Parameters

action	The action to invoke.
exception The exception that occurred when invoking action. null if true was	

20 Class Documentation

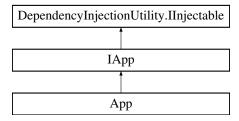
Returns

true if invoke succeeded with no exception.

4.2 App

Manages startup and quit processes.

Inheritance diagram for App:



Classes

class StartupProps

An object that persists start properties across domain reload, which is needed when configurable enter play mode is set to reload domain on enter play mode.

Public Member Functions

- void Restart (StartupProps props=null)
- Async< bool > RestartAsync (StartupProps props=null)
- void RegisterQuitCallback (IEnumerator coroutine)

Register a callback to be called before quit.

void UnregisterQuitCallback (IEnumerator coroutine)

Unregister a callback that was to be called before quit.

void CancelQuit ()

Cancels a quit in progress.

• void Quit (bool fade=true, Color? fadeColor=null, float fadeDuration=1)

Quits the game, and calls quitCallbacks, optionally with a fade animation.

• void Exit ()

Exits the game like you normally would in unity.

Properties

• StartupProps startupProps [get, set]

Gets the props that should be used for startup process.

bool isStartupFinished [get]

Gets if startup process is finished.

bool isRestart [get]

Gets if ASM has been restarted, or is currently restarting.

bool isQuitting [get]

Gets whatever ASM is currently in the process of quitting.

4.2 App 21

Events

· Action beforeRestart

Occurs before restart process has begun, but has been initiated.

Action afterRestart

Occurs after restart has been completed.

4.2.1 Detailed Description

Manages startup and quit processes.

Usage: SceneManager.app.

4.2.2 Member Function Documentation

4.2.2.1 CancelQuit()

```
void CancelQuit ( )
```

Cancels a quit in progress.

Only usable during a RegisterQuitCallback(IEnumerator) or while isQuitting is true.

Implements IApp.

4.2.2.2 Exit()

```
void Exit ( )
```

Exits the game like you normally would in unity.

No callbacks will be called, and no fade out will occur.

Implements IApp.

4.2.2.3 Quit()

Quits the game, and calls quitCallbacks, optionally with a fade animation.

Parameters

fade	Specifies whatever screen should fade out.
fadeColor	Defaults to ProjectSettings.buildUnitySplashScreenColor.
fadeDuration	Specifies the duration of the fade out.

22 Class Documentation

Implements IApp.

4.3 ASMFilePathAttribute

A FilePathAttribute that supports build.

Inherits Attribute.

Properties

• string path [get]

The path to the associated ScriptableSingleton<T>.

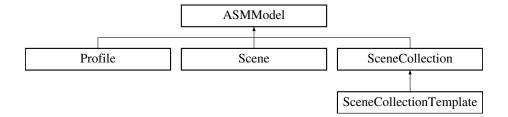
4.3.1 Detailed Description

A FilePathAttribute that supports build.

4.4 ASMModel

A base class for Profile, SceneCollection and Scene.

Inheritance diagram for ASMModel:



Public Member Functions

• virtual void Save ()

Saves the singleton to disk after a delay.

• void SaveNow ()

Saves the singleton to disk.

• void SaveNow (bool setDirty=true)

Saves the singleton to disk.

• virtual bool IsMatch (string q)

Gets if q matches name.

Static Public Member Functions

• static string GenerateID ()

Generate id.

4.5 ASMModelExtensions 23

Static Protected Member Functions

• static T CreateInternal < T > (string name)

Creates a profile. Throws if name is invalid.

Properties

```
string id [get]
Gets the id of this ASMModel.
new string name [get, protected set]
```

4.4.1 Detailed Description

A base class for Profile, SceneCollection and Scene.

4.4.2 Member Function Documentation

4.4.2.1 Save()

```
virtual void Save ( ) [virtual]
```

Saves the singleton to disk after a delay.

Can be called outside of editor, but has no effect.

4.4.2.2 SaveNow() [1/2]

```
void SaveNow ( )
```

Saves the singleton to disk.

Can be called outside of editor, but has no effect.

4.4.2.3 SaveNow() [2/2]

Saves the singleton to disk.

Can be called outside of editor, but has no effect.

4.5 ASMModelExtensions

Provides utility methods for working with SceneCollection.

Static Public Member Functions

• static int IndexOf< T > (this T collection, Scene scene)

Finds the index of scene.

• static SceneOperation OpenAdditive (this IEnumerable < SceneCollection > collections)

Opens the collections as additive.

static SceneOperation OpenAdditive (this IEnumerable < SceneCollection > collections, SceneCollection activeCollection)

Opens the collections as additive.

static SceneOperation OpenAdditive (this IEnumerable < SceneCollection > collections, SceneCollection activeCollection, Scene loadingScene)

Opens the collections as additive.

static SceneOperation OpenWithAdditive (this SceneCollection collection, params SceneCollection[] extraAdditiveCollections)

Opens this collection and then opens extraAdditiveCollections as additive.

static SceneOperation OpenAII (this IEnumerable < Scene > scenes)

Opens the scenes .

static SceneOperation OpenAll (this IEnumerable < Scene > scenes, Scene loadingScene)

Opens the scenes .

static SceneOperation CloseAll (this IEnumerable < Scene > scenes)

Closes the scenes .

static SceneOperation CloseAll (this IEnumerable < Scene > scenes, Scene loadingScene)

Closes the scenes .

4.5.1 Detailed Description

Provides utility methods for working with SceneCollection.

4.5.2 Member Function Documentation

4.5.2.1 CloseAll()

Closes the scenes.

Parameters

```
loadingScene Cover this operation with loadingScene.
```

4.5.2.2 IndexOf< T >()

4.6 ASMSceneHelper 25

Finds the index of scene.

Returns -1 if it does not exist.

Type Constraints

T: ISceneCollection

4.5.2.3 OpenAdditive() [1/2]

Opens the collections as additive.

If activeCollection is part of collections, then it will only be opened once, not as additive.

4.5.2.4 OpenAdditive() [2/2]

Opens the collections as additive.

If activeCollection is part of collections, then it will only be opened once, not as additive.

4.5.2.5 OpenAII()

Opens the scenes.

Parameters

```
loadingScene Cover this operation with loadingScene.
```

4.6 ASMSceneHelper

Represents scene helper. Contains functions for opening / closing collections and scenes from UnityEngine.
Events. UnityEvent.

Inheritance diagram for ASMSceneHelper:



Public Member Functions

- SceneOperation Open (SceneCollection collection, bool openAll=false)
- SceneOperation OpenAdditive (SceneCollection collection, bool openAll=false)
- SceneOperation ToggleOpen (SceneCollection collection, bool openAll=false)
- SceneOperation Close (SceneCollection collection)
- void **Open** (SceneCollection collection)
- void _OpenAdditive (SceneCollection collection)
- void _ToggleOpen (SceneCollection collection)
- void _Close (SceneCollection collection)
- SceneOperation Open (Scene scene)

Opens the specified scene.

SceneOperation OpenAndActivate (Scene scene)

Opens the scene and activates it.

• SceneOperation ToggleOpen (Scene scene)

Toggles the open state of the specified scene.

SceneOperation Close (Scene scene)

Closes the specified scene.

• SceneOperation Preload (Scene scene, Action onPreloaded=null)

Preloads the specified scene, to be displayed at a later time. See also: FinishPreload(Scene), DiscardPreload(← Scene).

SceneOperation FinishPreload (Scene scene)

Finishes preloading the specified scene, displaying it.

SceneOperation DiscardPreload (Scene scene)

Discards the specified scene, if preloaded.

• SceneOperation OpenWithLoadingScreen (Scene scene, Scene loadingScene)

Opens the specified scene while a loading screen is open.

• void **SetActive** (Scene scene)

Sets the specified scene as active in heirarchy.

- void _Open (Scene scene)
- void _OpenAndActivate (Scene scene)
- void _ToggleOpen (Scene scene)
- void Close (Scene scene)
- void _Preload (Scene scene)
- void _FinishPreload (Scene scene)
- void _DiscardPreload (Scene scene)
- void _SetActive (Scene scene)
- void OpenWhereNameStartsWith (string name)

Open all scenes that starts with the specified name.

- void Quit ()
- void Restart ()
- void RestartCollection ()

Properties

- new string name [get]
- static ASMSceneHelper instance [get]

Gets a reference to scene helper.

4.6 ASMSceneHelper 27

4.6.1 Detailed Description

Represents scene helper. Contains functions for opening / closing collections and scenes from UnityEngine. ← Events.UnityEvent.

4.6.2 Member Function Documentation

4.6.2.1 Close()

```
SceneOperation Close (
Scene scene)
```

Closes the specified scene.

Already closed scenes not affected.

Implements Scene.IMethods_Target.

4.6.2.2 FinishPreload()

Finishes preloading the specified scene, displaying it.

Scene must be preloaded beforehand.

Implements Scene.IMethods_Target.

4.6.2.3 Open()

```
SceneOperation Open (
Scene scene)
```

Opens the specified scene.

Already open scenes not affected.

Implements Scene.IMethods_Target.

4.6.2.4 Preload()

Preloads the specified scene, to be displayed at a later time. See also: FinishPreload(Scene), DiscardPreload(← Scene).

Scene must be closed beforehand.

Implements Scene.IMethods_Target.

4.7 ASMScriptableSingleton < T >

A ScriptableSingleton<T> that supports build.

Inherits ScriptableObject, and INotifyPropertyChanged.

Public Member Functions

• virtual void Save ()

Saves the singleton to disk after a delay.

• void SaveNow ()

Saves the singleton to disk.

Properties

virtual bool editorOnly [get]
 Specifies that build support will not be applied to this ScriptableSingleton<T>.

4.7.1 Detailed Description

A ScriptableSingleton<T> that supports build.

Type Constraints

T: ASMScriptableSingleton<T>

4.7.2 Member Function Documentation

4.7.2.1 Save()

```
virtual void Save ( ) [virtual]
```

Saves the singleton to disk after a delay.

Can be called outside of editor, but has no effect.

4.7.2.2 SaveNow()

```
void SaveNow ( )
```

Saves the singleton to disk.

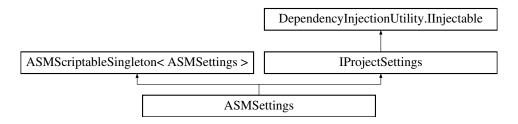
Can be called outside of editor, but has no effect.

4.8 ASMSettings 29

4.8 ASMSettings

Contains the core of ASM assets. Contains projectSettings and assets

Inheritance diagram for ASMSettings:



Static Public Member Functions

· static void OnInitialized (Action action)

Runs the callback when ASMSettings has initialized.

Properties

Profile defaultProfile [get, set]

The profile to use when none is set.

Profile forceProfile [get, set]

The profile to force everyone in this project to use.

• Profile buildProfile [get]

The profile to use during build.

• bool checkForDuplicateSceneOperations [get, set]

By default, ASM checks for duplicate scene operations, since this is usually caused by mistake, but this will disable that.

• bool preventSpammingEventMethods [get, set]

By default, ASM will prevent spam calling event methods (i.e. calling Scene.Open() from a button press), but this will disable that.

float spamCheckCooldown [get, set]

Sets the default cooldown for SpamCheck.

• bool enableCrossSceneReferences [get, set]

Gets or sets whatever cross-scene references should be enabled.

SceneImportOption sceneImportOption [get, set]

Gets or sets when to automatically import scenes.

• bool allowExcludingCollectionsFromBuild [get, set]

Specifies whatever collections can be excluded from build.

• bool reverseUnloadOrderOnCollectionClose [get, set]

Specifies whatever collections should unload scenes in the reverse order.

string assetPath [get, set]

Specifies the path where profiles and imported scenes should be generated to.

CustomData customData [get]

Specifies custom data.

Color buildUnitySplashScreenColor [get]

This is the color of the unity splash screen, used to make the transition from unity splash screen to ASM smooth, this is set before building. Color.black is used when the unity splash screen is disabled.

```
• bool allowSceneLocking [get, set]
```

Specifies whatever asm will allow locking scenes.

bool allowCollectionLocking [get, set]

Specifies whatever asm will allow locking collections.

• Scene fadeScene [get, set]

Specifies the scene to use for certain methods, i.e. LoadingScreenUtility.FadeOut(float, Color?, Action<float>).

Properties inherited from ASMScriptableSingleton < ASMSettings >

virtual bool editorOnly [get]

Specifies that build support will not be applied to this ScriptableSingleton< T>.

Properties inherited from IProjectSettings

Additional Inherited Members

Public Member Functions inherited from ASMScriptableSingleton < ASMSettings >

· virtual void Save ()

Saves the singleton to disk after a delay.

• void SaveNow ()

Saves the singleton to disk.

Public Member Functions inherited from IProjectSettings

- void Save ()
- void SaveNow ()

4.8.1 Detailed Description

Contains the core of ASM assets. Contains projectSettings and assets

Only available in editor.

4.8.2 Property Documentation

4.8.2.1 allowExcludingCollectionsFromBuild

bool allowExcludingCollectionsFromBuild [get], [set]

Specifies whatever collections can be excluded from build.

When true, a toggle will be shown in scene manager window.

Implements IProjectSettings.

4.9 Assets 31

4.8.2.2 enableCrossSceneReferences

```
bool enableCrossSceneReferences [get], [set]
```

Gets or sets whatever cross-scene references should be enabled.

Experimental.

Implements IProjectSettings.

4.9 Assets

Manages all ASM assets.

Properties

```
• static IEnumerable < Profile > profiles [get]
```

Enumerates all imported profiles.

• static IEnumerable < SceneCollection > collections [get]

Enumerates all imported collections.

static IEnumerable < SceneCollectionTemplate > collectionTemplates [get]

Enumerates all imported collection templates.

• static | Enumerable < Scene > scenes [get]

Enumerates all imported scenes.

• static ASMSceneHelper sceneHelper [get]

Gets scene helper singleton.

• static DefaultScenes defaultScenes [get]

Gets default scenes singleton.

• static | Enumerable < Object > all Assets [get]

Enumerates all imported assets.

static string assetPath [get]

Gets the import path.

• static string fallbackScenePath [get]

Gets the path to the fallback scene.

4.9.1 Detailed Description

Manages all ASM assets.

4.9.2 Property Documentation

4.9.2.1 assetPath

```
string assetPath [static], [get]
```

Gets the import path.

Can be changed using ProjectSettings.assetPath

4.10 AssetSearchUtility

Provides utility functions for searching ASM assets.

Static Public Member Functions

• static T **Find**< **T** > (string q)

Finds the T with the specified name.

static bool TryFind< T > (string q, out T result)

Finds the T with the specified name.

- static T **Find**< **T** > (this T[] list, string q)
- static bool TryFind< T > (this T[] list, string q, out T result)
- static T **Find**< T > (this IEnumerable< T > list, string q)
- static bool TryFind< T > (this IEnumerable< T > list, string q, out T result)

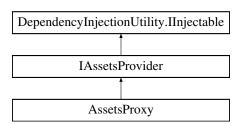
4.10.1 Detailed Description

Provides utility functions for searching ASM assets.

4.11 AssetsProxy

Provides access to the scenes, collections and profiles managed by ASM.

Inheritance diagram for AssetsProxy:



Public Member Functions

IEnumerable < T > Enumerate < T > ()
 Enumerates T.

Properties

• IEnumerable < Profile > profiles [get]

Enumerates all profiles in the project.

• IEnumerable < Scene > scenes [get]

Enumerates all scenes.

• IEnumerable < SceneCollection > collections [get]

Enumerates all collections.

• IEnumerable < SceneCollectionTemplate > templates [get]

Enumerates all templates.

• DefaultScenes defaults [get]

Provides access to the default ASM scenes.

4.12 Async< T > 33

4.11.1 Detailed Description

Provides access to the scenes, collections and profiles managed by ASM.

4.12 Async< T >

Represents a async operation that returns a value.

Inherits CustomYieldInstruction.

Public Member Functions

void OnComplete (Action< T > callback)

Calls the callback when the async operation is complete.

Properties

```
    static Async< T > complete = new(null) [get]
    Gets a Async<T> that is already completed.
```

• T value [get, set]

Gets the value that was produced by the async operation.

4.12.1 Detailed Description

Represents a async operation that returns a value.

4.13 BuildOption

Represents an enabled state depending on build context (editor, dev build, non-dev build).

Inherits INotifyPropertyChanged.

Public Member Functions

• bool GetIsEnabledInCurrentContext ()

Get whatever we should be enabled in the current context.

Properties

bool enableInEditor [get, set]

Gets whatever we should be enabled in editor.

• bool enableInDevBuild [get, set]

Gets whatever we should be enabled in dev build.

bool enableInNonDevBuild [get, set]

Gets whatever we should be enabled in non-dev build.

4.13.1 Detailed Description

Represents an enabled state depending on build context (editor, dev build, non-dev build).

4.14 Callback

Represents a callback that can be run on Phase change, or right before loading screen hide (or when it would, if it was enabled).

Public Types

• enum When { Before , After }

Specifies when to run the callback on a given Phase.

Public Member Functions

· Callback Do (Action action)

Performs the specified callback.

Callback **Do** (Func< IEnumerator > enumerator)

Performs the specified callback.

Callback **Do** (GlobalCoroutine coroutine)

Performs the specified callback.

• Callback Do (float delay)

Performs the specified callback.

Static Public Member Functions

static implicit operator Callback (Action action)

Converts to Callback.

• static implicit **operator Callback** (Func< IEnumerator > enumerator)

Converts to Callback.

static implicit operator Callback (GlobalCoroutine coroutine)

Converts to Callback.

• static Callback AfterLoadingScreenOpen ()

Runs a callback after loading screen would have opened, if one was specified.

• static Callback BeforeLoadingScreenClose ()

Runs a callback before loading screen would close, if one was opened.

• static Callback Before (Phase phase, Scene on)

Runs a callback before the specified phase, when processing the specified scene.

· static Callback After (Phase phase, Scene on)

Runs a callback after the specified phase.

static Callback Before (Phase phase)

Runs a callback before the specified phase, when processing the specified scene.

static Callback After (Phase phase)

Runs a callback after the specified phase.

4.14 Callback 35

Properties

• Phase? phase [get]

Specifies on what phase this callback should run at.

• When when [get]

Specifies when to run the callback on a given Phase.

• Action action [get]

The Action to run.

• Func< |Enumerator > enumerator [get]

The IEnumerator coroutine to run.

• GlobalCoroutine coroutine [get]

The GlobalCoroutine to run.

• Scene scene [get]

Specifies the scene that this callback should run on.

4.14.1 Detailed Description

Represents a callback that can be run on Phase change, or right before loading screen hide (or when it would, if it was enabled).

4.14.2 Member Enumeration Documentation

4.14.2.1 When

```
enum When
```

Specifies when to run the callback on a given Phase.

Enumerator

	Before	Run callback before any scene actions have started during a given Phase.
	After	Run callback after all scene actions have run during a given Phase.

4.14.3 Member Function Documentation

4.14.3.1 Before() [1/2]

Runs a callback before the specified phase, when processing the specified scene.

Phase will still have changed to the next though, but scene actions won't run until after callback.

4.14.3.2 Before() [2/2]

Runs a callback before the specified phase, when processing the specified scene.

Phase will still have changed to the next though, but scene actions won't run until after callback.

4.14.4 Property Documentation

4.14.4.1 scene

```
Scene scene [get]
```

Specifies the scene that this callback should run on.

Specify null to run on all scenes.

4.15 CallbackUtility

An utility class that invokes callbacks (defined in interfaces based on ISceneCallbacks), and tracks performance and provides tools for optimizing and diagnosing bottlenecks in these callbacks.

Classes

class FluentInvokeAPI

An helper class to facilitate a fluent api.

4.15.1 Detailed Description

An utility class that invokes callbacks (defined in interfaces based on ISceneCallbacks), and tracks performance and provides tools for optimizing and diagnosing bottlenecks in these callbacks.

4.16 CanvasSortOrderUtility

An utility class to manage sort order on canvases.

Static Public Member Functions

• static void Remove (Canvas canvas)

Removes this canvas from the managed list.

• static void PutOnTop (this Canvas canvas)

Sets the sort order on this canvas to be on top of all other canvases managed by CanvasSortOrderUtility.

static void PutAtBottom (this Canvas canvas)

Sets the sort order on this canvas to be on bottom of all other canvases managed by CanvasSortOrderUtility.

• static void MakeSure (this Canvas canvas, Canvas above=null, Canvas below=null)

Adds a constraint on the sort order of this Canvas based on one or two other canvases.

4.17 CoroutineUtility 37

4.16.1 Detailed Description

An utility class to manage sort order on canvases.

4.16.2 Member Function Documentation

4.16.2.1 MakeSure()

Adds a constraint on the sort order of this Canvas based on one or two other canvases.

Parameters

canvas	The canvas to constrain.
above	Makes sure that this canvas is always above this one.
below	Makes sure that this canvas is always below this one.

See parameter comments for more info.

4.17 CoroutineUtility

An utility class that helps with running coroutines detached from MonoBehaviour.

Classes

· class Events

Provides events for coroutine events.

Static Public Member Functions

• static GlobalCoroutine Timer (Action action, TimeSpan interval, [CallerFilePath] string callerFile="", [Caller← LineNumber] int callerLine=0, [CallerMemberName] string callerName="")

Runs the action every interval.

• static void **Run** (Action action, TimeSpan after, [CallerFilePath] string callerFile="", [CallerLineNumber] int callerLine=0, [CallerMemberName] string callerName="")

Runs the action after the specified time.

• static void **Run** (Action action, float? after=null, bool nextFrame=false, Func< bool > when=null, [CallerFile← Path] string callerFile="", [CallerLineNumber] int callerLine=0, [CallerMemberName] string callerName="")

Runs the action after the specified time.

• static GlobalCoroutine StartCoroutineGlobal (this MonoBehaviour _, IEnumerator coroutine, Action on ← Complete=null, string description="", [CallerFilePath] string callerFile="", [CallerLineNumber] int callerLine=0)

Runs the coroutine using CoroutineUtility, which means it won't be tied to a MonoBehaviour and will persist through scene close.

• static GlobalCoroutine StartCoroutine (this IEnumerator coroutine, Action onComplete=null, string description="", [CallerFilePath] string callerFile="", [CallerLineNumber] int callerLine=0)

• static GlobalCoroutine Chain (params Func< IEnumerator >[] coroutines)

Runs the coroutines in sequence, wrapped in a single GlobalCoroutine.

static void StopCoroutine (GlobalCoroutine coroutine)

Stops the coroutine.

• static void StopAllCoroutines ()

Stops all global coroutines.

static IEnumerator WaitAll (params IEnumerator[] coroutines)

Wait for all coroutines to complete.

 static IEnumerator WaitAII (this IEnumerable < IEnumerator > coroutines, Func < bool > isCancelled=null, string debugText=null)

Wait for all coroutines to complete.

• static IEnumerator WaitAII (params GlobalCoroutine[] coroutines)

Wait for all coroutines to complete.

• static IEnumerator WaitAll (this GlobalCoroutine[] coroutines, Func< bool > isCancelled=null)

Wait for all coroutines to complete.

4.17.1 Detailed Description

An utility class that helps with running coroutines detached from MonoBehaviour.

4.17.2 Member Function Documentation

4.17.2.1 StartCoroutineGlobal()

Runs the coroutine using CoroutineUtility, which means it won't be tied to a MonoBehaviour and will persist through scene close.

You may yield return this method.

4.17.2.2 StopAllCoroutines()

```
static void StopAllCoroutines ( ) [static]
```

Stops all global coroutines.

No effect if outside of play mode.

4.18 CrossSceneReference 39

4.17.2.3 Timer()

Runs the action every interval.

Automatically stops when Application.isPlaying changes.

4.18 CrossSceneReference

A reference to a variable that references another object in some other scene.

4.18.1 Detailed Description

A reference to a variable that references another object in some other scene.

4.19 CrossSceneReferenceUtility

An utility for saving and restoring cross-scene references.

Static Public Member Functions

• static void Initialize ()

Initializes cross-scene references, if it is enabled in settings.

• static void Initialize (bool? enabled=null)

Initializes cross-scene references, if it is enabled in settings.

 $\bullet \ \ \text{static IEnumerable} < \textbf{ResolvedCrossReference} > \textbf{GetResolvedReferences} \ ()$

Gets all references for all scenes.

static IEnumerable < ResolvedCrossReference > GetResolvedReferences (scene scene)

Gets all references for this scene.

static IEnumerable < ResolvedCrossReference > GetResolvedReferences (GameObject obj)

Gets all references for this game object.

static IEnumerable < ResolvedCrossReference > GetResolvedReferencesValue (GameObject obj)

Gets all references for this game object.

static bool CanSceneBeSaved (scene scene)

Gets if the cross-scene references can be saved.

static bool GetResolved (CrossSceneReference reference, out ResolvedCrossReference? resolved)

Get the resolve result for a cross scene reference, if it has been resolved.

static ResolvedCrossReference GetResolved (CrossSceneReference reference)

Get the resolve result for a cross scene reference, if it has been resolved.

• static void ResolveAllScenes ()

Resolves all scenes.

static IEnumerable < ResolvedCrossReference > ResolveScene (scene scene)

Resolves cross-scene references in the scene.

• static void ResetAllScenes ()

Resets all cross-scene references in all scenes.

• static void ResetScene (scene scene)

Resets all cross-scene references in scene.

static | Enumerable < CrossSceneReference > FindCrossSceneReferences (params scene[] scenes)

Finds all cross-scene references in the scenes.

4.19.1 Detailed Description

An utility for saving and restoring cross-scene references.

4.19.2 Member Function Documentation

4.19.2.1 CanSceneBeSaved()

Gets if the cross-scene references can be saved.

This would be if status: SceneStatus.Restored and no resolve errors.

4.20 DefaultScenes

Provides access to the default ASM scenes.

Inherits ScriptableObject.

Public Member Functions

 $\bullet \ \ \mathsf{IEnumerable} < \mathsf{Scene} > \mathsf{Enumerate} \ (\mathsf{bool} \ \mathsf{listNulls=false})$

Enumerates all default scenes.

4.20 DefaultScenes 41

Properties

• Scene splashASMScene [get]

Gets the default ASM splash scene.

• Scene splashFadeScene [get]

Gets the default fade splash scene.

• Scene fadeScene [get]

Gets the default fade loading scene.

• Scene progressBarScene [get]

Gets the default progress bar loading scene.

• Scene iconBounceScene [get]

Gets the default icon bounce loading scene.

• Scene pressAnyKeyScene [get]

Gets the default press any button loading scene.

• Scene quoteScene [get]

Gets the default quote loading scene.

• Scene pauseScene [get]

Gets the default pause scene.

• Scene inGameToolbarScene [get]

Gets the default in-game-toolbar scene.

4.20.1 Detailed Description

Provides access to the default ASM scenes.

4.20.2 Member Function Documentation

4.20.2.1 Enumerate()

```
IEnumerable< Scene > Enumerate (
    bool listNulls = false )
```

Enumerates all default scenes.

Parameters

listNulls | Specifies whatever null will be returned for scenes that could not be found.

4.20.3 Property Documentation

4.20.3.1 fadeScene

```
Scene fadeScene [get]
```

Gets the default fade loading scene.

May be null if scene has been removed, or is not imported.

4.20.3.2 iconBounceScene

```
Scene iconBounceScene [get]
```

Gets the default icon bounce loading scene.

May be null if scene has been removed, or is not imported.

4.20.3.3 inGameToolbarScene

```
Scene inGameToolbarScene [get]
```

Gets the default in-game-toolbar scene.

May be null if scene has been removed, or is not imported.

4.20.3.4 pauseScene

```
Scene pauseScene [get]
```

Gets the default pause scene.

May be null if scene has been removed, or is not imported.

4.20.3.5 pressAnyKeyScene

```
Scene pressAnyKeyScene [get]
```

Gets the default press any button loading scene.

May be $\verb"null"$ if scene has been removed, or is not imported.

4.20.3.6 progressBarScene

```
Scene progressBarScene [get]
```

Gets the default progress bar loading scene.

May be null if scene has been removed, or is not imported.

4.20.3.7 quoteScene

```
Scene quoteScene [get]
```

Gets the default quote loading scene.

May be null if scene has been removed, or is not imported.

4.20.3.8 splashASMScene

```
Scene splashASMScene [get]
```

Gets the default ASM splash scene.

May be null if scene has been removed, or is not imported.

4.20.3.9 splashFadeScene

```
Scene splashFadeScene [get]
```

Gets the default fade splash scene.

May be null if scene has been removed, or is not imported.

4.21 DependencylnjectionUtility

Contains utility methods for depdency injection.

Classes

• interface IInjectable

Base interface for all injectable services.

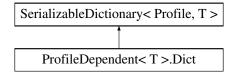
4.21.1 Detailed Description

Contains utility methods for depdency injection.

4.22 ProfileDependent< T >.Dict

A dictionary of type Profile, T.

Inheritance diagram for ProfileDependent< T >.Dict:



4.22.1 Detailed Description

A dictionary of type Profile, T.

4.23 DictionaryUtility

Contains utility functions for working with dictionaries.

Static Public Member Functions

- static void Add < TKey, TValue > (this Dictionary < TKey, TValue > d, TKey key, TValue value)
 Adds or sets the value of a key.
- $\bullet \ \ \text{static void } \textbf{Add} < \textbf{TKey, TList, TItem} > (\text{this Dictionary} < \textbf{TKey, TList} > \textbf{d}, \ \textbf{TKey key, TItem item}) \\$

Adds the value to the list with the specified key. Creates list automatically if null and adds key if necessary.

static void AddRange< TKey, TList, TItem > (this Dictionary< TKey, TList > d, TKey key, IEnumerable<
TItem > items)

Adds the values to the list with the specified key. Creates list automatically if null and adds key if necessary.

static void AddRange < TKey, TList, TItem > (this Dictionary < TKey, TList > d, TKey key, params TItem[] items)

Adds the values to the list with the specified key. Creates list automatically if null and adds key if necessary.

- static void **Remove**< **TKey, TList, TItem** > (this Dictionary< TKey, TList > d, TKey key, TItem value)

 **Removes the value to the list with the specified key.
- static TValue GetValue < TKey, TValue > (this Dictionary < TKey, TValue > d, TKey key, TValue default ← Value=default)

Gets the value of the specified key, returns default if it does not exist.

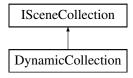
4.23.1 Detailed Description

Contains utility functions for working with dictionaries.

4.24 DynamicCollection

Represents a collection that can take a path and then gather all scenes within, guaranteeing that they are all added to build, including non-imported and blacklisted scenes.

Inheritance diagram for DynamicCollection:



Public Member Functions

bool Contains (string path)

Gets if the specified SceneAsset path is tracked by this dynamic collection.

Properties

• string id [get]

Gets the id of this collection.

• Profile profile [get]

Finds the profile associated with this dynamic collection.

• string path [get, set]

Specifies the path that this dynamic collection will gather scenes from.

• string title [get, set]

Gets the title of this collection.

• string description [get, set]

Gets the description of this collection.

IEnumerable < string > scenePaths [get]

Gets the scenes of this collection.

Scene this[int index] [get]

Gets the scene at the specified index.

int count [get]

Gets the scene count of this collection.

Properties inherited from ISceneCollection

4.24.1 Detailed Description

Represents a collection that can take a path and then gather all scenes within, guaranteeing that they are all added to build, including non-imported and blacklisted scenes.

Represents a dynamic scene collection.

4.25 CoroutineUtility.Events

Provides events for coroutine events.

Public Member Functions

delegate void CoroutineEvent (GlobalCoroutine coroutine)

Parameters

coroutine The coroutine that this event was called for.

- delegate object CoroutineFrameStartEvent (GlobalCoroutine coroutine, object data, int level, object parent
 UserData, bool isPause)
- delegate void CoroutineFrameEndEvent (GlobalCoroutine coroutine, object userData)

Static Public Attributes

static CoroutineEvent onCreated

Occurs when created. Note that GlobalCoroutine is pooled, the same object instance will be used multiple times, and this event is called when the pooled instance is 'constructed', meaning this event will be called multiple times for the same object instance.

static CoroutineEvent onDestroyed

Occurs when a GlobalCoroutine is 'destroyed'. Note that GlobalCoroutine is pooled, the same object instance will be used multiple times, and this event is called when the pooled instance is 'destroyed', meaning this event will be called multiple times for the same object instance.

static CoroutineEvent onCoroutineStarted

Occurs when a GlobalCoroutine is started.

static CoroutineEvent onCoroutineEnded

Occurs when a GlobalCoroutine is ended.

static CoroutineFrameStartEvent onSubroutineStart

Occurs before a subroutine in an executing GlobalCoroutine is started.

static CoroutineFrameEndEvent onSubroutineEnd

Occurs when a subroutine in an executing GlobalCoroutine has ended.

Properties

• static bool enableEvents [get]

Enables or disables events. Setter not available, and getter always returns false, in build. Default is false.

4.25.1 Detailed Description

Provides events for coroutine events.

4.25.2 Member Function Documentation

4.25.2.1 CoroutineFrameEndEvent()

Parameters

coroutine	The coroutine that this event was called for.
userData	The userdata that was passed to onSubroutineStart.

4.25.2.2 CoroutineFrameStartEvent()

Parameters

coroutine	The coroutine that this event was called for.
data	The object returned from IEnumerator.Current.
level	The level, or depth, of the current subroutine.
parentUserData	The userdata of the subroutine above this one, depth-wise.
isPause	GlobalCoroutine.Pause is reported as a subroutine, this is true when that is the case.

4.25.3 Member Data Documentation

4.25.3.1 onSubroutineStart

CoroutineFrameStartEvent onSubroutineStart [static]

Occurs before a subroutine in an executing GlobalCoroutine is started.

A user object can be returned, which is then passed to onSubroutineEnd.

4.26 FallbackSceneUtility

An utility class that manages the default scene, called 'AdvancedSceneManager'.

Static Public Member Functions

• static bool IsFallbackScene (scene scene)

Gets whatever the specified scene is the default scene.

4.26.1 Detailed Description

An utility class that manages the default scene, called 'AdvancedSceneManager'.

The default scene allows us to more easily close all scenes when we need to, since unity requires at least one scene to be open at any time.

4.27 CallbackUtility.FluentInvokeAPI< T>

An helper class to facilitate a fluent api.

Public Member Functions

FluentInvokeAPI< T > WithCallback (Callback callback)

Specify a callback, this should point to the interface method which provides a IEnumerator.

• FluentInvokeAPI< T > WithParam (object param)

Specify a parameter to use when invoking the callback.

• IEnumerator On (SceneCollection collection, params Scene[] additionalScenes)

Specify the collection scenes to run this callback on and start execution.

• IEnumerator OnAllOpenScenes ()

Specify the collection scenes to run this callback on and start execution..

• IEnumerator On (params Scene[] scenes)

Specify the scenes to run this callback on and start execution.

• IEnumerator **On** (params ScriptableObject[] scriptableObjects)

Specify the scenes to run this callback on and start execution.

Properties

• bool hasDefaultCallback [get]

Gets whatever T has a default callback. All callbacks inheriting from ISceneCallbacks should have one.

4.27.1 Detailed Description

An helper class to facilitate a fluent api.

Usage: Invoke<T>

4.27.2 Member Function Documentation

4.27.2.1 WithCallback()

Specify a callback, this should point to the interface method which provides a IEnumerator.

This is not needed for callback interfaces inheriting from ISceneCallbacks.

4.28 GlobalCoroutine

Represents a IEnumerator coroutine started using CoroutineUtility.

Inherits CustomYieldInstruction.

4.29 GuidReference 49

Public Member Functions

· void Pause ()

Pauses the coroutine. Make sure to not use this from within a coroutine, unless you also make sure to unpause it from outside. No effect if already paused.

• void Resume ()

Resumes a paused coroutine. No effect if not paused.

· void Stop ()

Stops the coroutine.

• override string ToString ()

Public Attributes

· MethodBase method

Gets the caller info of this coroutine.

Properties

• Action onComplete [get]

The callback that is executed when coroutine is finished.

• bool isPaused [get]

Gets whatever this coroutine is paused.

bool isComplete [get]

Gets whatever this coroutine is completed.

• bool isRunning [get]

Gets whatever this coroutine is currently running. This will still return true when paused.

• bool wasCancelled [get]

Gets whatever this coroutine was cancelled.

• string description [get, set]

Gets the user defined message that was associated with this coroutine.

• override bool keepWaiting [get]

CustomYieldInstruction.keepWaiting, this is how unity knows if this coroutine is done or not.

4.28.1 Detailed Description

Represents a IEnumerator coroutine started using CoroutineUtility.

4.29 GuidReference

Represents a persistent reference to the GameObject that this is attached to, see also GuidReferenceUtility .

Inherits MonoBehaviour.

4.29.1 Detailed Description

Represents a persistent reference to the GameObject that this is attached to, see also GuidReferenceUtility .

4.30 GuidReferenceUtility

An utility for referencing objects globally.

Static Public Member Functions

• static bool **IsRegistered** (GuidReference reference)

Gets if reference exists.

• static bool HasReference (string id)

Gets if reference exists.

• static bool TryFind (string id, out GuidReference obj)

Tries to find the reference.

• static GuidReference Find (string id)

Finds a reference if it exists.

• static string GetOrAddPersistent (GameObject obj)

Adds a persistent reference to this GameObject.

• static string GenerateID ()

Generates an id.

4.30.1 Detailed Description

An utility for referencing objects globally.

4.30.2 Member Function Documentation

4.30.2.1 GenerateID()

```
static string GenerateID ( ) [static]
```

Generates an id.

Uses https://blog.codinghorror.com/equipping-our-ascii-armor.

4.30.2.2 GetOrAddPersistent()

```
static string GetOrAddPersistent ( {\tt GameObject}\ obj\ )\ \ [{\tt static}]
```

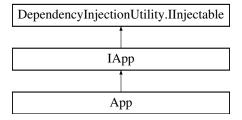
Adds a persistent reference to this GameObject.

Can only add in editor, returns null otherwise.

4.31 IApp 51

4.31 IApp

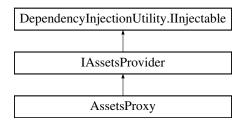
Inheritance diagram for IApp:



4.31.1 Detailed Description

4.32 IAssetsProvider

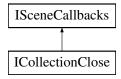
Inheritance diagram for IAssetsProvider:



4.32.1 Detailed Description

4.33 ICollectionClose

Inheritance diagram for ICollectionClose:



Public Member Functions

• void OnCollectionClose (SceneCollection collection)

4.33.1 Detailed Description

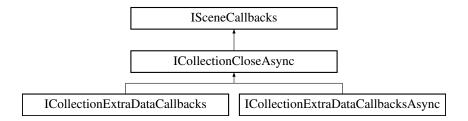
Callback for when a scene in a collection that a MonoBehaviour is contained within is closed.

Called after loading screen has opened, if one is defined, or else just before collection is closed.

See also: ICollectionCloseAsync.

4.34 ICollectionCloseAsync

Inheritance diagram for ICollectionCloseAsync:



Public Member Functions

IEnumerator OnCollectionClose (SceneCollection collection)

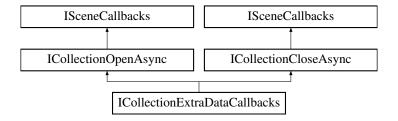
4.34.1 Detailed Description

Scene operation will wait for coroutine callback before continuing.

4.35 ICollectionExtraDataCallbacks

Callbacks for a ScriptableObject that has been set as extra data for a collection.

Inheritance diagram for ICollectionExtraDataCallbacks:



Additional Inherited Members

Public Member Functions inherited from ICollectionOpenAsync

• IEnumerator OnCollectionOpen (SceneCollection collection)

Public Member Functions inherited from ICollectionCloseAsync

• IEnumerator OnCollectionClose (SceneCollection collection)

4.35.1 Detailed Description

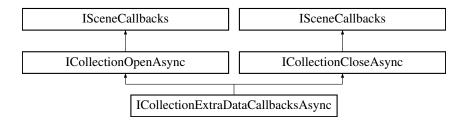
Callbacks for a ScriptableObject that has been set as extra data for a collection.

See also: ICollectionExtraDataCallbacksAsync.

4.36 ICollectionExtraDataCallbacksAsync

Callbacks for a ScriptableObject that has been set as extra data for a collection.

Inheritance diagram for ICollectionExtraDataCallbacksAsync:



Additional Inherited Members

Public Member Functions inherited from ICollectionOpenAsync

• IEnumerator OnCollectionOpen (SceneCollection collection)

Public Member Functions inherited from ICollectionCloseAsync

• IEnumerator OnCollectionClose (SceneCollection collection)

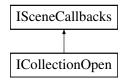
4.36.1 Detailed Description

Callbacks for a ScriptableObject that has been set as extra data for a collection.

Scene operation will wait for coroutine callback before continuing.

4.37 ICollectionOpen

Inheritance diagram for ICollectionOpen:



Public Member Functions

• void OnCollectionOpen (SceneCollection collection)

4.37.1 Detailed Description

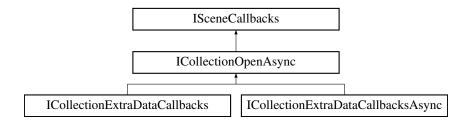
Callback for when a scene in a collection that a MonoBehaviour is contained within is opened.

Called before loading screen is hidden, if one is defined, or else just when collection has opened.

See also: ICollectionOpenAsync.

4.38 ICollectionOpenAsync

Inheritance diagram for ICollectionOpenAsync:



Public Member Functions

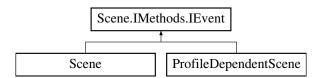
• IEnumerator OnCollectionOpen (SceneCollection collection)

4.38.1 Detailed Description

Scene operation will wait for coroutine callback before continuing.

4.39 Scene.IMethods.IEvent

Inheritance diagram for Scene.IMethods.IEvent:



Public Member Functions

• void _Open ()

Event method. Its meant for UnityEngine.Events.UnityEvent.

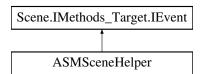
- void _ToggleOpen ()
- · void Close ()
- · void _Preload ()
- void _FinishPreload ()
- void DiscardPreload ()
- void _OpenWithLoadingScreen (Scene loadingScene)
- void _SetActive ()
- void _OpenAndActivate ()

4.39.1 Detailed Description

Specifies methods to be used in UnityEvent, using the scene itself.

4.40 Scene.lMethods_Target.lEvent

Inheritance diagram for Scene.IMethods_Target.IEvent:



Public Member Functions

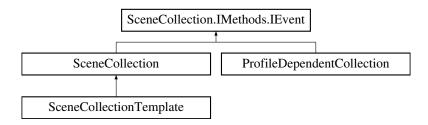
- void **Open** (Scene scene)
- void _ToggleOpen (Scene scene)
- void _Close (Scene scene)
- void _Preload (Scene scene)
- void _FinishPreload (Scene scene)
- void _DiscardPreload (Scene scene)
- void _SetActive (Scene scene)
- void _OpenAndActivate (Scene scene)

4.40.1 Detailed Description

Specifies methods to be used in UnityEvent, when not using scene itself.

4.41 SceneCollection.IMethods.IEvent

Inheritance diagram for SceneCollection.IMethods.IEvent:



Public Member Functions

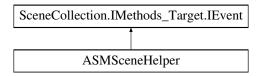
- void _Open (bool openAll=false)
- void _OpenAdditive (bool openAll=false)
- void _ToggleOpen ()
- void _ToggleOpen (bool openAll=false)
- · void Close ()

4.41.1 Detailed Description

Specifies methods to be used in UnityEvent, using the collection itself.

4.42 SceneCollection.IMethods_Target.IEvent

Inheritance diagram for SceneCollection.IMethods_Target.IEvent:



Public Member Functions

- void _Open (SceneCollection collection)
- void _OpenAdditive (SceneCollection collection)
- void _ToggleOpen (SceneCollection collection)
- void _Close (SceneCollection collection)

4.42.1 Detailed Description

Specifies methods to be used in UnityEvent, when not using collection itself.

4.43 IFadeLoadingScreen

Used to pass arguments from LoadingScreenUtility.FadeIn(LoadingScreen, float, Color?)

Properties

float fadeDuration [get, set]

Specifies the fade duration.

• Color color [get, set]

Specifies the color of the fade.

4.43.1 Detailed Description

Used to pass arguments from LoadingScreenUtility.FadeIn(LoadingScreen, float, Color?)

4.44 DependencylnjectionUtility.llnjectable

Base interface for all injectable services.

Inheritance diagram for DependencyInjectionUtility.IInjectable:



4.44.1 Detailed Description

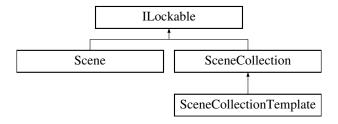
Base interface for all injectable services.

Injectable classes implementing an interface based on IInjectable is automatically registered in Dependency \leftarrow Injection Utility static constructor.

4.45 ILockable

Specifies a object that can be locked, using LockUtility.

Inheritance diagram for ILockable:



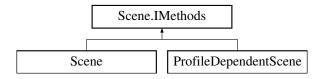
4.45.1 Detailed Description

Specifies a object that can be locked, using LockUtility.

Available, but no effect in build.

4.46 Scene.IMethods

Inheritance diagram for Scene.IMethods:



Classes

interface IEvent

Public Member Functions

• SceneOperation Open ()

Opens the scene.

• SceneOperation ToggleOpen ()

Toggles this scene open or closed.

• SceneOperation Close ()

Closes the scene.

• SceneOperation Preload (Action onPreloaded=null)

Preloads the scene, to be displayed at a later time. See also: FinishPreload, DiscardPreload.

SceneOperation FinishPreload ()

Finishes preloading the scene, displaying it.

• SceneOperation DiscardPreload ()

Discards the scene, if preloaded.

• SceneOperation OpenWithLoadingScreen (Scene loadingScene)

Opens the scene while a loading screen is open.

• SceneOperation OpenAndActivate ()

Opens the scene and sets it as active.

· void SetActive ()

Sets the scene as active in heirarchy.

4.46.1 Detailed Description

Specified methods to be used programmatically, on the scene itself.

4.46.2 Member Function Documentation

4.46.2.1 Close()

```
SceneOperation Close ( )
```

Closes the scene.

No effect if scene is already closed.

Implemented in ProfileDependentScene, and Scene.

4.46.2.2 FinishPreload()

```
SceneOperation FinishPreload ( )
```

Finishes preloading the scene, displaying it.

Scene must be preloaded beforehand.

Implemented in ProfileDependentScene, and Scene.

4.46.2.3 Open()

```
SceneOperation Open ( )
```

Opens the scene.

No effect if scene is already open.

Implemented in ProfileDependentScene, and Scene.

4.46.2.4 Preload()

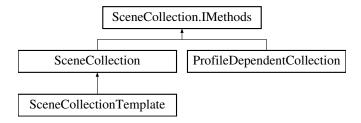
Preloads the scene, to be displayed at a later time. See also: FinishPreload, DiscardPreload.

Scene must be closed beforehand.

Implemented in ProfileDependentScene, and Scene.

4.47 SceneCollection.IMethods

Inheritance diagram for SceneCollection.IMethods:



Classes

interface IEvent

Public Member Functions

• SceneOperation Open (bool openAll=false)

Opens this collection.

• SceneOperation OpenAdditive (bool openAll=false)

Opens this collection as additive.

• SceneOperation ToggleOpen (bool openAll=false)

Toggles this collection open or closed.

• SceneOperation Close ()

Closes this collection.

4.47.1 Detailed Description

Specified methods to be used programmatically, on the collection itself.

4.47.2 Member Function Documentation

4.47.2.1 Close()

```
SceneOperation Close ()
```

Closes this collection.

No effect if collection is already closed. Note that "additive collections" are not actually opened, only its scenes are.

Implemented in SceneCollection, and ProfileDependentCollection.

4.47.2.2 Open()

```
SceneOperation Open (
          bool openAll = false )
```

Opens this collection.

Parameters

openAll Specifies whatever scenes flagged to not open with collection, should.

Reopens all non-persistent scenes.

Implemented in SceneCollection, and ProfileDependentCollection.

4.47.2.3 OpenAdditive()

```
SceneOperation OpenAdditive (
          bool openAll = false )
```

Opens this collection as additive.

Parameters

openAll	Specifies whatever scenes flagged to not open with collection, should.
---------	--

Additive collections are not "opened", all scenes within are merely opened like normal scenes. Mostly intended for convenience.

Implemented in SceneCollection, and ProfileDependentCollection.

4.47.2.4 ToggleOpen()

```
SceneOperation ToggleOpen (
          bool openAll = false )
```

Toggles this collection open or closed.

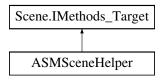
Parameters

openState	Specifies whatever you have a preferred state to toggle to, this means collection will not be closed
	if true is passed. This can be used to ensure collection is open, without having an explicit check
	beforehand. The inverse is also the case for false.
openAll	Specifies whatever scenes flagged to not open with collection, should.

Implemented in SceneCollection, and ProfileDependentCollection.

4.48 Scene.IMethods_Target

Inheritance diagram for Scene.IMethods_Target:



Classes

interface IEvent

Public Member Functions

SceneOperation Open (Scene scene)

Opens the specified scene.

• SceneOperation ToggleOpen (Scene scene)

Toggles the open state of the specified scene.

• SceneOperation Close (Scene scene)

Closes the specified scene.

• SceneOperation Preload (Scene scene, Action onPreloaded=null)

Preloads the specified scene, to be displayed at a later time. See also: FinishPreload(Scene), DiscardPreload(← Scene).

SceneOperation FinishPreload (Scene scene)

Finishes preloading the specified scene, displaying it.

• SceneOperation **DiscardPreload** (Scene scene)

Discards the specified scene, if preloaded.

• SceneOperation OpenWithLoadingScreen (Scene scene, Scene loadingScene)

Opens the specified scene while a loading screen is open.

SceneOperation OpenAndActivate (Scene scene)

Opens the scene and activates it.

• void **SetActive** (Scene scene)

Sets the specified scene as active in heirarchy.

4.48.1 Detailed Description

Specifies methods to be used programmatically, using scene as first parameter.

4.48.2 Member Function Documentation

4.48.2.1 Close()

```
SceneOperation Close (
Scene scene)
```

Closes the specified scene.

Already closed scenes not affected.

Implemented in ASMSceneHelper.

4.48.2.2 FinishPreload()

Finishes preloading the specified scene, displaying it.

Scene must be preloaded beforehand.

Implemented in ASMSceneHelper.

4.48.2.3 Open()

```
SceneOperation Open (
Scene scene)
```

Opens the specified scene.

Already open scenes not affected.

Implemented in ASMSceneHelper.

4.48.2.4 Preload()

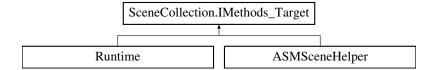
Preloads the specified scene, to be displayed at a later time. See also: FinishPreload(Scene), DiscardPreload(\leftarrow Scene).

Scene must be closed beforehand.

Implemented in ASMSceneHelper.

4.49 SceneCollection.IMethods_Target

Inheritance diagram for SceneCollection.IMethods_Target:



Classes

• interface IEvent

Public Member Functions

- SceneOperation Open (SceneCollection collection, bool openAll=false)
- SceneOperation OpenAdditive (SceneCollection collection, bool openAll=false)
- SceneOperation ToggleOpen (SceneCollection collection, bool openAll=false)
- SceneOperation Close (SceneCollection collection)

4.49.1 Detailed Description

Specifies methods to be used programmatically, using collection as first parameter.

4.50 InitializeInEditorAttribute

Initializes a class in editor on recompile.

Inherits Attribute.

4.50.1 Detailed Description

Initializes a class in editor on recompile.

Available in build, but no effect.

4.51 InitializeInEditorMethodAttribute

Initializes a class in editor on recompile.

Inherits Attribute.

4.51.1 Detailed Description

Initializes a class in editor on recompile.

Available in build, but no effect.

4.52 InputBinding

Represents a input binding for InputSystem. Available even when InputSystem is uninstalled.

Properties

```
• List < InputButton > buttons [get]
```

Specifies the buttons.

• bool openCollectionAsAdditive [get, set]

Specifies whatever collection should be opened as a collection.

• InputBindingInteractionType interactionType [get, set]

Specifies the interaction type.

4.52.1 Detailed Description

Represents a input binding for InputSystem. Available even when InputSystem is uninstalled.

4.53 InputButton 65

4.53 InputButton

Specifies a input binding for use with InputSystem.

Public Attributes

• string name

Specifies the name of this binding.

• string path

Specifies the path of this binding.

Properties

• readonly bool isValid [get]

Gets if this binding is valid.

4.53.1 Detailed Description

Specifies a input binding for use with InputSystem.

4.53.2 Member Data Documentation

4.53.2.1 path

string path

Specifies the path of this binding.

This would be UnityEngine.InputSystem.InputBinding.path.

4.54 IProfileManager

Manages the current profile.

Inheritance diagram for IProfileManager:



Properties

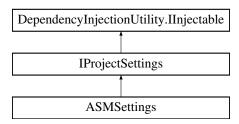
• Profile current [get]

4.54.1 Detailed Description

Manages the current profile.

4.55 IProjectSettings

Inheritance diagram for IProjectSettings:



Public Member Functions

- · void Save ()
- · void SaveNow ()

Properties

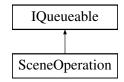
- bool allowCollectionLocking [get, set]
- bool allowExcludingCollectionsFromBuild [get, set]
- bool allowSceneLocking [get, set]
- string assetPath [get, set]
- Profile buildProfile [get]
- Color buildUnitySplashScreenColor [get]
- bool checkForDuplicateSceneOperations [get, set]
- ASMSettings.CustomData customData [get]
- Profile defaultProfile [get, set]
- bool enableCrossSceneReferences [get, set]
- Scene fadeScene [get, set]
- Profile forceProfile [get, set]
- bool preventSpammingEventMethods [get, set]
- bool reverseUnloadOrderOnCollectionClose [get, set]
- SceneImportOption sceneImportOption [get, set]
- float spamCheckCooldown [get, set]

4.55.1 Detailed Description

4.56 | Queueable

Represents a queueable item.

Inheritance diagram for IQueueable:



4.57 IRuntime 67

Public Member Functions

• void OnTurn (Action onComplete)

Called when it is this queueables turn.

· void OnCancel ()

Called when queueable is cancelled.

• bool CanQueue ()

Called to make sure the item can actually be queued.

4.56.1 Detailed Description

Represents a queueable item.

See also QueueUtility<T>.

4.56.2 Member Function Documentation

4.56.2.1 OnTurn()

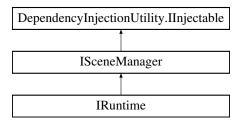
Called when it is this queueables turn.

Parameters

onComplete	Must be called when operation is done, otherwise queue will be stuck.
------------	---

4.57 IRuntime

Inheritance diagram for IRuntime:



Additional Inherited Members

Public Member Functions inherited from ISceneManager

• void AddSceneLoader< T > ()

- SceneOperation Close (IEnumerable < Scene > scenes)
- SceneOperation Close (params Scene[] scenes)
- SceneOperation Close (Scene scene)
- SceneOperation Close (SceneCollection collection)
- SceneOperation CloseAll (bool exceptLoadingScreens=true, bool exceptUnimported=true, params Scene[] except)
- SceneOperation DiscardPreload ()
- SceneOperation DiscardPreload (Scene scene)
- SceneOperation FinishPreload ()
- SceneOperation FinishPreload (Scene scene)
- SceneLoader GetLoaderForScene (Scene scene)
- SceneState GetState (Scene scene)
- IEnumerable < SceneLoader > GetToggleableSceneLoaders ()
- bool **IsTracked** (Scene scene)
- bool IsTracked (SceneCollection collection)
- SceneOperation Open (IEnumerable < Scene > scenes)
- SceneOperation Open (params Scene[] scenes)
- SceneOperation Open (Scene scene)
- SceneOperation Open (SceneCollection collection, bool openAll=false)
- SceneOperation OpenAdditive (IEnumerable< SceneCollection > collections, SceneCollection active
 — Collection=null, Scene loadingScene=null)
- SceneOperation OpenAdditive (SceneCollection collection, bool openAll=false)
- SceneOperation OpenAndActivate (Scene scene)
- SceneOperation OpenWithLoadingScreen (IEnumerable < Scene > scene, Scene loadingScreen)
- SceneOperation OpenWithLoadingScreen (Scene scene, Scene loadingScreen)
- SceneOperation Preload (Scene scene, Action onPreloaded=null)
- void RemoveSceneLoader< T > ()
- void **SetActive** (Scene scene)
- SceneOperation ToggleOpen (Scene scene)
- SceneOperation ToggleOpen (SceneCollection collection, bool openAll=false)
- void Track (Scene scene)
- void Track (Scene scene, UnityEngine.SceneManagement.Scene unityScene)
- void Track (SceneCollection collection, bool isAdditive=false)
- bool **Untrack** (Scene scene)
- void Untrack (SceneCollection collection, bool isAdditive=false)
- void UntrackCollections ()
- void UntrackScenes ()

Properties inherited from ISceneManager

- Scene activeScene [get]
- SceneOperation currentOperation [get]
- Scene dontDestroyOnLoad [get]
- bool isBusy [get]
- IEnumerable < SceneCollection > openAdditiveCollections [get]
- SceneCollection openCollection [get]
- IEnumerable < Scene > openScenes [get]
- Scene preloadedScene [get]
- IEnumerable < SceneOperation > queuedOperations [get]
- IEnumerable < SceneOperation > runningOperations [get]

4.58 ISceneCallbacks 69

Events inherited from ISceneManager

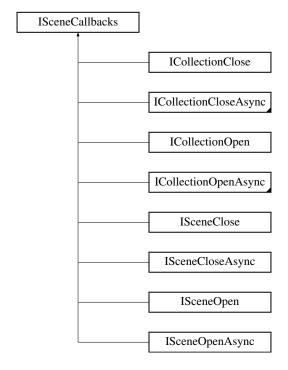
- Action < SceneCollection > collectionClosed
- Action < SceneCollection > collectionOpened
- Action < Scene > sceneClosed
- Action< Scene > sceneOpened
- Action < Scene > scenePreloaded
- Action < Scene > scenePreloadFinished
- Action startedWorking
- · Action stoppedWorking

4.57.1 Detailed Description

4.58 ISceneCallbacks

Base interface for ISceneOpen, ISceneClose, ICollectionOpen, ICollectionClose. Does nothing on its own, used by CallbackUtility.

Inheritance diagram for ISceneCallbacks:



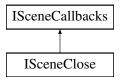
4.58.1 Detailed Description

Base interface for ISceneOpen, ISceneClose, ICollectionOpen, ICollectionClose. Does nothing on its own, used by CallbackUtility.

4.59 ISceneClose

Callback for when the scene that a MonoBehaviour is contained within is closed.

Inheritance diagram for ISceneClose:



Public Member Functions

• void OnSceneClose ()

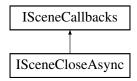
4.59.1 Detailed Description

Callback for when the scene that a MonoBehaviour is contained within is closed.

See also: ISceneCloseAsync.

4.60 ISceneCloseAsync

Inheritance diagram for ISceneCloseAsync:



Public Member Functions

• IEnumerator OnSceneClose ()

4.60.1 Detailed Description

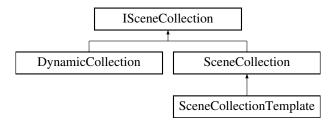
Scene operation will wait for coroutine callback before continuing.

4.61 ISceneCollection 71

4.61 ISceneCollection

Represents the core variables of what makes up a scene collection.

Inheritance diagram for ISceneCollection:



Properties

```
• IEnumerable < Scene > scenes [get]
```

Gets the scenes of this collection.

• IEnumerable < string > scenePaths [get]

Gets the scenes of this collection.

• string title [get]

Gets the title of this collection.

• string description [get]

Gets the description of this collection.

• int count [get]

Gets the scene count of this collection.

• string id [get]

Gets the id of this collection.

• Scene this[int index] [get]

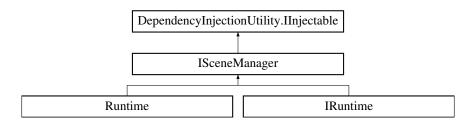
Gets the scene at the specified index.

4.61.1 Detailed Description

Represents the core variables of what makes up a scene collection.

4.62 ISceneManager

Inheritance diagram for ISceneManager:



Public Member Functions

- void AddSceneLoader< T > ()
- SceneOperation Close (IEnumerable < Scene > scenes)
- SceneOperation Close (params Scene[] scenes)
- SceneOperation Close (Scene scene)
- SceneOperation Close (SceneCollection collection)
- SceneOperation CloseAll (bool exceptLoadingScreens=true, bool exceptUnimported=true, params Scene[] except)
- SceneOperation DiscardPreload ()
- SceneOperation DiscardPreload (Scene scene)
- SceneOperation FinishPreload ()
- SceneOperation FinishPreload (Scene scene)
- SceneLoader GetLoaderForScene (Scene scene)
- SceneState GetState (Scene scene)
- IEnumerable < SceneLoader > GetToggleableSceneLoaders ()
- bool **IsTracked** (Scene scene)
- bool IsTracked (SceneCollection collection)
- SceneOperation Open (IEnumerable < Scene > scenes)
- SceneOperation Open (params Scene[] scenes)
- SceneOperation Open (Scene scene)
- SceneOperation Open (SceneCollection collection, bool openAll=false)
- SceneOperation OpenAdditive (IEnumerable< SceneCollection > collections, SceneCollection active
 — Collection=null, Scene loadingScene=null)
- SceneOperation OpenAdditive (SceneCollection collection, bool openAll=false)
- SceneOperation OpenAndActivate (Scene scene)
- SceneOperation OpenWithLoadingScreen (IEnumerable < Scene > scene, Scene loadingScreen)
- SceneOperation OpenWithLoadingScreen (Scene scene, Scene loadingScreen)
- SceneOperation Preload (Scene scene, Action onPreloaded=null)
- void RemoveSceneLoader< T > ()
- void **SetActive** (Scene scene)
- SceneOperation ToggleOpen (Scene scene)
- SceneOperation ToggleOpen (SceneCollection collection, bool openAll=false)
- void Track (Scene scene)
- void **Track** (Scene scene, UnityEngine.SceneManagement.Scene unityScene)
- void Track (SceneCollection collection, bool isAdditive=false)
- bool **Untrack** (Scene scene)
- void **Untrack** (SceneCollection collection, bool isAdditive=false)
- void UntrackCollections ()
- void UntrackScenes ()

Properties

- Scene activeScene [get]
- SceneOperation currentOperation [get]
- Scene dontDestroyOnLoad [get]
- bool isBusy [get]
- IEnumerable < SceneCollection > openAdditiveCollections [get]
- SceneCollection openCollection [get]
- IEnumerable < Scene > openScenes [get]
- Scene preloadedScene [get]
- IEnumerable < SceneOperation > queuedOperations [get]
- IEnumerable < SceneOperation > runningOperations [get]

4.63 ISceneOpen 73

Events

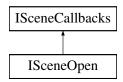
- Action< SceneCollection > collectionClosed
- Action< SceneCollection > collectionOpened
- Action < Scene > sceneClosed
- Action< Scene > sceneOpened
- Action < Scene > scenePreloaded
- Action < Scene > scenePreloadFinished
- Action startedWorking
- · Action stoppedWorking

4.62.1 Detailed Description

4.63 ISceneOpen

Callback for when the scene that a MonoBehaviour is contained within is opened.

Inheritance diagram for ISceneOpen:



Public Member Functions

void OnSceneOpen ()

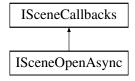
4.63.1 Detailed Description

Callback for when the scene that a MonoBehaviour is contained within is opened.

See also: ISceneOpenAsync.

4.64 ISceneOpenAsync

Inheritance diagram for ISceneOpenAsync:



Public Member Functions

• IEnumerator OnSceneOpen ()

4.64.1 Detailed Description

Scene operation will wait for coroutine callback before continuing.

4.65 LerpUtility

Provides some convinience functions for lerping.

Static Public Member Functions

• static IEnumerator Lerp (float start, float end, float duration, Action< float > callback, Action on ← Complete=null)

Lerp from start to end over duration seconds.

- static IEnumerator Lerp (Vector3 start, Vector3 end, float duration, Action< Vector3 > callback, Action on ←
 Complete=null)
- static IEnumerator Lerp (Vector2 start, Vector2 end, float duration, Action < Vector2 > callback, Action on ← Complete=null)

4.65.1 Detailed Description

Provides some convinience functions for lerping.

4.65.2 Member Function Documentation

4.65.2.1 Lerp()

Lerp from *start* to *end* over *duration* seconds.

Parameters

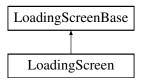
start	The start value.
end	The end value.
duration	The duration in seconds to lerp for.
callback	The callback each lerp interval.
onComplete	Callback when complete.

4.66 LoadingScreen 75

4.66 LoadingScreen

A class that contains callbacks for loading screens.

Inheritance diagram for LoadingScreen:



Public Member Functions

• override IEnumerator OnOpen ()

Called when loading scene is opened.

• override IEnumerator OnClose ()

Called when loading scene is closed.

Public Member Functions inherited from LoadingScreenBase

• IEnumerator OnOpen ()

Called when the loading screen is opened.

• IEnumerator OnClose ()

Called when the loading screen is about to close.

virtual void OnProgressChanged (float progress)

Called when progress has changed.

• bool HasPressedAnyKey ()

Gets if any key has been pressed this frame.

WaitUntil WaitForAnyKey ()

Returns WaitUntil that waits for user to press any key.

Properties

SceneOperation operation [get, set]

The current scene operation that this loading screen is associated with. May be null for the first few frames, before loading has actually begun.

Properties inherited from LoadingScreenBase

• bool isOpening [get]

Gets whatever we're currently opening.

• bool isOpen [get]

Gets whatever we're currently open.

bool isClosing [get]

Gets whatever we're currently closing.

Additional Inherited Members

Public Attributes inherited from LoadingScreenBase

Action < LoadingScreenBase > onDestroy
 Occurs when loading screen is destroyed.

· Canvas canvas

4.66.1 Detailed Description

A class that contains callbacks for loading screens.

SplashScreen and LoadingScreen cannot co-exist within the same scene.

4.66.2 Member Function Documentation

4.66.2.1 OnClose()

```
override IEnumerator OnClose ( ) [abstract]
```

Called when loading scene is closed.

Use this callback to hide your loading screen, the scene manager will wait until its done.

4.66.2.2 OnOpen()

```
override IEnumerator OnOpen ( ) [abstract]
```

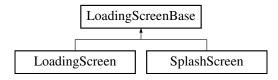
Called when loading scene is opened.

Use this callback to show your loading screen, the scene manager will wait until its done.

4.67 LoadingScreenBase

A generic base class for loading screens. You probably want to inherit from LoadingScreen though.

Inheritance diagram for LoadingScreenBase:



Public Member Functions

• IEnumerator OnOpen ()

Called when the loading screen is opened.

• IEnumerator OnClose ()

Called when the loading screen is about to close.

• virtual void OnProgressChanged (float progress)

Called when progress has changed.

• bool HasPressedAnyKey ()

Gets if any key has been pressed this frame.

• WaitUntil WaitForAnyKey ()

Returns WaitUntil that waits for user to press any key.

Public Attributes

Action < LoadingScreenBase > onDestroy

Occurs when loading screen is destroyed.

· Canvas canvas

Properties

```
bool isOpening [get]
```

Gets whatever we're currently opening.

• bool isOpen [get]

Gets whatever we're currently open.

bool isClosing [get]

Gets whatever we're currently closing.

4.67.1 Detailed Description

A generic base class for loading screens. You probably want to inherit from LoadingScreen though.

When multiple loading screens exist within the same scene, only the first found one will be used.

4.67.2 Member Function Documentation

4.67.2.1 HasPressedAnyKey()

```
bool HasPressedAnyKey ( )
```

Gets if any key has been pressed this frame.

Returns false if not is Open.

4.67.3 Member Data Documentation

4.67.3.1 canvas

Canvas canvas

The canvas that this loading screen uses.

This will automatically register canvas with CanvasSortOrderUtility, to automatically manage canvas sort order.

You probably want to set this through the inspector.

4.67.4 Property Documentation

4.67.4.1 isOpen

bool isOpen [get]

Gets whatever we're currently open.

This is still true is Closing is true.

4.68 LoadingScreenUtility

Manager for loading screens.

Static Public Member Functions

static bool IsLoadingScreenOpen (Scene scene)

Gets if this scene is a loading screen.

 static Async< T > OpenLoadingScreen< T > (Scene loadingScene, SceneOperation operation=null, Action< T > callbackBeforeBegin=null, Action< float > progress=null)

Shows a loading screen.

- static IEnumerator CloseLoadingScreen (Scene scene)
- static IEnumerator CloseLoadingScreen (LoadingScreenBase loadingScreen, Action < float > progress=null, bool closeScene=true)

Hide the loading screen.

static IEnumerator CloseLoadingScreenScene (Scene scene, Action < float > progress=null)

Close the scene that contained a loading screen.

static IEnumerator CloseAll ()

Hide all loading screens.

- static Async < LoadingScreen > FadeOut (float duration=1, Color? color=null, Action < float > progress=null)
 Fades out the screen.
- static IEnumerator FadeIn (LoadingScreenBase loadingScreen, float duration=1, Color? color=null, Action < float > progress=null)

Fades in the screen.

- static IEnumerator DoAction (Scene scene, Action action, Action < LoadingScreenBase > loadingScreen ← Callback=null)
- static IEnumerator DoAction (Scene scene, Func< IEnumerator > coroutine, Action< LoadingScreenBase > loadingScreenCallback=null)

Opens loading screen, performs action and hides loading screen again.

• static IEnumerator WithProgress (this AsyncOperation asyncOperation, Action < float > onProgress)

Returns a coroutine that returns when AsyncOperation.isDone becomes true. onProgress will be called every frame with AsyncOperation.progress.

static AsyncOperation Preload (this AsyncOperation asyncOperation, out Func< IEnumerator > activate ←
 Callback)

 ${\it Sets AsyncOperation. allow Scene Activation \ to \ false}.$

Properties

• static bool isAnyLoadingScreenOpen [get]

Gets if any loading screens are open.

• static Scene fade [get]

Finds the default fade loading screen. Can be set through project settings, or in scene loading section of the settings populo.

• static Scene defaultLoadingScreen [get]

Gets the current default loading screen.

• static IEnumerable < LoadingScreenBase > loadingScreens [get]

The currently open loading screens.

4.68.1 Detailed Description

Manager for loading screens.

4.68.2 Member Function Documentation

4.68.2.1 CloseLoadingScreen()

Hide the loading screen.

Parameters

loadingScreen	The loading screen to hide.	
progress	The callback to receive progress.	
closeScene	Specifies whatever the scene should be closed afterwards. Use CloseLoadingScreenScene(Scene, Action <float>) if false.</float>	

4.68.2.2 DoAction()

Opens loading screen, performs action and hides loading screen again.

Parameters

scene	The loading screen scene.
coroutine	To coroutine to execute.
loadingScreenCallback	The callback to perform when loading script is loaded, but before ASM has called LoadingScreenBase.OnOpen().

4.69 MainThreadUtility

Static Public Member Functions

```
    static T Invoke
    T > (Func
    T > func)
```

Queues the function to be run on the main thread, during the next frame.

• static void Invoke (Action action)

Queues the action to be run on the main thread, during the next frame.

• static T Invoke< T > (this Func< T > func, bool mainThread=false)

Invokes the action .

static void Invoke (this Action action, bool mainThread=false)

Invokes the action .

· static void Start ()

Starts main thread utility coroutine.

• static void Stop ()

Stops main thread utility coroutine.

Properties

• static bool isEnabled [get]

Gets whatever MainThreadUtility is enabled, set to false in source code to disable.

• static bool isOnMainThread [get]

Gets if the thread we're currently on is the main thread.

• static bool **IsRunning** [get]

Gets if main thread utility is running.

4.69.1 Detailed Description

An utility for running actions on the main thread.

Only usable in play mode.

4.69.2 Member Function Documentation

4.69.2.1 Invoke() [1/2]

Queues the action to be run on the main thread, during the next frame.

Parameters

action Th	ne action to invoke.	
-------------	----------------------	--

4.70 ObjectReference 81

4.69.2.2 Invoke() [2/2]

Invokes the action .

Parameters

action	The action to invoke.
mainThread	Queues the action to be run on the main thread, during the next frame.

4.69.2.3 Invoke < T >() [1/2]

```
static T Invoke< T > ( Func < T > func > func > [static]
```

Queues the function to be run on the main thread, during the next frame.

Parameters

```
func The action to invoke.
```

4.69.2.4 Invoke< T >() [2/2]

Invokes the action .

Parameters

func	The action to invoke.
mainThread	Queues the action to be run on the main thread, during the next frame.

4.70 ObjectReference

A reference to an object in a scene.

Inherits IEqualityComparer< ObjectReference >.

Public Member Functions

ObjectReference With (Component component)

Adds data about a component.

• ObjectReference With (int? unityEventIndex=null, int? arrayIndex=null)

Adds data about an unity event.

• bool IsValid (bool returnTrueWhenSceneIsUnloaded=false)

Returns true if the reference is still valid.

4.70.1 Detailed Description

A reference to an object in a scene.

4.71 ParallelASMCallbacks

Specifies whatever the ASM callbacks should be run in parallel for any callbacks defined in this script.

Inherits Attribute.

4.71.1 Detailed Description

Specifies whatever the ASM callbacks should be run in parallel for any callbacks defined in this script.

4.72 Profile

A profile for ASM, contains settings and collections.

Inheritance diagram for Profile:



Public Member Functions

bool IsStartupCollection (SceneCollection collection)

Gets whatever the specified collection is a startup collection.

int IndexOf (SceneCollection collection)

Gets the index of the specified collection.

int IndexOf (DynamicCollection collection)

Gets the index of the specified collection.

bool Contains (ISceneCollection collection, bool checkRemoved=false)

Gets whatever this profile contains the specified collection.

• IEnumerable < ISceneCollection > FindCollections (Scene scene)

Finds all collection that the scene is included in. Includes dynamic collections.

4.72 Profile 83

Public Member Functions inherited from ASMModel

· virtual void Save ()

Saves the singleton to disk after a delay.

void SaveNow ()

Saves the singleton to disk.

void SaveNow (bool setDirty=true)

Saves the singleton to disk.

virtual bool IsMatch (string q)

Gets if q matches name.

Static Public Member Functions

• static Profile Find (string q)

Finds the profile with the specified name or id.

static bool TryFind (string q, out Profile profile)

Finds the profile with the specified name or id.

Static Public Member Functions inherited from ASMModel

• static string GenerateID ()

Generate id.

Static Public Attributes

static readonly string AssetSearchString = "t:" + typeof(Profile).FullName
 Gets 't:AdvancedSceneManager.Models.Profile', the string to use in AssetDatabase.FindAssets(string).

Properties

• static Profile current [get]

Gets the currently active profile.

• IEnumerable < Scene > scenes [get]

Gets the scenes managed by this profile.

IEnumerable < SceneCollection > collections [get]

Gets the collections contained within this profile.

• IEnumerable < DynamicCollection > dynamicCollections [get]

Gets the dynamic collections contained within this profile.

• StandaloneCollection standaloneScenes [get]

Gets the standalone scenes contained within this profile.

IEnumerable < Scene > startupScenes [get]

Gets the scenes flagged to open on startup.

• IEnumerable < ISceneCollection > removedCollections [get]

Gets all removed collections in this profile.

• IEnumerable < ISceneCollection > allCollections [get]

Gets collections, standaloneScenes, dynamicCollections.

• IEnumerable < Scene > specialScenes [get]

Gets default loading screen, splash screen and startup loading screen.

• IEnumerable < SceneCollection > startupCollections [get]

Gets the collections that will be opened on startup.

• Scene startupScene [get, set]

The startup scene.

Scene loadingScene [get, set]

The default loading scene.

• Scene splashScene [get, set]

The splash scene.

• ThreadPriority backgroundLoadingPriority [get, set]

Application.backgroundLoadingPriority setting is not saved, and must be manually set every time build or editor starts, this property persists the value and automatically sets it during startup.

bool enableChangingBackgroundLoadingPriority [get, set]

Enable or disable ASM automatically changing Application.backgroundLoadingPriority.

• bool unloadUnusedAssetsForStandalone [get, set]

Enable or disable ASM calling Resources. Unload Unused Assets after standalone scenes has been opened or closed.

Properties inherited from ASMModel

```
• string id [get]
```

Gets the id of this ASMModel.

• new string name [get, protected set]

Additional Inherited Members

Static Protected Member Functions inherited from ASMModel

static T CreateInternal < T > (string name)

Creates a profile. Throws if name is invalid.

4.72.1 Detailed Description

A profile for ASM, contains settings and collections.

4.72.2 Property Documentation

4.72.2.1 removedCollections

```
IEnumerable<ISceneCollection> removedCollections [get]
```

Gets all removed collections in this profile.

Removed collections still exist until deleted, and may be manually opened, but they will not be listed in collections or dynamicCollections.

4.72.2.2 scenes

Gets the scenes managed by this profile.

Includes both collection and standalone scenes.

4.72.2.3 specialScenes

```
IEnumerable < Scene > specialScenes [get]
```

Gets default loading screen, splash screen and startup loading screen.

null is filtered out.

4.72.2.4 startupCollections

```
IEnumerable<SceneCollection> startupCollections [get]
```

Gets the collections that will be opened on startup.

If no collection is explicitly defined to be opened during startup, then the first available collection in list will be returned.

4.73 ProfileDependent < T >

Specifies a *T* that changes depending on active Profile.

Inherits ScriptableObject.

Classes

· class Dict

A dictionary of type Profile, T.

Public Member Functions

• bool GetModel (out T scene)

Gets the selected scene.

• T2 DoAction < T2 > (Func < T, T2 > action)

Performs an action on the scene.

void DoAction (Action< T > action)

Performs an action on the scene.

Public Attributes

• Dict list = new Dict()

The list of proxies for this ${\sf T}$.

Properties

• bool isValid [get]

Gets if the current state of this T is valid.

4.73.1 Detailed Description

Specifies a *T* that changes depending on active Profile.

Type Constraints

T: ASMModel

4.73.2 Member Function Documentation

4.73.2.1 DoAction()

```
void DoAction ( \label{eq:action} \mbox{Action} < \mbox{T} \mbox{$>$ action$ )}
```

Performs an action on the scene.

Does nothing if is Valid is false.

4.73.2.2 DoAction< T2 >()

Performs an action on the scene.

Does nothing if is Valid is false.

4.73.2.3 GetModel()

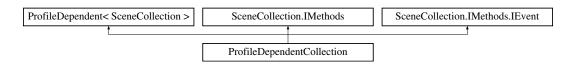
Gets the selected scene.

Returns null if scene something went wrong.

4.74 ProfileDependentCollection

Represents a SceneCollection that changes depending on active Profile.

Inheritance diagram for ProfileDependentCollection:



Public Member Functions

• SceneOperation Open (bool openAll=false)

Opens this collection.

• SceneOperation OpenAdditive (bool openAll=false)

Opens this collection as additive.

• SceneOperation ToggleOpen (bool openAll=false)

Toggles this collection open or closed.

SceneOperation Close ()

Closes this collection.

- void **Open** (bool openAll=false)
- void **_OpenAdditive** (bool openAll=false)
- void _ToggleOpen ()
- void _ToggleOpen (bool openAll)
- void Close ()

Public Member Functions inherited from ProfileDependent < SceneCollection >

• bool GetModel (out T scene)

Gets the selected scene.

• T2 DoAction < T2 > (Func < T, T2 > action)

Performs an action on the scene.

void DoAction (Action < T > action)

Performs an action on the scene.

Additional Inherited Members

Public Attributes inherited from ProfileDependent < SceneCollection >

• Dict list

The list of proxies for this T.

Properties inherited from ProfileDependent < SceneCollection >

• bool isValid [get]

Gets if the current state of this T is valid.

4.74.1 Detailed Description

Represents a SceneCollection that changes depending on active Profile.

4.74.2 Member Function Documentation

4.74.2.1 Close()

SceneOperation Close ()

Closes this collection.

No effect if collection is already closed. Note that "additive collections" are not actually opened, only its scenes are.

Implements SceneCollection.IMethods.

4.74.2.2 Open()

```
SceneOperation Open (
          bool openAll = false )
```

Opens this collection.

Parameters

openAll	Specifies whatever scenes flagged to not open with collection, should.
---------	--

Reopens all non-persistent scenes.

Implements SceneCollection.IMethods.

4.74.2.3 OpenAdditive()

```
SceneOperation OpenAdditive (
          bool openAll = false )
```

Opens this collection as additive.

Parameters

оре	enAll	Specifies whatever scenes flagged to not open with collection, should.
-----	-------	--

Additive collections are not "opened", all scenes within are merely opened like normal scenes. Mostly intended for convenience.

Implements SceneCollection.IMethods.

4.74.2.4 ToggleOpen()

```
SceneOperation ToggleOpen (
          bool openAll = false )
```

Toggles this collection open or closed.

Parameters

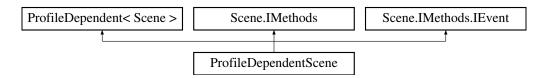
openState	Specifies whatever you have a preferred state to toggle to, this means collection will not be closed
	if true is passed. This can be used to ensure collection is open, without having an explicit check
	beforehand. The inverse is also the case for false.
openAll	Specifies whatever scenes flagged to not open with collection, should.

Implements SceneCollection.IMethods.

4.75 ProfileDependentScene

Represents a Scene that changes depending on active Profile.

Inheritance diagram for ProfileDependentScene:



Public Member Functions

• SceneOperation Open ()

Opens the scene.

• SceneOperation OpenAndActivate ()

Opens the scene and sets it as active.

• SceneOperation ToggleOpen ()

Toggles this scene open or closed.

• SceneOperation Close ()

Closes the scene.

SceneOperation Preload (Action onPreloaded=null)

Preloads the scene, to be displayed at a later time. See also: FinishPreload, DiscardPreload.

• SceneOperation FinishPreload ()

Finishes preloading the scene, displaying it.

• SceneOperation DiscardPreload ()

Discards the scene, if preloaded.

• SceneOperation OpenWithLoadingScreen (Scene loadingScreen)

Opens the scene while a loading screen is open.

· void SetActive ()

Sets the scene as active in heirarchy.

· void Open ()

Event method. Its meant for UnityEngine.Events.UnityEvent.

- void _OpenAndActivate ()
- void _ToggleOpen ()
- void _Close ()
- · void Preload ()
- void _FinishPreload ()
- void _DiscardPreload ()
- void _OpenWithLoadingScreen (Scene loadingScene)
- void SetActive ()

Public Member Functions inherited from ProfileDependent < Scene >

• bool GetModel (out T scene)

Gets the selected scene.

• T2 DoAction < T2 > (Func < T, T2 > action)

Performs an action on the scene.

void DoAction (Action< T > action)

Performs an action on the scene.

Additional Inherited Members

Public Attributes inherited from ProfileDependent < Scene >

• Dict list

The list of proxies for this T.

Properties inherited from ProfileDependent< Scene >

• bool isValid [get]

Gets if the current state of this T is valid.

4.75.1 Detailed Description

Represents a Scene that changes depending on active Profile.

4.75.2 Member Function Documentation

4.75.2.1 Close()

```
SceneOperation Close ( )
```

Closes the scene.

No effect if scene is already closed.

Implements Scene. IMethods.

4.75.2.2 FinishPreload()

```
SceneOperation FinishPreload ( )
```

Finishes preloading the scene, displaying it.

Scene must be preloaded beforehand.

Implements Scene.IMethods.

4.75.2.3 Open()

```
SceneOperation Open ( )
```

Opens the scene.

No effect if scene is already open.

Implements Scene.IMethods.

4.75.2.4 Preload()

Preloads the scene, to be displayed at a later time. See also: FinishPreload, DiscardPreload.

Scene must be closed beforehand.

Implements Scene.IMethods.

4.76 QueueUtility<T>

A utility that provides queuing.

Static Public Member Functions

• static bool **IsQueued** (T queueable)

Get if the item is queued.

• static bool **IsRunning** (T queueable)

Gets if the item is running.

• static void **Stop** (T queueable)

Cancels the queuable.

static void StopAll ()

Cancels all queued and running items.

Properties

• static bool isBusy [get]

Gets whatever any items in the queue are running.

• static | Enumerable < T > queue [get]

Gets the items currently in queue.

static IEnumerable < T > running [get]

Gets the items that are currently running.

Events

static Action queueEmpty

Occurs when an queued item finishes and queue is empty.

• static Action queueFilled

Occurs when an queued is added.

4.76.1 Detailed Description

A utility that provides queuing.

Type Constraints

T: IQueueable

4.77 ResolvedCrossReference

Represents a resolved reference.

Public Attributes

• CrossSceneReference reference

The unresolved reference.

• ObjectReference reference

The unresolved and resolved reference to the variable.

• ResolveStatus result

The result when setting value.

4.77.1 Detailed Description

Represents a resolved reference.

4.77.2 Member Data Documentation

4.77.2.1 reference [1/2]

ObjectReference reference

The unresolved reference.

The unresolved and resolved reference to the value.

4.77.2.2 reference [2/2]

ObjectReference reference

The unresolved and resolved reference to the variable.

The unresolved and resolved reference to the value.

4.78 ResolvedReference

Represents a resolved ObjectReference.

4.78.1 Detailed Description

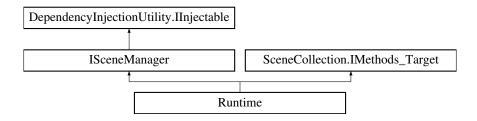
Represents a resolved ObjectReference.

4.79 Runtime 93

4.79 Runtime

Manages runtime functionality for Advanced Scene Manager such as open scenes and collection.

Inheritance diagram for Runtime:



Public Member Functions

IEnumerable < SceneLoader > GetToggleableSceneLoaders ()

Gets a list of all added scene loaders that can be toggled scene by scene.

• SceneLoader GetLoaderForScene (Scene scene)

Gets the loader for scene .

void AddSceneLoader< T > ()

Adds a scene loader.

void RemoveSceneLoader< T > ()

Removes a scene loader.

- SceneOperation Open (Scene scene)
- SceneOperation OpenAndActivate (Scene scene)
- SceneOperation Open (params Scene[] scenes)
- SceneOperation Open (IEnumerable < Scene > scenes)

Opens the scenes.

- SceneOperation OpenWithLoadingScreen (Scene scene, Scene loadingScreen)
- SceneOperation OpenWithLoadingScreen (IEnumerable < Scene > scene, Scene loadingScreen)

Opens a scene with a loading screen.

- SceneOperation Close (Scene scene)
- SceneOperation Close (params Scene[] scenes)
- SceneOperation Close (IEnumerable < Scene > scenes)

Closes the scenes.

- SceneOperation Preload (Scene scene, Action onPreloaded=null)
- SceneOperation FinishPreload (Scene scene)
- SceneOperation FinishPreload ()

Finishes the preload of the currently preloaded scene.

• SceneOperation DiscardPreload (Scene scene)

Discards preload of the scene, if preloaded.

SceneOperation DiscardPreload ()

Discards the preload of the currently preloaded scene.

• SceneOperation ToggleOpen (Scene scene)

Toggles the open state of this scene.

• void SetActive (Scene scene)

Sets the scene as active.

- SceneOperation Open (SceneCollection collection, bool openAll=false)
- SceneOperation OpenAdditive (SceneCollection collection, bool openAll=false)

Opens the collection without closing existing scenes.

SceneOperation OpenAdditive (IEnumerable < SceneCollection > collections, SceneCollection active ← Collection=null, Scene loadingScene=null)

Opens the collection without closing existing scenes.

- SceneOperation Close (SceneCollection collection)
- SceneOperation ToggleOpen (SceneCollection collection, bool openAll=false)
- SceneState GetState (Scene scene)

Gets the current state of the scene.

• void Track (Scene scene, UnityEngine.SceneManagement.Scene unityScene)

Tracks the specified scene as open.

- void Track (Scene scene)
- bool Untrack (Scene scene)

Untracks the specified scene as open.

· void UntrackScenes ()

Untracks all open scenes.

void Track (SceneCollection collection, bool isAdditive=false)

Tracks the collection as open.

• void Untrack (SceneCollection collection, bool isAdditive=false)

Untracks the collection.

void UntrackCollections ()

Untracks all collections.

• bool IsTracked (Scene scene)

Gets whatever this scene is tracked as open.

• bool IsTracked (SceneCollection collection)

Gets whatever this collection is tracked as open.

SceneOperation CloseAll (bool exceptLoadingScreens=true, bool exceptUnimported=true, params Scene[] except)

Closes all scenes and collections.

Public Attributes

Action onAllScenesClosed

Occurs when the last user scene closes.

Properties

• IEnumerable < Scene > openScenes [get]

Gets the scenes that are open.

• IEnumerable < SceneCollection > openAdditiveCollections [get]

Gets the collections that are opened as additive.

SceneCollection openCollection [get]

Gets the collection that is currently open.

• Scene preloadedScene [get]

Gets the scene that is currently preloaded.

• Scene activeScene [get]

Gets the active scene.

Scene dontDestroyOnLoad [get]

Gets the dontDestroyOnLoad scene.

bool isBusy [get]

Gets whatever ASM is busy with any scene operations.

IEnumerable < SceneOperation > runningOperations [get]

The currently running scene operations.

• IEnumerable < SceneOperation > queuedOperations [get]

Gets the current scene operation queue.

SceneOperation currentOperation [get]

Gets the current active operation in the queue.

4.79 Runtime 95

Properties inherited from ISceneManager

Events

Action < Scene > sceneOpened

Occurs when a scene is opened.

Action < Scene > sceneClosed

Occurs when a scene is closed.

Action < SceneCollection > collectionOpened

Occurs when a collection is opened.

Action < SceneCollection > collectionClosed

Occurs when a collection is closed.

• Action < Scene > scenePreloaded

Occurs when a scene is preloaded.

Action < Scene > scenePreloadFinished

Occurs when a previously preloaded scene is opened.

· Action startedWorking

Occurs when ASM has started working and is running scene operations.

· Action stoppedWorking

Occurs when ASM has finished working and no scene operations are running.

Events inherited from ISceneManager

```
    Action < SceneCollection > collectionClosed
```

- Action < SceneCollection > collectionOpened
- Action < Scene > sceneClosed
- Action < Scene > sceneOpened
- Action < Scene > scenePreloaded
- Action < Scene > scenePreloadFinished
- Action startedWorking
- Action stoppedWorking

4.79.1 Detailed Description

Manages runtime functionality for Advanced Scene Manager such as open scenes and collection.

4.79.2 Member Function Documentation

4.79.2.1 Close()

```
\begin{tabular}{ll} Scene Operation Close ( & & \\ IEnumerable < Scene > scenes ) \end{tabular}
```

Closes the scenes.

Closes persistent scenes.

4.79.2.2 Open()

```
SceneOperation Open ( {\tt IEnumerable} < {\tt Scene} > {\tt scenes} \ )
```

Opens the scenes.

Open scenes will not be re-opened, please close it first.

Implements ISceneManager.

4.79.2.3 OpenAdditive() [1/2]

Opens the collection without closing existing scenes.

No effect if no additive collections could be opened. Note that *activeCollection* will be removed from *collections* if it is contained within.

Implements ISceneManager.

4.79.2.4 OpenAdditive() [2/2]

Opens the collection without closing existing scenes.

Parameters

collection	The collection to open.
openAll	Specifies whatever all scenes should open, regardless of open flag.

Implements ISceneManager.

4.79.2.5 SetActive()

Sets the scene as active.

No effect if not open.

4.79 Runtime 97

4.79.2.6 Track() [1/2]

Tracks the specified scene as open.

Does not open scene.

Implements ISceneManager.

4.79.2.7 Track() [2/2]

Tracks the collection as open.

Does not open collection.

Implements ISceneManager.

4.79.2.8 Untrack() [1/2]

Untracks the specified scene as open.

Does not close scene.

Implements ISceneManager.

4.79.2.9 Untrack() [2/2]

Untracks the collection.

Does not close the collection.

4.79.2.10 UntrackCollections()

```
void UntrackCollections ( )
```

Untracks all collections.

Does not close collections.

Implements ISceneManager.

4.79.2.11 UntrackScenes()

```
void UntrackScenes ( )
```

Untracks all open scenes.

Does not close scenes.

Implements ISceneManager.

4.79.3 Member Data Documentation

4.79.3.1 onAllScenesClosed

Action onAllScenesClosed

Occurs when the last user scene closes.

This usually happens by mistake, and likely means that no user code would run, this is your chance to restore to a known state (return to main menu, for example), or crash to desktop.

Returning to main menu can be done like this:

SceneManager.app.Restart()

4.79.4 Property Documentation

4.79.4.1 activeScene

```
Scene activeScene [get]
```

Gets the active scene.

Returns null if the active scene is not imported.

4.80 Scene 99

4.79.4.2 dontDestroyOnLoad

```
Scene dontDestroyOnLoad [get]
```

 $Gets\ the\ dont Destroy On Load\ scene.$

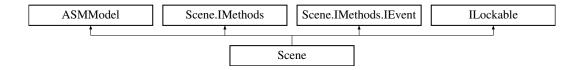
Returns null outside of play mode.

Implements ISceneManager.

4.80 Scene

Represents a scene.

Inheritance diagram for Scene:



Classes

- · interface IMethods
- interface IMethods_Target

Public Member Functions

IEnumerable < GameObject > GetRootGameObjects ()

Gets the root game objects in this Scene.

T FindObject< T > ()

Finds the object in the hierarchy of this Scene.

- bool FindObject < T > (out T component)
- IEnumerable < T > FindObjects < T > ()

Finds the objects in the hierarchy of this Scene.

• SceneOperation Open ()

Opens the scene.

• SceneOperation OpenAndActivate ()

Opens the scene and sets it as active.

• SceneOperation ToggleOpen ()

Toggles this scene open or closed.

• SceneOperation Close ()

Closes the scene.

SceneOperation Preload (Action onPreloaded=null)

Preloads the scene, to be displayed at a later time. See also: FinishPreload, DiscardPreload.

SceneOperation FinishPreload ()

Finishes preloading the scene, displaying it.

SceneOperation DiscardPreload ()

Discards the scene, if preloaded.

SceneOperation OpenWithLoadingScreen (Scene loadingScreen)

Opens the scene while a loading screen is open.

· void SetActive ()

Sets the scene as active in heirarchy.

- · void Activate ()
- void _Open ()

Event method. Its meant for UnityEngine.Events.UnityEvent.

- void OpenAndActivate ()
- void _ToggleOpen ()
- void Close ()
- void _Preload ()
- void FinishPreload ()
- void _DiscardPreload ()
- void OpenWithLoadingScreen (Scene loadingScene)
- · void SetActive ()
- override bool IsMatch (string q)

Gets if q matches ASMModel.name, id, path.

bool EvalOpenAsPersistent (SceneCollection parentCollection, SceneCollection collectionToOpen=null)

Gets whatever this scene will be opened as persistent.

void SetSceneLoader< T > ()

Specifies the scene loader to use for this scene.

• Type GetSceneLoader ()

Gets the scene loader specified for this scene. null if none set.

SceneLoader GetEffectiveSceneLoader ()

Gets the effective, contextual, scene loader for this scene. null if none found (this means normal ASM loader will be used).

void ClearSceneLoader ()

Clears custom scene loader for this scene. This means normal ASM functionality will be used.

Public Member Functions inherited from ASMModel

• virtual void Save ()

Saves the singleton to disk after a delay.

void SaveNow ()

Saves the singleton to disk.

void SaveNow (bool setDirty=true)

Saves the singleton to disk.

Static Public Member Functions

- static IEnumerable < Scene > Find (Func < Scene, bool > predicate)
- static Scene Find (string q)
- static bool TryFind (string q, out Scene scene)
- static IEnumerable < Scene > FindOpen (string q)
- static IEnumerable < Scene > FindOpen (Func < Scene, bool > predicate)

Static Public Member Functions inherited from ASMModel

• static string GenerateID ()

Generate id.

4.80 Scene 101

Static Public Attributes

static readonly string AssetSearchString = "t:" + typeof(Scene).FullName

Gets 't:AdvancedSceneManager.Models.Scene', the string to use in AssetDatabase.FindAssets(string).

Properties

• bool isImported [get]

Gets whatever we are tracked by AssetRef.

bool isIncluded [get]

Gets whatever this scene is included in build.

bool hasSceneAsset [get]

Gets if m sceneAsset has a value.

string path [get, set]

Gets the path of the associated SceneAsset.

• bool isLoadingScreen [get, set]

Gets if this scene is a loading screen.

• bool isSplashScreen [get, set]

Gets if this scene is a splash screen.

• bool isSpecial [get]

Gets if this is a 'special' scene.

• bool keepOpenWhenCollectionsClose [get, set]

Specifies whatever this scene will remain open when collections close.

• bool keepOpenWhenNewCollectionWouldReopen [get, set]

Specifies whatever this will remain open when a newly opened collection would have reopened it.

• bool isNonPersistant [get]

Gets whatever this scene will close normally after a collection closes.

bool openOnStartup [get, set]

Specifies whatever this scene should be opened on startup.

• bool openOnPlayMode [get, set]

Specifies whatever this scene should be opened when entering playmode.

• EditorPersistentOption autoOpenInEditor [get, set]

Specifies whatever this scene should be opened automatically outside of play-mode.

List < Scene > autoOpenInEditorScenes [get]

Specifies the scenes that should trigger this scene to open when autoOpenInEditor is set to EditorPersistentOption. ← WhenAnyOfTheFollowingScenesAreOpened.

• boolisLocked [get, set]

Gets if this scene is locked.

• string lockMessage [get, set]

Gets the lock message for this scene.

• IEnumerable < CrossSceneReference > crossSceneReferences [get, set]

Enumerates the cross-scene references defined on this scene.

bool isActive [get]

Gets if this scene is currently active.

• bool isOpenInHierarchy [get]

Gets whatever the scene is open in the hierarchy, this is true if scene is currently loading, if scene is preloaded, if scene is fully open.

- SceneState state [get]
- bool isOpen [get]

Gets whatever the scene is open.

• bool isPreloaded [get]

Gets whatever the scene is preloaded.

• unityScene? internalScene [get, set]

Gets the unityScene that this scene is associated with.

bool isPersistent [get]

Gets if this scene is opened as persistent.

• bool isDefaultASMScene [get]

Gets if this is a default ASM scene. These are located in 'Packages/Advanced Scene Manager/Default scenes/'.

bool isDontDestroyOnLoad [get]

Gets if this scene is the dontDestroyOnLoad scene.

bool isDynamic [get]

Gets if this scene is dynamic, it is not persisted to disk.

• string sceneLoader [get, set]

Specifies what SceneManagement.SceneLoader to use.

Properties inherited from ASMModel

```
• string id [get]
```

Gets the id of this ASMModel.

• new string name [get, protected set]

Additional Inherited Members

Static Protected Member Functions inherited from ASMModel

• static T CreateInternal < T > (string name)

Creates a profile. Throws if name is invalid.

4.80.1 Detailed Description

Represents a scene.

4.80.2 Member Function Documentation

4.80.2.1 Close()

```
SceneOperation Close ( )
```

Closes the scene.

No effect if scene is already closed.

Implements Scene.IMethods.

4.80.2.2 EvalOpenAsPersistent()

Gets whatever this scene will be opened as persistent.

4.80 Scene 103

Parameters

parentCollection	Specifies the parent collection that was opened before finalCollection .	
collectionToOpen	en Specifies the collection that will be opened, if you are not evaluating state after it would	
	have opened, pass null. If multiple collections are opened in sequence, then pass the	
	final one.	

4.80.2.3 FindObject< T >()

```
T FindObject< T > ( )
```

Finds the object in the hierarchy of this Scene.

Only works if scene is loaded.

Type Constraints

T: Component

4.80.2.4 FindObjects< T >()

```
IEnumerable < T > FindObjects < T > ()
```

Finds the objects in the hierarchy of this Scene.

Only works if scene is loaded.

Type Constraints

T: Component

4.80.2.5 FinishPreload()

```
SceneOperation FinishPreload ( )
```

Finishes preloading the scene, displaying it.

Scene must be preloaded beforehand.

Implements Scene.IMethods.

4.80.2.6 GetRootGameObjects()

```
IEnumerable< GameObject > GetRootGameObjects ( )
```

Gets the root game objects in this Scene.

Only usable if scene is open.

4.80.2.7 Open()

```
SceneOperation Open ( )
```

Opens the scene.

No effect if scene is already open.

Implements Scene. IMethods.

4.80.2.8 Preload()

Preloads the scene, to be displayed at a later time. See also: FinishPreload, DiscardPreload.

Scene must be closed beforehand.

Implements Scene.IMethods.

4.80.2.9 SetSceneLoader < T >()

```
void SetSceneLoader< T > ( )
```

Specifies the scene loader to use for this scene.

If the specified scene loader is not registered when scene is opened, then ASM will fallback to other scene loaders, if any (normal ASM functionality is used if not).

Type Constraints

T: SceneLoader

4.80.3 Property Documentation

4.80.3.1 hasSceneAsset

```
bool hasSceneAsset [get]
```

Gets if m_sceneAsset has a value.

Only available in the editor.

4.80.3.2 internalScene

```
unityScene? internalScene [get], [set]
```

Gets the unityScene that this scene is associated with.

null if scene is not open.

4.80 Scene 105

4.80.3.3 isImported

```
bool isImported [get]
```

Gets whatever we are tracked by AssetRef.

Only available in editor.

4.80.3.4 isLoadingScreen

```
bool isLoadingScreen [get], [set]
```

Gets if this scene is a loading screen.

Automatically updated.

If this is false for an actual loading screen, please make sure scene contains a Callbacks.LoadingScreen script.

Scene might sometimes have to be re-saved for this flag to appear.

4.80.3.5 isSpecial

```
bool isSpecial [get]
```

Gets if this is a 'special' scene.

A scene is special if any of the following is true: isSplashScreen, isLoadingScreen or isDontDestroyOnLoad.

4.80.3.6 isSplashScreen

```
bool isSplashScreen [get], [set]
```

Gets if this scene is a splash screen.

Automatically updated.

If this is false for an actual splash screen screen, please make sure scene contains a Callbacks. Splash Screen script.

Scene might sometimes have to be re-saved for this flag to appear.

4.80.3.7 keepOpenWhenCollectionsClose

```
bool keepOpenWhenCollectionsClose [get], [set]
```

Specifies whatever this scene will remain open when collections close.

You'll have to close it manually, if needed.

4.80.3.8 keepOpenWhenNewCollectionWouldReopen

```
bool keepOpenWhenNewCollectionWouldReopen [get], [set]
```

Specifies whatever this will remain open when a newly opened collection would have reopened it.

You'll have to close it manually, if needed.

4.80.3.9 openOnPlayMode

```
bool openOnPlayMode [get], [set]
```

Specifies whatever this scene should be opened when entering playmode.

Only effective when scene added to Profile.standaloneScenes.

4.80.3.10 openOnStartup

```
bool openOnStartup [get], [set]
```

Specifies whatever this scene should be opened on startup.

Only effective when scene added to Profile.standaloneScenes.

4.81 SceneCollection

Represents a collection of scenes.

Inheritance diagram for SceneCollection:



Classes

- interface IMethods
- interface IMethods_Target

4.81 SceneCollection 107

Public Member Functions

• override bool **IsMatch** (string q)

Gets if q matches ASMModel.name.

SceneOperation Open (bool openAll=false)

Opens this collection.

• SceneOperation OpenAdditive (bool openAll=false)

Opens this collection as additive.

SceneOperation ToggleOpen (bool openAll=false)

Toggles this collection open or closed.

SceneOperation Close ()

Closes this collection.

- void _Open (bool openAll=false)
- void _OpenAdditive (bool openAll=false)
- void _ToggleOpen ()
- void _ToggleOpen (bool openAll=false)
- · void _Close ()
- bool FindProfile (out Profile profile)

Find the Profile that this collection is associated with.

• Profile FindProfile ()

Find the Profile that this collection is associated with.

• T UserData < T > ()

Casts and returns userData as the specified type. Returns null if invalid type.

• bool Contains (Scene scene)

Gets if this collection contains scene .

• bool AutomaticallyOpenScene (Scene scene, bool? value=null)

Gets or sets whatever the scene should automatically open, when this collection is open. Default is true.

Public Member Functions inherited from ASMModel

· virtual void Save ()

Saves the singleton to disk after a delay.

• void SaveNow ()

Saves the singleton to disk.

void SaveNow (bool setDirty=true)

Saves the singleton to disk.

Static Public Member Functions

• static SceneCollection Find (string q, bool activeProfile=true)

Finds a collection based on its title or id.

• static bool **TryFind** (string q, out SceneCollection collection, bool activeProfile=true)

Finds a collection based on its title or id.

Static Public Member Functions inherited from ASMModel

• static string GenerateID ()

Generate id.

Static Public Attributes

• static readonly string **AssetSearchString** = "t:" + typeof(SceneCollection).FullName

Gets 't:AdvancedSceneManager.Models.SceneCollection', the string to use in AssetDatabase.FindAssets(string).

Properties

```
• int count [get]
```

Gets the scene count of this collection.

Scene this[int index] [get]

Gets the scene at the specified index.

• string title [get]

Gets the title of this collection.

• string description [get, set]

Gets the description of this collection.

IEnumerable < string > scenePaths [get]

Gets the scenes of this collection.

• IEnumerable < Scene > scenes [get]

Gets the scenes of this collection.

• IEnumerable < Scene > all Scenes [get]

Gets both scenes and loadingScreen.

• bool hasScenes [get]

Gets if this collection has any scenes.

• bool isStartupCollection [get]

Gets if this is a startup collection.

• ScriptableObject userData [get, set]

The extra data that is associated with this collection.

• bool isIncluded [get, set]

Gets whatever this collection should be included in build.

• Scene loadingScreen [get, set]

The loading screen that is associated with this collection.

• Scene effectiveLoadingScreen [get]

Gets effective loading screen depending on loadingScreenUsage.

• LoadingScreenUsage loadingScreenUsage [get, set]

Specifies what loading screen to use.

• Scene activeScene [get, set]

Specifies the scene that should be activated after collection is opened.

CollectionStartupOption startupOption [get, set]

Specifies startup option.

• CollectionLoadingThreadPriority loadingPriority [get, set]

Specifies the thread priority to use when opening this collection.

bool openAsPersistent [get, set]

Specifies whatever this collection should be opened as persistent.

bool openAsDisabled [get, set]

Specifies whatever this collection should be opened as disabled.

bool unloadUnusedAssets [get, set]

Calls Resources. Unload Unused Assets after collection is opened or closed.

List < Scene > scenesThatShouldNotAutomaticallyOpen [get]

Specifies scenes that should not open automatically.

• InputBinding binding [get]

4.81 SceneCollection 109

Specifies bindings for this scene.

• bool isLocked [get, set]

Gets if this collection is locked.

• string lockMessage [get, set]

Gets the lock message for this collection.

• bool **setActiveSceneWhenOpenedAsActive** [get, set]

Specifies whatever activeScene should be set, when collection is opened as additive.

• bool isOpen [get]

Gets if this collection is open.

bool isOpenAdditive [get]

Gets if this collection is opened additively.

bool isOpenNonAdditive [get]

Gets if this collection is opened additively.

Properties inherited from ASMModel

```
• string id [get]
```

Gets the id of this ASMModel.

• new string name [get, protected set]

Properties inherited from ISceneCollection

```
• string id [get]
```

Gets the id of this collection.

Additional Inherited Members

Static Protected Member Functions inherited from ASMModel

static T CreateInternal < T > (string name)

Creates a profile. Throws if name is invalid.

4.81.1 Detailed Description

Represents a collection of scenes.

Only one collection can be open at a time.

4.81.2 Member Function Documentation

4.81.2.1 Close()

```
SceneOperation Close ( )
```

Closes this collection.

No effect if collection is already closed. Note that "additive collections" are not actually opened, only its scenes are. Implements SceneCollection.IMethods.

4.81.2.2 Open()

```
SceneOperation Open (
          bool openAll = false )
```

Opens this collection.

Parameters

openAll Specifies whatever scenes flagged to not open with collection	tion, should.
---	---------------

Reopens all non-persistent scenes.

Implements SceneCollection.IMethods.

4.81.2.3 OpenAdditive()

```
SceneOperation OpenAdditive (
          bool openAll = false )
```

Opens this collection as additive.

Parameters

openAll	Specifies whatever scenes flagged to not open with collection, should.
---------	--

Additive collections are not "opened", all scenes within are merely opened like normal scenes. Mostly intended for convenience.

Implements SceneCollection.IMethods.

4.81.2.4 ToggleOpen()

```
SceneOperation ToggleOpen (
          bool openAll = false )
```

Toggles this collection open or closed.

Parameters

openState	Specifies whatever you have a preferred state to toggle to, this means collection will not be closed
	if true is passed. This can be used to ensure collection is open, without having an explicit check
	beforehand. The inverse is also the case for false.
openAll	Specifies whatever scenes flagged to not open with collection, should.

Implements SceneCollection.IMethods.

4.81.3 Property Documentation

4.81.3.1 allScenes

```
IEnumerable < Scene > all Scenes [get]
```

Gets both scenes and loadingScreen.

null is filtered out.

4.81.3.2 isStartupCollection

```
bool isStartupCollection [get]
```

Gets if this is a startup collection.

Only available in editor.

4.81.3.3 userData

```
ScriptableObject userData [get], [set]
```

The extra data that is associated with this collection.

Use UserData<T> to cast it to the desired type.

4.82 SceneCollectionTemplate

Represents a template for a SceneCollection.

Inheritance diagram for SceneCollectionTemplate:



Properties

• new string name [get]

Properties inherited from SceneCollection

int count [get]

Gets the scene count of this collection.

Scene this[int index] [get]

Gets the scene at the specified index.

• string title [get]

Gets the title of this collection.

• string description [get, set]

Gets the description of this collection.

• IEnumerable < string > scenePaths [get]

Gets the scenes of this collection.

• IEnumerable < Scene > scenes [get]

Gets the scenes of this collection.

• IEnumerable < Scene > all Scenes [get]

Gets both scenes and loadingScreen.

bool hasScenes [get]

Gets if this collection has any scenes.

• bool isStartupCollection [get]

Gets if this is a startup collection.

• ScriptableObject userData [get, set]

The extra data that is associated with this collection.

• bool isIncluded [get, set]

Gets whatever this collection should be included in build.

• Scene loadingScreen [get, set]

The loading screen that is associated with this collection.

Scene effectiveLoadingScreen [get]

Gets effective loading screen depending on loadingScreenUsage.

LoadingScreenUsage loadingScreenUsage [get, set]

Specifies what loading screen to use.

• Scene activeScene [get, set]

Specifies the scene that should be activated after collection is opened.

CollectionStartupOption startupOption [get, set]

Specifies startup option.

• CollectionLoadingThreadPriority loadingPriority [get, set]

Specifies the thread priority to use when opening this collection.

bool openAsPersistent [get, set]

Specifies whatever this collection should be opened as persistent.

• bool openAsDisabled [get, set]

Specifies whatever this collection should be opened as disabled.

bool unloadUnusedAssets [get, set]

Calls Resources. Unload Unused Assets after collection is opened or closed.

• List< Scene > scenesThatShouldNotAutomaticallyOpen [get]

Specifies scenes that should not open automatically.

InputBinding binding [get]

Specifies bindings for this scene.

• boolisLocked [get, set]

Gets if this collection is locked.

• string lockMessage [get, set]

Gets the lock message for this collection.

 $\bullet \ bool \ \textbf{setActiveSceneWhenOpenedAsActive} \quad [\texttt{get, set}]$

Specifies whatever activeScene should be set, when collection is opened as additive.

• bool isOpen [get]

Gets if this collection is open.

bool isOpenAdditive [get]

Gets if this collection is opened additively.

bool isOpenNonAdditive [get]

Gets if this collection is opened additively.

Properties inherited from ASMModel

• string id [get]

Gets the id of this ASMModel.

• new string name [get, protected set]

Properties inherited from ISceneCollection

string id [get]

Gets the id of this collection.

Additional Inherited Members

Public Member Functions inherited from SceneCollection

• override bool **IsMatch** (string q)

Gets if q matches ASMModel.name.

• SceneOperation Open (bool openAll=false)

Opens this collection.

• SceneOperation OpenAdditive (bool openAll=false)

Opens this collection as additive.

• SceneOperation ToggleOpen (bool openAll=false)

Toggles this collection open or closed.

SceneOperation Close ()

Closes this collection.

- void _Open (bool openAll=false)
- void _OpenAdditive (bool openAll=false)
- void _ToggleOpen ()
- void _ToggleOpen (bool openAll=false)
- void _Close ()
- bool **FindProfile** (out **Profile** profile)

Find the Profile that this collection is associated with.

• Profile FindProfile ()

Find the Profile that this collection is associated with.

• T UserData < T > ()

Casts and returns userData as the specified type. Returns null if invalid type.

bool Contains (Scene scene)

Gets if this collection contains scene .

• bool AutomaticallyOpenScene (Scene scene, bool? value=null)

Gets or sets whatever the scene should automatically open, when this collection is open. Default is true.

Public Member Functions inherited from ASMModel

• virtual void Save ()

Saves the singleton to disk after a delay.

· void SaveNow ()

Saves the singleton to disk.

void SaveNow (bool setDirty=true)

Saves the singleton to disk.

Static Public Member Functions inherited from SceneCollection

• static SceneCollection Find (string q, bool activeProfile=true)

Finds a collection based on its title or id.

• static bool **TryFind** (string q, out SceneCollection collection, bool activeProfile=true)

Finds a collection based on its title or id.

Static Public Member Functions inherited from ASMModel

static string GenerateID ()

Generate id.

Static Public Attributes inherited from SceneCollection

static readonly string AssetSearchString = "t:" + typeof(SceneCollection).FullName
 Gets 't:AdvancedSceneManager.Models.SceneCollection', the string to use in AssetDatabase.FindAssets(string).

Static Protected Member Functions inherited from ASMModel

static T CreateInternal < T > (string name)

Creates a profile. Throws if name is invalid.

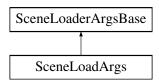
4.82.1 Detailed Description

Represents a template for a SceneCollection.

4.83 SceneLoadArgs

Specifies arguments for SceneLoader.LoadScene(Models.Scene, SceneLoadArgs).

Inheritance diagram for SceneLoadArgs:



Public Member Functions

• void SetCompleted (UnityEngine.SceneManagement.Scene scene)

Notifies ASM that the load is done.

- void SetCompleted (UnityEngine.SceneManagement.Scene scene, Func< IEnumerator > preloadCallback)
- void SetCompletedWithoutScene ()

Sets this loader as complete even though no scene was loaded.

bool CheckIsIncluded (bool logError=true)

Checks if the scene is actually included in build.

• UnityEngine.SceneManagement.Scene GetOpenedScene ()

Gets the UnityEngine.SceneManagement.Scene that was opened by this override.

Properties

• bool isPreload [get, set]

Specifies if the scene should be preloaded.

4.83 SceneLoadArgs 115

Properties inherited from SceneLoaderArgsBase

```
• bool isLoadingScreen [get]
```

Gets if this scene is a loading screen.

• bool isSplashScreen [get]

Gets if this scene is a splash screen.

4.83.1 Detailed Description

Specifies arguments for SceneLoader.LoadScene(Models.Scene, SceneLoadArgs).

4.83.2 Member Function Documentation

4.83.2.1 GetOpenedScene()

```
UnityEngine.SceneManagement.Scene GetOpenedScene ( )
```

Gets the UnityEngine.SceneManagement.Scene that was opened by this override.

Will return default if not found.

4.83.2.2 SetCompleted() [1/2]

```
void SetCompleted ( \label{total constraint} \mbox{UnityEngine::SceneManagement::Scene} \ scene \ )
```

Notifies ASM that the load is done.

Parameters

scene	The opened scene.
-------	-------------------

4.83.2.3 SetCompleted() [2/2]

Parameters

scene	The opened scene.]
preloadCallback	Specifies a callback that will be called when it is time to activate preloaded scene.	1

4.84 SceneLoader

Specifies a scene loader.

Inherited by RuntimeSceneLoader.

Public Member Functions

• virtual bool CanOpen (Scene scene)

Gets whatever this scene loader can open the scene.

• IEnumerator LoadScene (Scene scene, SceneLoadArgs e)

Loads the scene specified in e.scene.

• IEnumerator UnloadScene (Scene scene, SceneUnloadArgs e)

Unloads the scene specified in e.scene.

Static Public Member Functions

static string GetKey
 T > ()

Gets the key for the specified scene loader.

static string **GetKey** T > (T obj)

Gets the key for the specified scene loader.

Properties

• string Key [get]

Gets the key for this scene loader.

• virtual string sceneToggleText [get]

Specifies the text to display on the toggle in scene popup. Only has an effect if isGlobal is false.

• virtual Indicator indicator [get]

Specifies the indicator on scene fields for this scene loader.

• virtual bool isGlobal = true [get]

Specifies if this scene loader will can be applied to all scenes. Otherwise scenes will have to be explicitly flagged to open with this loader.

virtual bool activeOutsideOfPlayMode [get]

Specifies whatever this loader will run outside of play mode or not.

• virtual bool activeInPlayMode = true [get]

Specifies whatever this loader will run in play mode or not.

bool canBeActivated [get]

Gets whatever this loader may be activated in the current context.

4.84.1 Detailed Description

Specifies a scene loader.

4.84.2 Property Documentation

4.84.2.1 isGlobal

```
virtual bool isGlobal = true [get]
```

Specifies if this scene loader will can be applied to all scenes. Otherwise scenes will have to be explicitly flagged to open with this loader.

To flag a scene to be opened with this loader, the following two methods can be used:

If sceneToggleText is non-empty, a toggle will be displayed in scene popup.

Programmatically Scene.SetSceneLoader<T> can be used.

4.84.2.2 Key

```
string Key [get]
```

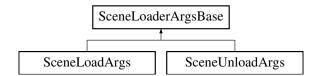
Gets the key for this scene loader.

This is equal to System. Type. Full Name.

4.85 SceneLoaderArgsBase

Base class for SceneLoadArgs and SceneUnloadArgs.

Inheritance diagram for SceneLoaderArgsBase:



Properties

- bool isLoadingScreen [get]
 - Gets if this scene is a loading screen.
- bool isSplashScreen [get]

Gets if this scene is a splash screen.

4.85.1 Detailed Description

Base class for SceneLoadArgs and SceneUnloadArgs.

4.86 SceneManager

The central Advanced Scene Manager API. Provides access to the most important things in ASM.

Static Public Member Functions

static void OnInitialized (Action action)
 Call action when ASM has initialized.

Properties

```
static AssetsProxy assets = new() [get]
static IEnumerable < Scene > openScenes [get]
static SceneCollection openCollection [get]
static Scene preloadedScene [get]
static Runtime runtime = new() [get]
static App app = new App() [get]
static SettingsProxy settings = new() [get]
static Profile profile [get]
static bool isInitialized [get]
```

Gets whatever ASM is initialized. Calling ASM methods may fail if false, this is due to ASMSettings singleton not being loaded yet.

4.86.1 Detailed Description

The central Advanced Scene Manager API. Provides access to the most important things in ASM.

4.86.2 Member Function Documentation

4.86.2.1 OnInitialized()

```
static void OnInitialized ( \mbox{Action action }) \mbox{ [static]}
```

Call action when ASM has initialized.

Will call immediately if already initialized.

4.86.3 Property Documentation

4.86.3.1 isInitialized

```
bool isInitialized [static], [get]
```

Gets whatever ASM is initialized. Calling ASM methods may fail if false, this is due to ASMSettings singleton not being loaded yet.

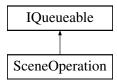
See also OnInitialized(Action).

4.87 SceneOperation 119

4.87 SceneOperation

A scene operation is a queueable operation that can open or close scenes. See also: SceneAction.

Inheritance diagram for SceneOperation:



Public Member Functions

SceneOperation WithFriendlyText (string text)

Specifies description for coroutine.

· void Cancel ()

Cancel this operation.

• SceneOperation OpenAndActivate (Scene scene)

Opens the scene, and makes sure it is activated afterwards.

SceneOperation Focus (Scene scene)

Sets focus to the specified scene. Overrides selected scene in collections.

SceneOperation Activate (Scene scene=null)

Sets focus to the specified scene. Overrides selected scene in collections. If null, then the first scene opened will be set as active.

• SceneOperation With (SceneCollection collection, bool setActiveScene=false)

Specifies an associated collection.

- SceneOperation Open (SceneCollection collection, bool openAll=false)
- SceneOperation Close (SceneCollection collection)
- SceneOperation Open (params Scene[] scenes)

Specifies the scenes to open.

• SceneOperation Close (params Scene[] scenes)

Specifies the scenes to close.

SceneOperation Callback (params Callback[] callbacks)

Specifies user callbacks.

• SceneOperation Preload (Scene scene)

Specifies a scene to preload.

- SceneOperation Open (IEnumerable < Scene > scenes)
- SceneOperation Close (IEnumerable < Scene > scenes)
- SceneOperation Callback (IEnumerable < Callback > callbacks)
- SceneOperation With (Scene loadingScene, bool useLoadingScene=true)

Specifies loading screen to use.

SceneOperation With (Action < LoadingScreen > loadingScreenCallback)

Specifies a callback when loading screen is opened, before LoadingScreen.OnOpen is called.

• SceneOperation DisableLoadingScreen (bool useLoadingScene=false)

Specifies whatever loading screen should be used.

• SceneOperation **EnableLoadingScreen** (bool useLoadingScene=true)

Specifies whatever loading screen should be used.

SceneOperation With (ThreadPriority loadingPriority)

Specifies the ThreadPriority to use.

SceneOperation UnloadUsedAssets ()

Specifies whatever Resources. UnloadUnusedAssets should be called at the end (before loading screen).

SceneOperation With (Progress< float > customProgress)

Specifies custom progress that will be counted as part of progress.

· void CloseAll (params Scene[] except)

Closes all scenes prior to opening any scenes.

void CloseAllNonPersistent (params Scene[] except)

Closes all non-persistent scenes prior to opening any scenes.

SceneOperation ReportProgress (Action < float > progress)

Specifies a callback for when progress changes.

Static Public Member Functions

• static SceneOperation Queue ()

Queues a new scene operation.

• static SceneOperation Start ()

Starts a new scene operation, ignoring queue.

static void AddCallback (Callback callback)

Adds the callback to every scene operation.

static void RemoveCallback (Callback callback)

Removes a callback that was added to every scene operation.

Properties

```
• static SceneOperation done = new SceneOperation() { hasRun = true } [get]
```

Gets a SceneOperation that has already completed.

string description [get, protected set]

Specifies description for coroutine.

• SceneCollection collection [get]

Specifies the collection that is being opened or closed.

bool setActiveCollectionScene [get]

Specifies whatever active scene should be set when possible.

• Scene focus [get]

Sets focus to the specified scene. Overrides selected scene in collections.

• bool focusSingleScene [get]

Sets the first opened scene as active.

• IEnumerable < Scene > open [get]

Gets the scenes specified to open.

• IEnumerable < Scene > close [get]

Gets the scenes specified to close.

• IEnumerable < Callback > callbacks [get]

Gets the user defined callbacks.

Scene preload [get]

Gets the scene specified to preload.

• Scene loadingScene [get]

Gets the specified loading screen.

Action < LoadingScreen > loadingScreenCallback [get]

Gets the specified loading screen callback.

bool useLoadingScene [get]

Gets whatever a loading screen should be used.

• ThreadPriority? **loadingPriority** [get]

4.87 SceneOperation 121

Gets the specified ThreadPriority to be used.

bool? unloadUnusedAssets [get]

Gets whatever Resources.UnloadUnusedAssets should be called at the end (before loading screen).

• IEnumerable < Scene > closedScenes [get]

Gets the scenes that was closed during this operation.

IEnumerable < Scene > openedScenes [get]

Gets the scenes that was opened during this operation.

• bool isLoadingScreen [get, set]

Specifies whatever this scene operation was started by ASM to open a loading screen.

override bool keepWaiting [get]

Inherited from CustomYieldInstruction. Tells unity whatever the operation is done or not.

Phase phase [get]

The phase the this scene operation is currently in.

bool wasCancelled [get]

Gets if this scene operation is cancelled.

• Progress< float > customProgress [get]

Gets custom progress, if there is any. Will be counted as part of progress.

LoadingScreen openedLoadingScreen [get]

Gets the loading screen that was opened by this operation.

float progress [get]

Gets the current progress.

Events

· static OnBeforeStart beforeStart

Occurs before operation has started working.

4.87.1 Detailed Description

A scene operation is a queueable operation that can open or close scenes. See also: SceneAction.

4.87.2 Member Function Documentation

4.87.2.1 Activate()

Sets focus to the specified scene. Overrides selected scene in collections. If null, then the first scene opened will be set as active.

No effect if preloading.

4.87.2.2 Cancel()

```
void Cancel ( )
```

Cancel this operation.

Note that the operation might not be cancelled immediately, if user defined callbacks are currently running.

4.87.2.3 Close()

```
SceneOperation Close (
          params Scene[] scenes )
```

Specifies the scenes to close.

Can be called multiple times to add more scenes.

4.87.2.4 Focus()

```
SceneOperation Focus (
Scene scene )
```

Sets focus to the specified scene. Overrides selected scene in collections.

No effect if preloading.

4.87.2.5 Open()

```
SceneOperation Open (
          params Scene[] scenes )
```

Specifies the scenes to open.

Can be called multiple times to add more scenes.

4.87.2.6 Preload()

Specifies a scene to preload.

A scene specified to preload cannot also be added to open or close lists.

4.87.2.7 ReportProgress()

```
\begin{tabular}{ll} Scene Operation Report Progress ( \\ Action < float > progress ) \end{tabular}
```

Specifies a callback for when progress changes.

Only one callback can be registered, previous one will be replaced by *progress*.

4.87 SceneOperation 123

4.87.2.8 With()

Specifies loading screen to use.

Has no effect if useLoadingScene is false.

4.87.3 Property Documentation

4.87.3.1 close

```
IEnumerable<Scene> close [get]
```

Gets the scenes specified to close.

List will change depending on when its called (i.e. only open scenes can be closed).

4.87.3.2 focus

```
Scene focus [get]
```

Sets focus to the specified scene. Overrides selected scene in collections.

No effect if preloading.

4.87.3.3 keepWaiting

```
override bool keepWaiting [get]
```

Inherited from CustomYieldInstruction. Tells unity whatever the operation is done or not.

Inherited from CustomYieldInstruction. Tells unity whatever the operation is done or not.

4.87.3.4 open

```
IEnumerable<Scene> open [get]
```

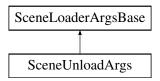
Gets the scenes specified to open.

List will change depending on when its called (i.e. only closed scenes can be opened).

4.88 SceneUnloadArgs

 $Specifies\ arguments\ for\ Scene Loader. Unload Scene (Models. Scene,\ Scene Unload Args).$

Inheritance diagram for SceneUnloadArgs:



Public Member Functions

void SetCompleted ()

Notifies ASM that the unload is done.

Additional Inherited Members

Properties inherited from SceneLoaderArgsBase

• bool isLoadingScreen [get]

Gets if this scene is a loading screen.

• bool isSplashScreen [get]

Gets if this scene is a splash screen.

4.88.1 Detailed Description

 $Specifies\ arguments\ for\ Scene Loader. Unload Scene (Models. Scene,\ Scene Unload Args).$

4.89 SceneUtility

An utility class to perform actions on scenes.

4.89 SceneUtility 125

Static Public Member Functions

static IEnumerable < scene > GetAllOpenUnityScenes ()

Get all open unity scenes.

static bool IsIncluded (Scene scene)

Gets if the scene is included in build.

- static void Move (this GameObject obj, Scene scene)
- static void Move (this GameObject obj, scene scene)
- · static GameObject MoveHere (this MonoBehaviour mono, GameObject obj)

Moves obj to this scene.

static GameObject CreateHere (this MonoBehaviour mono)

Creates a game object in this scene.

static GameObject CreateHere (this MonoBehaviour mono, string name)

Creates a game object in this scene.

static GameObject CreateHere (this MonoBehaviour mono, string name, params Type[] components)

Creates a game object in this scene.

• static TComponent CreateHere < TComponent > (this MonoBehaviour mono, string gameObjectName)

Creates a game object in this scene. Adds and returns component TComponent .

static Scene CreateDynamic (string name, UnityEngine.SceneManagement.LocalPhysicsMode local
 — PhysicsMode=UnityEngine.SceneManagement.LocalPhysicsMode.None)

Creates a scene at runtime, that is not saved to disk.

- static bool FindCollection (this Scene scene, out SceneCollection collection)
- static SceneCollection FindCollection (this Scene scene)

Attempts to find best match for collection.

• static IEnumerable < Scene Collection > FindCollections (this Scene scene, bool allProfiles=false)

Finds which collections that this scene is a part of.

• static IEnumerable < SceneCollection > FindCollections (this Scene scene, Profile profile)

Finds which collections that this scene is a part of.

static IEnumerable < Scene > FindOpen (string q)

Find open scenes by name or path.

static Scene Find (string q)

Find scenes by name or path.

static IEnumerable < Scene > FindOpen (Func < Scene, bool > predicate)

Find open scenes by predicate.

static IEnumerable < Scene > Find (Func < Scene, bool > predicate)

Find scenes by predicate.

- static bool ASMScene (this Component component, out Scene scene)
- static Scene ASMScene (this GameObject gameObject, out Scene scene)
- static Scene ASMScene (this Component component)
- static Scene ASMScene (this GameObject gameObject)
- static bool ASMScene (this scene thisScene, out Scene scene)
- static Scene ASMScene (this scene scene)

Gets the associated ASM Scene.

static IEnumerable < Scene > EvaluateFinalSceneList (Profile profile, App.StartupProps props)

Evaluate the final scene list after startup.

• static IEnumerable < Scene > EvaluateFinalSceneList (IEnumerable < SceneCollection > collections)

Evaluate the final scene list after opening a sequence of collections.

static void SetEnabled (this Scene scene, bool isEnabled)

Sets all root objects as enabled / disabled.

• static void Enable (this Scene scene)

Sets all root objects as enabled.

static void Disable (this Scene scene)

Sets all root objects as disabled.

Properties

```
• static bool isStartupScene [get]
```

Gets if current, and only, scene is the startup scene.

• static scene dontDestroyOnLoadScene [get]

Gets the dontDestroyOnLoad scene. Returns null if not open.

static bool hasAnyScenes [get]

Gets if there are any scenes open that are not dynamically created, and not yet saved to disk.

• static int unitySceneCount [get]

4.89.1 Detailed Description

An utility class to perform actions on scenes.

4.89.2 Member Function Documentation

4.89.2.1 CreateDynamic()

Creates a scene at runtime, that is not saved to disk.

Returns null if scene could not be created.

4.89.2.2 Disable()

Sets all root objects as disabled.

Only has an effect if scene is open.

4.89.2.3 Enable()

```
static void Enable ( {\tt this~Scene~\it scene}~)~[{\tt static}]
```

Sets all root objects as enabled.

Only has an effect if scene is open.

4.89.2.4 EvaluateFinalSceneList() [1/2]

Evaluate the final scene list after opening a sequence of collections.

Parameters

collections	The sequence of collections that would be opened.

4.89.2.5 EvaluateFinalSceneList() [2/2]

Evaluate the final scene list after startup.

Parameters

profile	The profile that would be used to run startup process with.
props	The startup props that would be used to run process with.

4.89.2.6 FindCollection()

Attempts to find best match for collection.

Only checks current profile.

4.89.2.7 SetEnabled()

Sets all root objects as enabled / disabled.

Only has an effect if scene is open.

4.90 ScriptableObjectUtility

Contains utility methods for ScriptableObject.

Static Public Member Functions

static void Save (this ScriptableObject obj)
 Saves the ScriptableObject.

4.90.1 Detailed Description

Contains utility methods for ScriptableObject.

4.90.2 Member Function Documentation

4.90.2.1 Save()

Saves the ScriptableObject.

Safe to call from outside editor, but has no effect.

4.91 SerializableDictionary < TKey, TValue >

A serializable dictionary.

Inherits Dictionary < TKey, TValue >, and ISerializationCallbackReceiver.

4.91.1 Detailed Description

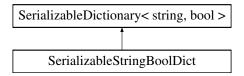
A serializable dictionary.

Older unity versions might need a wrapper class, since they won't support serializing generic types. Don't forget SerializableAttribute on wrapper!

4.92 SerializableStringBoolDict

A serializable dictionary of string and bool.

Inheritance diagram for SerializableStringBoolDict:



4.92.1 Detailed Description

A serializable dictionary of string and bool.

4.93 SettingsProxy 129

4.93 SettingsProxy

Provides access to ASM settings.

Properties

• ASMSettings project [get]

The project-wide ASM settings.

• Profile profile [get]

The current ASM profile.

4.93.1 Detailed Description

Provides access to ASM settings.

4.93.2 Property Documentation

4.93.2.1 profile

```
Profile profile [get]
```

The current ASM profile.

Could be null.

4.94 SpamCheck

Provides an easy way to check for spamming.

Public Member Functions

• bool IsSpam ()

Gets if an action was executed not long enough ago.

void MarkAsExecuted ()

Marks this spam check as executed, disallowing any actions until cooldown has run out.

· void Execute (Action action)

Runs action if allowed.

Properties

• static SpamCheck Global = new SpamCheck() [get]

Gets the global spam check.

• boolisEnabled [get, set]

Gets or sets if this SpamCheck is enabled.

• float executeCooldown [get, set]

Gets or sets the cooldown.

• float lastExecute [get]

Gets the time an action was executed last.

• float timeSinceLastExecute [get]

Gets the time an action was executed last.

4.94.1 Detailed Description

Provides an easy way to check for spamming.

4.94.2 Property Documentation

4.94.2.1 Global

```
SpamCheck Global = new SpamCheck() [static], [get]
```

Gets the global spam check.

Don't worry about conflicts with ASM stuff, we use a separate one.

4.94.2.2 isEnabled

```
bool isEnabled [get], [set]
```

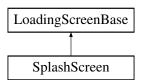
Gets or sets if this SpamCheck is enabled.

When disabled actions will run without checking whatever it is a spam call.

4.95 SplashScreen

A class that contains callbacks for splash screens.

Inheritance diagram for SplashScreen:



Public Member Functions

- override IEnumerator OnOpen ()
 Called when splash scene is opened.
- override IEnumerator OnClose ()

Called when splash scene is about to close.

4.95 SplashScreen 131

Public Member Functions inherited from LoadingScreenBase

• IEnumerator OnOpen ()

Called when the loading screen is opened.

• IEnumerator OnClose ()

Called when the loading screen is about to close.

virtual void OnProgressChanged (float progress)

Called when progress has changed.

• bool HasPressedAnyKey ()

Gets if any key has been pressed this frame.

WaitUntil WaitForAnyKey ()

Returns WaitUntil that waits for user to press any key.

Additional Inherited Members

Public Attributes inherited from LoadingScreenBase

Action < LoadingScreenBase > onDestroy

Occurs when loading screen is destroyed.

· Canvas canvas

Properties inherited from LoadingScreenBase

• bool isOpening [get]

Gets whatever we're currently opening.

• boolisOpen [get]

Gets whatever we're currently open.

• bool isClosing [get]

Gets whatever we're currently closing.

4.95.1 Detailed Description

A class that contains callbacks for splash screens.

SplashScreen and LoadingScreen cannot coexist within the same scene.

4.95.2 Member Function Documentation

4.95.2.1 OnClose()

```
override IEnumerator OnClose ( ) [abstract]
```

Called when splash scene is about to close.

Use this callback to hide your splash screen, the scene manager will wait until its done.

4.95.2.2 OnOpen()

```
override IEnumerator OnOpen ( ) [abstract]
```

Called when splash scene is opened.

Use this callback to show your splash screen, the scene manager will wait until its done.

4.96 StandaloneCollection

Represents a collection of standalone scenes. These scenes are guaranteed to be included in build (if the associated Profile is active).

Inherits ISceneCollection.IEditable.

Properties

```
• string id [get]
```

Gets the id of this collection.

• IEnumerable < Scene > scenes [get]

Gets the scenes of this collection.

• IEnumerable < string > scenePaths [get]

Gets the scenes of this collection.

• string title [get]

Gets the title of this collection.

• string description [get]

Gets the description of this collection.

• IEnumerable < Scene > startupScenes [get]

Gets all scenes that will be opened on startup.

• int count [get]

Gets the scene count of this collection.

Scene this[int index] [get]

Gets the scene at the specified index.

4.96.1 Detailed Description

Represents a collection of standalone scenes. These scenes are guaranteed to be included in build (if the associated Profile is active).

Usage: Profile.standaloneScenes.

4.97 App.StartupProps

An object that persists start properties across domain reload, which is needed when configurable enter play mode is set to reload domain on enter play mode.

4.98 UIFadeExtensions 133

Public Member Functions

• StartupProps (StartupProps props)

Creates a new props, from the specified props, copying its values.

Public Attributes

· bool forceOpenAllScenesOnCollection

Specifies whatever all scenes on openCollection should be opened.

· Color? fadeColor

The color for the fade out.

SceneCollection openCollection

Specifies a collection to be opened after startup process is done.

Properties

• bool runStartupProcessWhenPlayingCollection [get, set]

Specifies whatever startup process should run before openCollection is opened.

• bool runStartupProcess [get]

Gets if startup process should run.

• Color effectiveFadeColor [get]

Gets the effective fade animation color, uses fadeColor if specified. Otherwise PlayerSettings.SplashScreen.

backgroundColor will be used during first startup. On subsequent restarts Color.black will be used (ASM restart, not application restart!).

4.97.1 Detailed Description

An object that persists start properties across domain reload, which is needed when configurable enter play mode is set to reload domain on enter play mode.

4.97.2 Member Data Documentation

4.97.2.1 fadeColor

Color? fadeColor

The color for the fade out.

Unity splash screen color will be used if null.

4.98 UIFadeExtensions

Provides extension methods for CanvasGroup.

Static Public Member Functions

• static IEnumerator **Fade** (this CanvasGroup group, float to, float duration, bool setBlocksRaycasts=true)

Animates the alpha of a CanvasGroup.

- static IEnumerator **Fade** (this Graphic text, float to, float duration, bool ignoreTimeScale)

 Animates the alpha of a Graphic.
- static IEnumerator **Fade** (this RectTransform element, float to, float duration, bool ignoreTimeScale)

 Animates the alpha of all Graphic found on element and children.
- static IEnumerator **Fade** (this UIBehaviour element, float to, float duration, bool ignoreTimeScale)

 Animates the alpha of all Graphic found on element and children.

4.98.1 Detailed Description

Provides extension methods for CanvasGroup.

4.99 UnityCompatibiltyHelper

Contains helpers for dealing with multiple versions of unity.

Static Public Member Functions

• static T FindFirstObjectByType< T > ()

4.99.1 Detailed Description

Contains helpers for dealing with multiple versions of unity.

Index

ActionUtility, 19	UseDefault, 14
Trylnvoke, 19	WhenAnyOfTheFollowingScenesAreOpened, 14
Activate	AdvancedSceneManager.Models.Helpers, 15
SceneOperation, 121	AdvancedSceneManager.Models.Internal, 15
activeScene	AdvancedSceneManager.Models.Utility, 15
Runtime, 98	AdvancedSceneManager.Utility, 16
AdvancedSceneManager, 9	AdvancedSceneManager.Utility.CrossSceneReferences,
AdvancedSceneManager.Callbacks, 9	17
AdvancedSceneManager.Core, 10	Cleared, 17
CloseCallbacks, 11	Default, 17
CustomActions, 11	Restored, 17
LoadScenes, 11	SceneStatus, 17
OpenCallbacks, 11	AdvancedSceneManager::Utility::CrossSceneReferences::SceneReference
Phase, 10	17
UnloadScenes, 11	After
AdvancedSceneManager.DependencyInjection, 11	Callback, 35
AdvancedSceneManager.DependencyInjection.Editor,	allowExcludingCollectionsFromBuild
11	ASMSettings, 30
AdvancedSceneManager.Models, 11	allScenes
Hold, 12	SceneCollection, 110
InputBindingInteractionType, 12	AnySceneOpened
Open, 12	AdvancedSceneManager.Models.Enums, 14
Toggle, 12	App, 20
AdvancedSceneManager.Models.Enums, 12	CancelQuit, 21
AnySceneOpened, 14	Exit, 21
Auto, 13	Quit, 21
BelowNormal, 13	App.StartupProps, 132
CollectionLoadingThreadPriority, 13	fadeColor, 133
CollectionStartupOption, 13	ASMFilePathAttribute, 22
DoNotOpen, 13	ASMModel, 22
DoNotUse, 14	Save, 23
EditorPersistentOption, 13	SaveNow, 23
High, 13	ASMModelExtensions, 23
LoadingScreenUsage, 14	CloseAll, 24
Low, 13	IndexOf< T >, 24
Manual, 14	OpenAdditive, 25
Never, 14	OpenAll, 25
Normal, 13	ASMSceneHelper, 25
NotOpen, 14	Close, 27
Open, 13, 15	FinishPreload, 27
Opening, 15	Open, 27
Override, 14	Preload, 27
Preloaded, 15	ASMScriptableSingleton < T >, 28
Preloading, 15	Save, 28
Queued, 14	SaveNow, 28
SceneCreated, 14	ASMSettings, 29
SceneImportOption, 14	allowExcludingCollectionsFromBuild, 30
SceneState, 14	enableCrossSceneReferences, 30
Unknown, 14	assetPath
	400011 4111

Assets, 31	CoroutineFrameEndEvent
Assets, 31	CoroutineUtility.Events, 46
assetPath, 31	CoroutineFrameStartEvent
AssetSearchUtility, 32	CoroutineUtility.Events, 46
AssetsProxy, 32	CoroutineUtility, 37
Async< T >, 33	StartCoroutineGlobal, 38
Auto	StopAllCoroutines, 38
AdvancedSceneManager.Models.Enums, 13	Timer, 38
	CoroutineUtility.Events, 45
Before	CoroutineFrameEndEvent, 46
Callback, 35	CoroutineFrameStartEvent, 46
BelowNormal	onSubroutineStart, 47
AdvancedSceneManager.Models.Enums, 13	CreateDynamic
BuildOption, 33	SceneUtility, 126
	CrossSceneReference, 39
Callback, 34	
After, 35	CrossSceneReferenceUtility, 39
Before, 35	CanSceneBeSaved, 40
scene, 36	CustomActions
When, 35	AdvancedSceneManager.Core, 11
CallbackUtility, 36	Defends
CallbackUtility, 50 CallbackUtility.FluentInvokeAPI< T >, 47	Default
· · · · · · · · · · · · · · · · · · ·	AdvancedSceneManager.Utility.CrossSceneReferences
WithCallback, 48	17
Cancel	DefaultScenes, 40
SceneOperation, 121	Enumerate, 41
CancelQuit	fadeScene, 41
App, 21	iconBounceScene, 41
CanSceneBeSaved	inGameToolbarScene, 42
CrossSceneReferenceUtility, 40	pauseScene, 42
canvas	pressAnyKeyScene, 42
LoadingScreenBase, 78	progressBarScene, 42
CanvasSortOrderUtility, 36	quoteScene, 42
MakeSure, 37	splashASMScene, 42
Cleared	splashFadeScene, 43
AdvancedSceneManager.Utility.CrossSceneReferen	©®ependencylniectionUtility, 43
17	DependencyInjectionUtility.IInjectable, 57
Close	DictionaryUtility, 44
ASMSceneHelper, 27	Disable
ProfileDependentCollection, 87	SceneUtility, 126
ProfileDependentScene, 90	DoAction
Runtime, 95	LoadingScreenUtility, 79
Scene, 102	- · · · · · · · · · · · · · · · · · · ·
Scene.IMethods, 59	ProfileDependent < T >, 86
Scene.lMethods_Target, 62	DoAction < T2 >
SceneCollection, 109	ProfileDependent< T >, 86
SceneCollection.IMethods, 60	DoNotOpen
	AdvancedSceneManager.Models.Enums, 13
SceneOperation, 121	DoNotUse
close	AdvancedSceneManager.Models.Enums, 14
SceneOperation, 123	dontDestroyOnLoad
CloseAll	Runtime, 98
ASMModelExtensions, 24	DynamicCollection, 44
CloseCallbacks	
AdvancedSceneManager.Core, 11	EditorPersistentOption
CloseLoadingScreen	AdvancedSceneManager.Models.Enums, 13
LoadingScreenUtility, 79	Enable
CollectionLoadingThreadPriority	SceneUtility, 126
AdvancedSceneManager.Models.Enums, 13	enableCrossSceneReferences
CollectionStartupOption	ASMSettings, 30
AdvancedSceneManager.Models.Enums, 13	Enumerate

DefaultScenes, 41	ICollectionClose, 51
EvalOpenAsPersistent	ICollectionCloseAsync, 52
Scene, 102	ICollectionExtraDataCallbacks, 52
EvaluateFinalSceneList	ICollectionExtraDataCallbacksAsync, 53
SceneUtility, 126, 127	ICollectionOpen, 53
Exit	ICollectionOpenAsync, 54
App, 21	iconBounceScene
πρρ, 21	DefaultScenes, 41
fadeColor	IFadeLoadingScreen, 57
App.StartupProps, 133	ILockable, 57
fadeScene	IndexOf< T >
DefaultScenes, 41	ASMModelExtensions, 24
FallbackSceneUtility, 47	
FindCollection	inGameToolbarScene
SceneUtility, 127	DefaultScenes, 42
FindObject < T >	InitializeInEditorAttribute, 64
Scene, 103	InitializeInEditorMethodAttribute, 64
	InputBinding, 64
FindObjects < T >	InputBindingInteractionType
Scene, 103	AdvancedSceneManager.Models, 12
FinishPreload	InputButton, 65
ASMSceneHelper, 27	path, 65
ProfileDependentScene, 90	internalScene
Scene, 103	Scene, 104
Scene.IMethods, 59	Invoke
Scene.IMethods_Target, 62	MainThreadUtility, 80
Focus	Invoke < T >
SceneOperation, 122	MainThreadUtility, 81
focus	IProfileManager, 65
SceneOperation, 123	IProjectSettings, 66
	IQueueable, 66
GenerateID	OnTurn, 67
GuidReferenceUtility, 50	IRuntime, 67
GetModel	ISceneCallbacks, 69
ProfileDependent< T >, 86	ISceneClose, 70
GetOpenedScene	ISceneCloseAsync, 70
SceneLoadArgs, 115	ISceneCollection, 71
GetOrAddPersistent	ISceneManager, 71
GuidReferenceUtility, 50	ISceneOpen, 73
GetRootGameObjects	ISceneOpenAsync, 73
Scene, 103	isEnabled
Global	SpamCheck, 130
SpamCheck, 130	isGlobal
GlobalCoroutine, 48	SceneLoader, 117
GuidReference, 49	isImported
GuidReferenceUtility, 50	Scene, 104
GenerateID, 50	isInitialized
GetOrAddPersistent, 50	SceneManager, 118
	isLoadingScreen
HasPressedAnyKey	Scene, 105
LoadingScreenBase, 77	isOpen
hasSceneAsset	LoadingScreenBase, 78
Scene, 104	_
High	isSpecial Scene, 105
AdvancedSceneManager.Models.Enums, 13	
Hold	isSplashScreen
AdvancedSceneManager.Models, 12	Scene, 105
	isStartupCollection
IApp, 51	SceneCollection, 110
IAssetsProvider, 51	keepOpenWhenCollectionsClose

Scene, 105	OnTurn
keepOpenWhenNewCollectionWouldReopen	IQueueable, 67
Scene, 105	Open
keepWaiting	AdvancedSceneManager.Models, 12
SceneOperation, 123	AdvancedSceneManager.Models.Enums, 13, 15
Key	ASMSceneHelper, 27
SceneLoader, 117	ProfileDependentCollection, 87
	ProfileDependentScene, 90
Lazy, 18	Runtime, 95
Lazy.Utility, 18	Scene, 103
Lerp	Scene.IMethods, 59
LerpUtility, 74	Scene.IMethods Target, 62
LerpUtility, 74	SceneCollection, 109
Lerp, 74	SceneCollection.IMethods, 60
LoadingScreen, 75	SceneOperation, 122
OnClose, 76	open
OnOpen, 76	SceneOperation, 123
LoadingScreenBase, 76	OpenAdditive
canvas, 78	ASMModelExtensions, 25
HasPressedAnyKey, 77	ProfileDependentCollection, 88
isOpen, 78	Runtime, 96
LoadingScreenUsage	SceneCollection, 110
AdvancedSceneManager.Models.Enums, 14	SceneCollection.IMethods, 60
LoadingScreenUtility, 78	OpenAll
CloseLoadingScreen, 79	ASMModelExtensions, 25
DoAction, 79	OpenCallbacks
LoadScenes	AdvancedSceneManager.Core, 11
AdvancedSceneManager.Core, 11	Opening
Low	AdvancedSceneManager.Models.Enums, 15
AdvancedSceneManager.Models.Enums, 13	openOnPlayMode
	Scene, 106
MainThreadUtility, 80	openOnStartup
Invoke, 80	Scene, 106
Invoke $<$ T $>$, 81	Override
MakeSure	AdvancedSceneManager.Models.Enums, 14
CanvasSortOrderUtility, 37	
Manual	ParallelASMCallbacks, 82
AdvancedSceneManager.Models.Enums, 14	path
N	InputButton, 65
Never	pauseScene
AdvancedSceneManager.Models.Enums, 14	DefaultScenes, 42
Normal	Phase
AdvancedSceneManager.Models.Enums, 13	AdvancedSceneManager.Core, 10
NotOpen	Preload
AdvancedSceneManager.Models.Enums, 14	ASMSceneHelper, 27
ObjectReference, 81	ProfileDependentScene, 90
onAllScenesClosed	Scene, 104
Runtime, 98	Scene.lMethods, 59
OnClose	Scene.IMethods_Target, 63
LoadingScreen, 76	SceneOperation, 122
SplashScreen, 131	Preloaded
OnInitialized	AdvancedSceneManager.Models.Enums, 15
SceneManager, 118	Preloading
_	AdvancedSceneManager.Models.Enums, 15
OnOpen	pressAnyKeyScene
LoadingScreen, 76	DefaultScenes, 42
SplashScreen, 131 onSubroutineStart	Profile, 82
	removedCollections, 84
CoroutineUtility.Events, 47	scenes, 84

specialScenes, 84	SaveNow
startupCollections, 85	ASMModel, 23
profile	ASMScriptableSingleton< T >, 28
SettingsProxy, 129	Scene, 99
ProfileDependent< T >, 85	Close, 102
DoAction, 86	EvalOpenAsPersistent, 102
DoAction < T2 >, 86	FindObject< T >, 103
GetModel, 86	FindObjects < T >, 103
ProfileDependent< T >.Dict, 43	FinishPreload, 103
ProfileDependentCollection, 86	GetRootGameObjects, 103
Close, 87	hasSceneAsset, 104
Open, 87	internalScene, 104
OpenAdditive, 88	isImported, 104
ToggleOpen, 88	isLoadingScreen, 105
ProfileDependentScene, 89	isSpecial, 105
Close, 90	isSplashScreen, 105
FinishPreload, 90	keepOpenWhenCollectionsClose, 105
Open, 90	keepOpenWhenNewCollectionWouldReopen, 105
Preload, 90	Open, 103
progressBarScene	openOnPlayMode, 106
DefaultScenes, 42	
Delaultoceries, 42	openOnStartup, 106
Queued	Preload, 104
AdvancedSceneManager.Models.Enums, 14	SetSceneLoader< T >, 104
	scene
QueueUtility< T >, 91	Callback, 36
Quit	Scene.IMethods, 58
App, 21	Close, 59
quoteScene	FinishPreload, 59
DefaultScenes, 42	Open, 59
	Preload, 59
reference	Scene.IMethods.IEvent, 54
ResolvedCrossReference, 92	Scene.IMethods_Target, 61
removedCollections	Close, 62
Profile, 84	FinishPreload, 62
ReportProgress	Open, 62
SceneOperation, 122	Preload, 63
ResolvedCrossReference, 92	Scene.IMethods_Target.IEvent, 55
reference, 92	SceneCollection, 106
ResolvedReference, 92	allScenes, 110
Restored	Close, 109
AdvancedSceneManager.Utility.CrossSceneReference	ces, isStartupCollection, 110
17	Open, 109
Runtime, 93	OpenAdditive, 110
activeScene, 98	ToggleOpen, 110
Close, 95	userData, 111
dontDestroyOnLoad, 98	SceneCollection.IMethods, 59
onAllScenesClosed, 98	•
Open, 95	Close, 60
OpenAdditive, 96	Open, 60
SetActive, 96	OpenAdditive, 60
Track, 96, 97	ToggleOpen, 61
Untrack, 97	SceneCollection.IMethods.IEvent, 56
UntrackCollections, 97	SceneCollection.IMethods_Target, 63
UntrackScenes, 98	SceneCollection.IMethods_Target.IEvent, 56
ontrackoceries, 30	SceneCollectionTemplate, 111
Save	SceneCreated
ASMModel, 23	AdvancedSceneManager.Models.Enums, 14
ASMScriptableSingleton< T >, 28	SceneImportOption
ScriptableObjectUtility, 128	AdvancedSceneManager.Models.Enums, 14

SceneLoadArgs, 114	DefaultScenes, 42
GetOpenedScene, 115	splashFadeScene
SetCompleted, 115	DefaultScenes, 43
SceneLoader, 116	SplashScreen, 130
isGlobal, 117	OnClose, 131
Key, 117	OnOpen, 131
SceneLoaderArgsBase, 117	StandaloneCollection, 132
SceneManager, 118	StartCoroutineGlobal
isInitialized, 118	CoroutineUtility, 38
OnInitialized, 118	startupCollections
SceneOperation, 119	Profile, 85
Activate, 121	StopAllCoroutines
Cancel, 121	CoroutineUtility, 38
Close, 121	• ,
close, 123	Timer
Focus, 122	CoroutineUtility, 38
focus, 123	Toggle
keepWaiting, 123	AdvancedSceneManager.Models, 12
Open, 122	ToggleOpen
open, 123	ProfileDependentCollection, 88
Preload, 122	SceneCollection, 110
ReportProgress, 122	SceneCollection.IMethods, 61
With, 122	Track
scenes	Runtime, 96, 97
Profile, 84	TryInvoke
SceneState	ActionUtility, 19
	ristionic time, re
AdvancedSceneManager.Models.Enums, 14 SceneStatus	UIFadeExtensions, 133
AdvancedSceneManager.Utility.CrossSceneReferen	
AdvancedScene wanager. Office Scene neteren	Unknown
17 Seemal InlandArgo, 104	AdvancedSceneManager.Models.Enums, 14
SceneUnloadArgs, 124	UnloadScenes
SceneUtility, 124	AdvancedSceneManager.Core, 11
CreateDynamic, 126	Untrack
Disable, 126	Runtime, 97
Enable, 126	UntrackCollections
EvaluateFinalSceneList, 126, 127	Runtime, 97
FindCollection, 127	UntrackScenes
SetEnabled, 127	Runtime, 98
ScriptableObjectUtility, 127	UseDefault
Save, 128	AdvancedSceneManager.Models.Enums, 14
SerializableDictionary< TKey, TValue >, 128	userData
SerializableStringBoolDict, 128	SceneCollection, 111
SetActive	Scene Collection, 111
Runtime, 96	When
SetCompleted	Callback, 35
SceneLoadArgs, 115	WhenAnyOfTheFollowingScenesAreOpened
SetEnabled	AdvancedSceneManager.Models.Enums, 14
SceneUtility, 127	With
SetSceneLoader< T >	SceneOperation, 122
Scene, 104	WithCallback
SettingsProxy, 129	CallbackUtility.FluentInvokeAPI< T >, 48
profile, 129	CambackCunty.1 IdentifivoreAFIC 1 >, 40
SpamCheck, 129	
Global, 130	
isEnabled, 130	
specialScenes	
Profile, 84	
splashASMScene	