

Namespace CatSweeper.Core

Classes

[BootstrapEndState](#)

[GameEntry](#)

[GameEntry.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[GameEntry.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

[GameEntry.SignalName](#)

Cached StringNames for the signals contained in this class, for fast lookup.

[GameState](#)

[GameStateMachine](#)

[GameplayInitState](#)

[InfoLoadState](#)

[LoadStageState](#)

[LoadStageState.InitParam](#)

[LocalLoadState](#)

[LoginState](#)

[LoginState.Param](#)

[UILoadState](#)

[UserDataLoadState](#)

Enums

[GameStatelD](#)

Class BootstrapEndState

Inheritance

object → State<GameStatId, GameState, GameStateMachine> → GameState → BootstrapEndState

Implements

[IDisposable](#)

Inherited Members

[State<GameStatId, GameState, GameStateMachine>.Id](#)
[State<GameStatId, GameState, GameStateMachine>.StateMachine](#)
[State<GameStatId, GameState, GameStateMachine>.IsReady\(\)](#)
[State<GameStatId, GameState, GameStateMachine>.StartContext\(StateParam\)](#)
[State<GameStatId, GameState, GameStateMachine>.OnEndContext\(\)](#)
[State<GameStatId, GameState, GameStateMachine>.Dispose\(\)](#)
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class BootstrapEndState : GameState, IDisposable
```

Properties

Id

Declaration

```
public override GameStateId Id { get; }
```

Property Value

TYPE

GameStatId

Overrides

State<GameStatId, GameState, GameStateMachine>.Id

Methods

StartContext(StateParam)

Declaration

```
public override void StartContext(StateParam stateParam)
```

Parameters

TYPE	NAME
StateParam	stateParam

Overrides

State<GameStatId, GameState, GameStateMachine>.StartContext(StateParam)

Implements

IDisposable

© Bus Fighter. All rights reserved.

Class GameEntry

Inheritance

[object](#) → GodotObject → Node → GameEntry

Implements

[IDisposable](#)

Inherited Members

Node.NotificationEnterTree
Node.NotificationExitTree
Node.NotificationMovedInParent
Node.NotificationReady
Node.NotificationPaused
Node.NotificationUnpaused
Node.NotificationPhysicsProcess
Node.NotificationProcess
Node.NotificationParented
Node.NotificationUnparented
Node.NotificationSceneInstantiated
Node.NotificationDragBegin
Node.NotificationDragEnd
Node.NotificationPathRenamed
Node.NotificationChildOrderChanged
Node.NotificationInternalProcess
Node.NotificationInternalPhysicsProcess
Node.NotificationPostEnterTree
Node.NotificationDisabled
Node.NotificationEnabled
Node.NotificationResetPhysicsInterpolation
Node.NotificationEditorPreSave
Node.NotificationEditorPostSave
Node.NotificationWMMouseEnter
Node.NotificationWMMouseExit
Node.NotificationWMWindowFocusIn
Node.NotificationWMWindowFocusOut
Node.NotificationWMCloseRequest
Node.NotificationWMGoBackRequest
Node.NotificationWMSizeChanged
Node.NotificationWMDpiChange
Node.NotificationVpMouseEnter

Node.NotificationVpMouseExit
Node.NotificationWMPositionChanged
Node.NotificationOsMemoryWarning
Node.NotificationTranslationChanged
Node.NotificationWMAbout
Node.NotificationCrash
Node.NotificationOslmeUpdate
Node.NotificationApplicationResumed
Node.NotificationApplicationPaused
Node.NotificationApplicationFocusIn
Node.NotificationApplicationFocusOut
Node.NotificationTextServerChanged
Node.GetNode<T>(NodePath)
Node.GetNodeOrNull<T>(NodePath)
Node.GetChild<T>(int, bool)
Node.GetChildOrNull<T>(int, bool)
Node.GetOwner<T>()
Node.GetOwnerOrNull<T>()
Node.GetParent<T>()
Node.GetParentOrNull<T>()
Node._EnterTree()
Node._ExitTree()
Node._GetConfigurationWarnings()
Node._Input(InputEvent)
Node._PhysicsProcess(double)
Node._Process(double)
Node._Ready()
Node._ShortcutInput(InputEvent)
Node._UnhandledInput(InputEvent)
Node._UnhandledKeyInput(InputEvent)
Node.PrintOrphanNodes()
Node.AddSibling(Node, bool)
Node.AddChild(Node, bool, Node.InternalMode)
Node.RemoveChild(Node)
Node.Reparent(Node, bool)
Node.GetChildCount(bool)
Node.GetChildren(bool)
Node.GetChild(int, bool)
Node.HasNode(NodePath)
Node.GetNode(NodePath)
Node.GetNodeOrNull(NodePath)
Node.GetParent()
Node.FindChild(string, bool, bool)
Node.FindChildren(string, string, bool, bool)
Node.FindParent(string)
Node.HasNodeAndResource(NodePath)
Node.GetNodeAndResource(NodePath)
Node.IsInsideTree()
Node.IsPartOfEditedScene()
Node.IsAncestorOf(Node)

Node.IsGreater Than(Node)
Node.GetPath()
Node.GetPath To(Node, bool)
Node.AddToGroup(StringName, bool)
Node.RemoveFromGroup(StringName)
Node.IsInGroup(StringName)
Node.MoveChild(Node, int)
Node.GetGroups()
Node.GetIndex(bool)
Node.PrintTree()
Node.PrintTreePretty()
Node.GetTreeString()
Node.GetTreeStringPretty()
Node.PropagateNotification(int)
Node.PropagateCall(StringName, Array, bool)
Node.SetPhysicsProcess(bool)
Node.GetPhysicsProcessDelta Time()
Node.IsPhysicsProcessing()
Node.GetProcessDelta Time()
Node.SetProcess(bool)
Node.IsProcessing()
Node.SetProcessInput(bool)
Node.IsProcessingInput()
Node.SetProcessShortcutInput(bool)
Node.IsProcessingShortcutInput()
Node.SetProcessUnhandledInput(bool)
Node.IsProcessingUnhandledInput()
Node.SetProcessUnhandledKeyInput(bool)
Node.IsProcessingUnhandledKeyInput()
Node.CanProcess()
Node.SetDisplayFolded(bool)
Node.IsDisplayedFolded()
Node.SetProcessInternal(bool)
Node.IsProcessingInternal()
Node.SetPhysicsProcessInternal(bool)
Node.IsPhysicsProcessingInternal()
Node.IsPhysicsInterpolated()
Node.IsPhysicsInterpolatedAndEnabled()
Node.ResetPhysicsInterpolation()
Node.SetTranslationDomainInherited()
Node.GetWindow()
Node.GetLastExclusiveWindow()
Node.GetTree()
Node.CreateTween()
Node.Duplicate(int)
Node.ReplaceBy(Node, bool)
Node.setSceneInstanceLoadPlaceholder(bool)
Node.GetSceneInstanceLoadPlaceholder()
Node.setEditableInstance(Node, bool)
Node.setEditableInstance(Node)

```
Node.GetViewport()
Node.QueueFree()
Node.RequestReady()
Node.IsNodeReady()
Node.SetMultiplayerAuthority(int, bool)
Node.GetMultiplayerAuthority()
Node.IsMultiplayerAuthority()
Node.RpcConfig(StringName, Variant)
Node.GetRpcConfig()
Node.Atr(string, StringName)
Node.AtrN(string, StringName, int, StringName)
Node.Rpc(StringName, params Variant[])
Node.Rpc(StringName, ReadOnlySpan<Variant>)
Node.RpcId(long, StringName, params Variant[])
Node.RpcId(long, StringName, ReadOnlySpan<Variant>)
Node.UpdateConfigurationWarnings()
Node.CallDeferredThreadGroup(StringName, params Variant[])
Node.CallDeferredThreadGroup(StringName, ReadOnlySpan<Variant>)
Node.SetDeferredThreadGroup(StringName, Variant)
Node.NotifyDeferredThreadGroup(int)
Node.CallThreadSafe(StringName, params Variant[])
Node.CallThreadSafe(StringName, ReadOnlySpan<Variant>)
Node.SetThreadSafe(StringName, Variant)
Node.NotifyThreadSafe(int)
Node.EmitSignalReady()
Node.EmitSignalRenamed()
Node.EmitSignalTreeEntered()
Node.EmitSignalTreeExiting()
Node.EmitSignalTreeExited()
Node.EmitSignalChildEnteredTree(Node)
Node.EmitSignalChildExitingTree(Node)
Node.EmitSignalChildOrderChanged()
Node.EmitSignalReplacingBy(Node)
Node.EmitSignalEditorDescriptionChanged(Node)
Node.EmitSignalEditorStateChanged()
Node.InvokeGodotClassMethod(in godot_string_name, NativeVariantPtrArgs, out godot_variant)
Node.HasGodotClassMethod(in godot_string_name)
Node.HasGodotClassSignal(in godot_string_name)
Node.Name
Node.UniqueNameInOwner
Node.SceneFilePath
Node.Owner
Node.Multiplayer
Node.ProcessMode
Node.ProcessPriority
Node.ProcessPhysicsPriority
Node.ProcessThreadGroup
Node.ProcessThreadGroupOrder
Node.ProcessThreadMessages
Node.PhysicsInterpolationMode
```

Node.AutoTranslateMode
Node.EditorDescription
Node.Ready
Node.Renamed
Node.TreeEntered
Node.TreeExiting
Node.TreeExited
Node.ChildEnteredTree
Node.ChildExitingTree
Node.ChildOrderChanged
NodeReplacingBy
Node.EditorDescriptionChanged
Node.EditorStateChanged
GodotObject.NotificationPostinitialize
GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)

GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)
GodotObject.Cally(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObjectGetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Namespace: **CatSweeper.Core**

Assembly: CatSweeper.dll

```
[ScriptPath("res://Script/Core/GameEntry/GameEntry.cs")]
public class GameEntry : Node, IDisposable
```

Methods

RegisterGenericService(Game)

Declaration

```
public static Game RegisterGenericService(Game game)
```

Parameters

TYPE	NAME
Game	game

Returns

TYPE
Game

Implements

IDisposable

Extension Methods

NodeUtil.DontDestroyOnLoad(Node)

Class GameEntry.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.MethodName](#) → [Node.MethodName](#) → [GameEntry.MethodName](#)

Inherited Members

[Node.MethodName._EnterTree](#)

[Node.MethodName._ExitTree](#)

[Node.MethodName._GetConfigurationWarnings](#)

[Node.MethodName._Input](#)

[Node.MethodName._PhysicsProcess](#)

[Node.MethodName._Process](#)

[Node.MethodName._Ready](#)

[Node.MethodName._ShortcutInput](#)

[Node.MethodName._UnhandledInput](#)

[Node.MethodName._UnhandledKeyInput](#)

[Node.MethodName.PrintOrphanNodes](#)

[Node.MethodName.AddSibling](#)

[Node.MethodName.SetName](#)

[Node.MethodName.GetName](#)

[Node.MethodName.AddChild](#)

[Node.MethodName.RemoveChild](#)

[Node.MethodName.Reparent](#)

[Node.MethodName.GetChildCount](#)

[Node.MethodName.GetChildren](#)

[Node.MethodName.GetChild](#)

[Node.MethodName.HasNode](#)

[Node.MethodName.GetNode](#)

[Node.MethodName.GetNodeOrNull](#)

[Node.MethodName.GetParent](#)

[Node.MethodName.FindChild](#)

[Node.MethodName.FindChildren](#)

[Node.MethodName.FindParent](#)

[Node.MethodName.HasNodeAndResource](#)

[Node.MethodName.GetNodeAndResource](#)

[Node.MethodName.IsInsideTree](#)

[Node.MethodName.IsPartOfEditedScene](#)

[Node.MethodName.IsAncestorOf](#)

[Node.MethodName.IsGreaterThan](#)

[Node.MethodName.GetPath](#)

Node.MethodName.GetPathTo
Node.MethodName.AddToGroup
Node.MethodName.RemoveFromGroup
Node.MethodName.IsInGroup
Node.MethodName.MoveChild
Node.MethodName.GetGroups
Node.MethodName.SetOwner
Node.MethodName.GetOwner
Node.MethodName.GetIndex
Node.MethodName.PrintTree
Node.MethodName.PrintTreePretty
Node.MethodName.GetTreeString
Node.MethodName.GetTreeStringPretty
Node.MethodName.SetSceneFilePath
Node.MethodName.GetSceneFilePath
Node.MethodName.PropagateNotification
Node.MethodName.PropagateCall
Node.MethodName.SetPhysicsProcess
Node.MethodName.GetPhysicsProcessDeltaTime
Node.MethodName.IsPhysicsProcessing
Node.MethodName.GetProcessDeltaTime
Node.MethodName.SetProcess
Node.MethodName.SetProcessPriority
Node.MethodName.GetProcessPriority
Node.MethodName.SetPhysicsProcessPriority
Node.MethodName.GetPhysicsProcessPriority
Node.MethodName.IsProcessing
Node.MethodName.SetProcessInput
Node.MethodName.IsProcessingInput
Node.MethodName.SetProcessShortcutInput
Node.MethodName.IsProcessingShortcutInput
Node.MethodName.SetProcessUnhandledInput
Node.MethodName.IsProcessingUnhandledInput
Node.MethodName.SetProcessUnhandledKeyInput
Node.MethodName.IsProcessingUnhandledKeyInput
Node.MethodName.SetProcessMode
Node.MethodName.GetProcessMode
Node.MethodName.CanProcess
Node.MethodName.SetProcessThreadGroup
Node.MethodName.GetProcessThreadGroup
Node.MethodName.SetProcessThreadMessages
Node.MethodName.GetProcessThreadMessages
Node.MethodName.SetProcessThreadGroupOrder
Node.MethodName.GetProcessThreadGroupOrder
Node.MethodName.SetDisplayFolded
Node.MethodName.IsDisplayedFolded
Node.MethodName.SetProcessInternal
Node.MethodName.IsProcessingInternal
Node.MethodName.SetPhysicsProcessInternal
Node.MethodName.IsPhysicsProcessingInternal

Node.MethodName.SetPhysicsInterpolationMode
Node.MethodName.GetPhysicsInterpolationMode
Node.MethodName.IsPhysicsInterpolated
Node.MethodName.IsPhysicsInterpolatedAndEnabled
Node.MethodName.ResetPhysicsInterpolation
Node.MethodName.SetAutoTranslateMode
Node.MethodName.GetAutoTranslateMode
Node.MethodName.SetTranslationDomainInherited
Node.MethodName.GetWindow
Node.MethodName.GetLastExclusiveWindow
Node.MethodName.GetTree
Node.MethodName.CreateTween
Node.MethodName.Duplicate
Node.MethodName.ReplaceBy
Node.MethodName.SetSceneInstanceLoadPlaceholder
Node.MethodName.GetSceneInstanceLoadPlaceholder
Node.MethodName.SetEditableInstance
Node.MethodName.IsEditableInstance
Node.MethodName.GetViewport
Node.MethodName.QueueFree
Node.MethodName.RequestReady
Node.MethodName.IsNodeReady
Node.MethodName.SetMultiplayerAuthority
Node.MethodName.GetMultiplayerAuthority
Node.MethodName.IsMultiplayerAuthority
Node.MethodName.GetMultiplayer
Node.MethodName.RpcConfig
Node.MethodName.GetRpcConfig
Node.MethodName.SetEditorDescription
Node.MethodName.GetEditorDescription
Node.MethodName._SetImportPath
Node.MethodName._GetImportPath
Node.MethodName.SetUniqueNameInOwner
Node.MethodName.IsUniqueNameInOwner
Node.MethodName.Atr
Node.MethodName.AtrN
Node.MethodName.Rpc
Node.MethodName.RpcId
Node.MethodName.UpdateConfigurationWarnings
Node.MethodName.CallDeferredThreadGroup
Node.MethodName.SetDeferredThreadGroup
Node.MethodName.NotifyDeferredThreadGroup
Node.MethodName.CallThreadSafe
Node.MethodName.SetThreadSafe
Node.MethodName.NotifyThreadSafe
GodotObject.MethodName._Get
GodotObject.MethodName._GetPropertyList
GodotObject.MethodName._IterGet
GodotObject.MethodName._IterInit
GodotObject.MethodName._IterNext

GodotObject.MethodName._Notification
GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceId
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyListChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr
GodotObject.MethodName.TrN

GodotObject.MethodName.GetTranslationDomain
GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: **CatSweeper.Core**

Assembly: CatSweeper.dll

Syntax

```
public class GameEntry.MethodName : Node.MethodName
```

Fields

InfoBuildByte

Cached name for the 'InfoBuildByte' method.

Declaration

```
public static readonly StringName InfoBuildByte
```

Field Value

TYPE

StringName

RegisterJsonConverters

Cached name for the 'RegisterJsonConverters' method.

Declaration

```
public static readonly StringName RegisterJsonConverters
```

Field Value

TYPE

StringName

© Bus Fighter. All rights reserved.

Class GameEntry.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

[object](#) → GodotObject.PropertyName → Node.PropertyName → GameEntry.PropertyName

Inherited Members

Node.PropertyName._ImportPath
Node.PropertyName.Name
Node.PropertyName.UniqueNameInOwner
Node.PropertyName.SceneFilePath
Node.PropertyName.Owner
Node.PropertyName.Multiplayer
Node.PropertyName.ProcessMode
Node.PropertyName.ProcessPriority
Node.PropertyName.ProcessPhysicsPriority
Node.PropertyName.ProcessThreadGroup
Node.PropertyName.ProcessThreadGroupOrder
Node.PropertyName.ProcessThreadMessages
Node.PropertyName.PhysicsInterpolationMode
Node.PropertyName.AutoTranslateMode
Node.PropertyName.EditorDescription
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class GameEntry.PropertyName : Node.PropertyName
```

Fields

_logLevel

Cached name for the '_logLevel' field.

Declaration

```
public static readonly StringName _logLevel
```

Field Value

TYPE

StringName

© Bus Fighter. All rights reserved.

Class GameEntry.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

[object](#) → GodotObject.SignalName → Node.SignalName → GameEntry.SignalName

Inherited Members

Node.SignalName.Ready
Node.SignalName.Renamed
Node.SignalName.TreeEntered
Node.SignalName.TreeExiting
Node.SignalName.TreeExited
Node.SignalName.ChildEnteredTree
Node.SignalName.ChildExitingTree
Node.SignalName.ChildOrderChanged
Node.SignalName.ReplacingBy
Node.SignalName.EditorDescriptionChanged
Node.SignalName.EditorStateChanged
GodotObject.SignalName.ScriptChanged
GodotObject.SignalName.PropertyListChanged
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class GameEntry.SignalName : Node.SignalName
```


Class GameState

Inheritance

object → State<GameStatId, GameState, GameStateMachine> → GameState → BootstrapEndState → GameplayInitState → InfoLoadState → LoadStageState → LocalLoadState → LoginState → UILoadState → UserDataLoadState

Implements

[IDisposable](#)

Inherited Members

[State<GameStatId, GameState, GameStateMachine>.Id](#)
[State<GameStatId, GameState, GameStateMachine>.StateMachine](#)
[State<GameStatId, GameState, GameStateMachine>.IsReady\(\)](#)
[State<GameStatId, GameState, GameStateMachine>.StartContext\(StateParam\)](#)
[State<GameStatId, GameState, GameStateMachine>.OnEndContext\(\)](#)
[State<GameStatId, GameState, GameStateMachine>.Dispose\(\)](#)
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class GameState : State<GameStateId, GameState, GameStateMachine>, IDisposable
```

Implements

[IDisposable](#)

Enum GameStatId

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public enum GameStateId
```

Fields

NAME

BootstrapEnd

GameplayInit

InfoLoad

LoadStage

LocalLoad

Login

UILoad

UserDataTableLoad

Extension Methods

[EnumExtension.hasFlag\(Enum, Enum\)](#)

Class GameStateMachine

Inheritance

object → StateMachine<GameStatId, GameState, GameStateMachine> → GameStateMachine

Implements

IStateMachine<GameStatId>

IService

IDisposable

Inherited Members

StateMachine<GameStatId, GameState, GameStateMachine>.lastStatId

StateMachine<GameStatId, GameState, GameStateMachine>.currentStatId

StateMachine<GameStatId, GameState, GameStateMachine>.SubscribeBeforeStateChange(Action<StateChange Record<GameStatId>>)

StateMachine<GameStatId, GameState, GameStateMachine>.SubscribeAfterStateChange(Action<StateChange Record<GameStatId>>)

StateMachine<GameStatId, GameState, GameStateMachine>.RegisterState(GameState)

StateMachine<GameStatId, GameState, GameStateMachine>.CanGoToState(GameStatId, StateParam)

StateMachine<GameStatId, GameState, GameStateMachine>.TryGoToState(GameStatId, StateParam)

StateMachine<GameStatId, GameState, GameStateMachine>.ForceGoToState(GameStatId, StateParam)

StateMachine<GameStatId, GameState, GameStateMachine>.GetStateUnsafe(GameStatId)

StateMachine<GameStatId, GameState, GameStateMachine>.GetStateUnsafe<T>(GameStatId)

StateMachine<GameStatId, GameState, GameStateMachine>.TryGetState(GameStatId, out GameState)

StateMachine<GameStatId, GameState, GameStateMachine>.Dispose()

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class GameStateMachine : StateMachine<GameStateId, GameState, GameStateMachine>, IState
```

Constructors

GameStateMachine()

Declaration

```
public GameStateMachine()
```

Implements

IStateMachine<TStateId>

IService

IDisposable

© Bus Fighter. All rights reserved.

Class GameplayInitState

Inheritance

object → State<GameStatId, GameState, GameStateMachine> → GameState → GameplayInitState

Implements

IDisposable

Inherited Members

State<GameStatId, GameState, GameStateMachine>.Id
State<GameStatId, GameState, GameStateMachine>.StateMachine
State<GameStatId, GameState, GameStateMachine>.IsReady()
State<GameStatId, GameState, GameStateMachine>.StartContext(StateParam)
State<GameStatId, GameState, GameStateMachine>.OnEndContext()
State<GameStatId, GameState, GameStateMachine>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class GameplayInitState : GameState, IDisposable
```

Properties

Id

Declaration

```
public override GameStateId Id { get; }
```

Property Value

TYPE

GameStatId

Overrides

State<GameStatId, GameState, GameStateMachine>.Id

Methods

StartContext(StateParam)

Declaration

```
public override void StartContext(StateParam stateParam)
```

Parameters

TYPE	NAME
StateParam	stateParam

Overrides

State<GameStatId, GameState, GameStateMachine>.StartContext(StateParam)

Implements

IDisposable

© Bus Fighter. All rights reserved.

Class InfoLoadState

Inheritance

object → State<GameStatId, GameState, GameStateMachine> → GameState → InfoLoadState

Implements

[IDisposable](#)

Inherited Members

State<GameStatId, GameState, GameStateMachine>.Id
State<GameStatId, GameState, GameStateMachine>.StateMachine
State<GameStatId, GameState, GameStateMachine>.IsReady()
State<GameStatId, GameState, GameStateMachine>.StartContext(StateParam)
State<GameStatId, GameState, GameStateMachine>.OnEndContext()
State<GameStatId, GameState, GameStateMachine>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class InfoLoadState : GameState, IDisposable
```

Properties

Id

Declaration

```
public override GameStateId Id { get; }
```

Property Value

TYPE

GameStatId

Overrides

State<GameStatId, GameState, GameStateMachine>.Id

Methods

StartContext(StateParam)

Declaration

```
public override void StartContext(StateParam stateParam)
```

Parameters

TYPE	NAME
StateParam	stateParam

Overrides

State<GameStatId, GameState, GameStateMachine>.StartContext(StateParam)

Implements

IDisposable

© Bus Fighter. All rights reserved.

Class LoadStageState

Inheritance

object → State<GameStatId, GameState, GameStateMachine> → GameState → LoadStageState

Implements

[IDisposable](#)

Inherited Members

[State<GameStatId, GameState, GameStateMachine>.Id](#)
[State<GameStatId, GameState, GameStateMachine>.StateMachine](#)
[State<GameStatId, GameState, GameStateMachine>.IsReady\(\)](#)
[State<GameStatId, GameState, GameStateMachine>.StartContext\(StateParam\)](#)
[State<GameStatId, GameState, GameStateMachine>.OnEndContext\(\)](#)
[State<GameStatId, GameState, GameStateMachine>.Dispose\(\)](#)
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class LoadStageState : GameState, IDisposable
```

Properties

Id

Declaration

```
public override GameStateId Id { get; }
```

Property Value

TYPE

GameStatId

Overrides

[State<GameStatId, GameState, GameStateMachine>.Id](#)

Methods

OnEndContext()

Declaration

```
protected override void OnEndContext()
```

Overrides

[State<GameStatId, GameState, GameStateMachine>.OnEndContext\(\)](#)

StartContext(StateParam)

Declaration

```
public override void StartContext(StateParam param)
```

Parameters

TYPE	NAME
StateParam	param

Overrides

[State<GameStatId, GameState, GameStateMachine>.StartContext\(StateParam\)](#)

Implements

Class LoadStageState.InitParam

Inheritance

object → [StateParam](#) → LoadStageState.InitParam

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class LoadStageState.InitParam : StateParam
```

Fields

stageld

Declaration

```
public string stageId
```

Field Value

TYPE

string

Class LocalLoadState

Inheritance

object → State<GameStatId, GameState, GameStateMachine> → GameState → LocalLoadState

Implements

[IDisposable](#)

Inherited Members

[State<GameStatId, GameState, GameStateMachine>.Id](#)
[State<GameStatId, GameState, GameStateMachine>.StateMachine](#)
[State<GameStatId, GameState, GameStateMachine>.IsReady\(\)](#)
[State<GameStatId, GameState, GameStateMachine>.StartContext\(StateParam\)](#)
[State<GameStatId, GameState, GameStateMachine>.OnEndContext\(\)](#)
[State<GameStatId, GameState, GameStateMachine>.Dispose\(\)](#)
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class LocalLoadState : GameState, IDisposable
```

Properties

Id

Declaration

```
public override GameStateId Id { get; }
```

Property Value

TYPE

GameStatId

Overrides

[State<GameStatId, GameState, GameStateMachine>.Id](#)

Methods

StartContext(StateParam)

Declaration

```
public override void StartContext(StateParam param)
```

Parameters

TYPE	NAME
StateParam	param

Overrides

[State<GameStatId, GameState, GameStateMachine>.StartContext\(StateParam\)](#)

Implements

[IDisposable](#)

© Bus Fighter. All rights reserved.

Class LoginState

Inheritance

object → State<GameStatId, GameState, GameStateMachine> → GameState → LoginState

Implements

[IDisposable](#)

Inherited Members

[State<GameStatId, GameState, GameStateMachine>.Id](#)
[State<GameStatId, GameState, GameStateMachine>.StateMachine](#)
[State<GameStatId, GameState, GameStateMachine>.IsReady\(\)](#)
[State<GameStatId, GameState, GameStateMachine>.StartContext\(StateParam\)](#)
[State<GameStatId, GameState, GameStateMachine>.OnEndContext\(\)](#)
[State<GameStatId, GameState, GameStateMachine>.Dispose\(\)](#)
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class LoginState : GameState, IDisposable
```

Properties

Id

Declaration

```
public override GameStateId Id { get; }
```

Property Value

TYPE

GameStatId

Overrides

State<GameStatId, GameState, GameStateMachine>.Id

Methods

StartContext(StateParam)

Declaration

```
public override void StartContext(StateParam stateParam)
```

Parameters

TYPE	NAME
StateParam	stateParam

Overrides

State<GameStatId, GameState, GameStateMachine>.StartContext(StateParam)

Implements

IDisposable

© Bus Fighter. All rights reserved.

Class LoginState.Param

Inheritance

`object` → `StateParam` → `LoginState.Param`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class LoginState.Param : StateParam
```

Fields

Platform

Declaration

```
public LoginPlatform Platform
```

Field Value

TYPE

`LoginPlatform`

Token

Declaration

```
public LoginToken Token
```

Field Value

TYPE

LoginToken

© Bus Fighter. All rights reserved.

Class UILoadState

Inheritance

object → State<GameStatId, GameState, GameStateMachine> → GameState → UILoadState

Implements

IDisposable

Inherited Members

State<GameStatId, GameState, GameStateMachine>.Id
State<GameStatId, GameState, GameStateMachine>.StateMachine
State<GameStatId, GameState, GameStateMachine>.IsReady()
State<GameStatId, GameState, GameStateMachine>.StartContext(StateParam)
State<GameStatId, GameState, GameStateMachine>.OnEndContext()
State<GameStatId, GameState, GameStateMachine>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class UILoadState : GameState, IDisposable
```

Properties

Id

Declaration

```
public override GameStateId Id { get; }
```

Property Value

TYPE

GameStatId

Overrides

State<GameStatId, GameState, GameStateMachine>.Id

Methods

StartContext(StateParam)

Declaration

```
public override void StartContext(StateParam param)
```

Parameters

TYPE	NAME
StateParam	param

Overrides

State<GameStatId, GameState, GameStateMachine>.StartContext(StateParam)

Implements

IDisposable

© Bus Fighter. All rights reserved.

Class UserDataLoadState

Inheritance

object → State<GameStatId, GameState, GameStateMachine> → GameState → UserDataLoadState

Implements

IDisposable

Inherited Members

State<GameStatId, GameState, GameStateMachine>.Id
State<GameStatId, GameState, GameStateMachine>.StateMachine
State<GameStatId, GameState, GameStateMachine>.IsReady()
State<GameStatId, GameState, GameStateMachine>.StartContext(StateParam)
State<GameStatId, GameState, GameStateMachine>.OnEndContext()
State<GameStatId, GameState, GameStateMachine>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [CatSweeper.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class UserDataLoadState : GameState, IDisposable
```

Properties

Id

Declaration

```
public override GameStateId Id { get; }
```

Property Value

TYPE

GameStatId

Overrides

State<GameStatId, GameState, GameStateMachine>.Id

Methods

StartContext(StateParam)

Declaration

```
public override void StartContext(StateParam stateParam)
```

Parameters

TYPE	NAME
StateParam	stateParam

Overrides

State<GameStatId, GameState, GameStateMachine>.StartContext(StateParam)

Implements

IDisposable

© Bus Fighter. All rights reserved.

Namespace CatSweeper.Info

Classes

[GameSettingInfo](#)

[GameSettingInfoManager](#)

[StageInfo](#)

[StageInfoManager](#)

[StageMapInfo](#)

[StageMapInfoManager](#)

© Bus Fighter. All rights reserved.

Class GameSettingInfo

Inheritance

`object` → GameSettingInfo

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [CatSweeper.Info](#)

Assembly: CatSweeper.dll

Syntax

```
[Serializable]
public class GameSettingInfo
```

Properties

key

Declaration

```
public string key { get; set; }
```

Property Value

TYPE

`string`

Declaration

```
public string value { get; set; }
```

Property Value

TYPE

string

© Bus Fighter. All rights reserved.

Class GameSettingInfoManager

Inheritance

object → [InfoManager](#) → [ConfigInfoManager<string, GameSettingInfo>](#) → GameSettingInfoManager

Implements

[IInfoManager](#)

[IDisposable](#)

Inherited Members

[ConfigInfoManager<string, GameSettingInfo>.valueMap](#)

[ConfigInfoManager<string, GameSettingInfo>.GetAllValue\(\)](#)

[ConfigInfoManager<string, GameSettingInfo>.keyFn](#)

[ConfigInfoManager<string, GameSettingInfo>.infoType](#)

[ConfigInfoManager<string, GameSettingInfo>.LoadInfo\(\)](#)

[ConfigInfoManager<string, GameSettingInfo>.LoadInfoAsync\(CancellationToken\)](#)

[ConfigInfoManager<string, GameSettingInfo>.AddValue\(GameSettingInfo\)](#)

[ConfigInfoManager<string, GameSettingInfo>.TryGetValue\(string, out GameSettingInfo\)](#)

[ConfigInfoManager<string, GameSettingInfo>.Dispose\(\)](#)

[InfoManager.infoType](#)

[InfoManager.GetAllValue\(\)](#)

[InfoManager.LoadInfo\(\)](#)

[InfoManager.LoadInfoAsync\(CancellationToken\)](#)

[InfoManager.OnLoadCompleted\(\)](#)

[InfoManager.Dispose\(\)](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [CatSweeper.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public class GameSettingInfoManager : ConfigInfoManager<string, GameSettingInfo>, IInfoManager
```

Constructors

GameSettingInfoManager(IValueLoader<GameSettingInfo>)

Declaration

```
public GameSettingInfoManager(IValueLoader<GameSettingInfo> loader)
```

Parameters

TYPE	NAME
IValueLoader<GameSettingInfo>	loader

Properties

keyFn

Declaration

```
protected override Func<GameSettingInfo, string> keyFn { get; }
```

Property Value

TYPE
Func<GameSettingInfo, string>

Overrides

[ConfigInfoManager<string, GameSettingInfo>.keyFn](#)

Methods

GetBoolValue(string)

Declaration

```
public Res<bool, Exception> GetBoolValue(string id)
```

Parameters

TYPE	NAME
------	------

string	id
--------	----

Returns

TYPE

Res<bool, Exception>

GetFloatValue(string)

Declaration

```
public Res<float, Exception> GetFloatValue(string id)
```

Parameters

TYPE	NAME
------	------

string	id
--------	----

Returns

TYPE

Res<float, Exception>

GetIntValue(string)

Declaration

```
public Res<int, Exception> GetIntValue(string id)
```

Parameters

TYPE	NAME
------	------

string	id
--------	----

Returns

TYPE

Res<int, Exception>

GetStringValue(string)

Declaration

```
public Res<string, Exception> GetStringValue(string id)
```

Parameters

TYPE	NAME
------	------

string	id
--------	----

Returns

TYPE

Res<string, Exception>

Implements

IInfoManager

IDisposable

© Bus Fighter. All rights reserved.

Class StageInfo

Inheritance

[object](#) → StageInfo

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [CatSweeper.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public class StageInfo
```

Properties

map

Support Direct Size, or StageMapInfo Id reference

1. **Size(10, 10)** - Creates a map of size 10x10
2. **StageMapInfo(map_1)** - References a [StageMapInfo](#) with ID 'map_1'

Declaration

```
public string map { get; set; }
```

Property Value

TYPE

string

scenePath

Declaration

```
public string scenePath { get; set; }
```

Property Value

TYPE

string

stageId

Declaration

```
public string stageId { get; set; }
```

Property Value

TYPE

string

Methods

GetMap()

Declaration

```
public Res<StageMapInfo, Exception> GetMap()
```

Returns

Res<StageMapInfo, Exception>

© Bus Fighter. All rights reserved.

Class StageInfoManager

Inheritance

object → [InfoManager](#) → [ConfigInfoManager<string, StageInfo>](#) → StageInfoManager

Implements

[IInfoManager](#)

[IDisposable](#)

Inherited Members

[ConfigInfoManager<string, StageInfo>.valueMap](#)

[ConfigInfoManager<string, StageInfo>.GetAllValue\(\)](#)

[ConfigInfoManager<string, StageInfo>.keyFn](#)

[ConfigInfoManager<string, StageInfo>.infoType](#)

[ConfigInfoManager<string, StageInfo>.LoadInfo\(\)](#)

[ConfigInfoManager<string, StageInfo>.LoadInfoAsync\(CancellationToken\)](#)

[ConfigInfoManager<string, StageInfo>.AddValue\(StageInfo\)](#)

[ConfigInfoManager<string, StageInfo>.TryGetValue\(string, out StageInfo\)](#)

[ConfigInfoManager<string, StageInfo>.Dispose\(\)](#)

[InfoManager.infoType](#)

[InfoManager.GetAllValue\(\)](#)

[InfoManager.LoadInfo\(\)](#)

[InfoManager.LoadInfoAsync\(CancellationToken\)](#)

[InfoManager.OnLoadCompleted\(\)](#)

[InfoManager.Dispose\(\)](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [CatSweeper.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public class StageInfoManager : ConfigInfoManager<string, StageInfo>, IInfoManager, IDisposable
```

Constructors

StageInfoManager(IValueLoader<StageInfo>)

Declaration

```
public StageInfoManager(IValueLoader<StageInfo> loader)
```

Parameters

TYPE	NAME
IValueLoader<StageInfo>	loader

Properties

keyFn

Declaration

```
protected override Func<StageInfo, string> keyFn { get; }
```

Property Value

TYPE
Func<StageInfo, string>

Overrides

[ConfigInfoManager<string, StageInfo>.keyFn](#)

Implements

[IInfoManager](#)

[IDisposable](#)

Class StageMapInfo

Inheritance

`object` → StageMapInfo

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [CatSweeper.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public class StageMapInfo
```

Properties

floorGrids

Declaration

```
public Vector2I[] floorGrids { get; set; }
```

Property Value

TYPE

`Vector2I[]`

Declaration

```
public string mapId { get; set; }
```

Property Value

TYPE

string

© Bus Fighter. All rights reserved.

Class StageMapInfoManager

Inheritance

object → [InfoManager](#) → [ConfigInfoManager<string, StageMapInfo>](#) → StageMapInfoManager

Implements

[IInfoManager](#)

[IDisposable](#)

Inherited Members

[ConfigInfoManager<string, StageMapInfo>.valueMap](#)

[ConfigInfoManager<string, StageMapInfo>.GetAllValue\(\)](#)

[ConfigInfoManager<string, StageMapInfo>.keyFn](#)

[ConfigInfoManager<string, StageMapInfo>.infoType](#)

[ConfigInfoManager<string, StageMapInfo>.LoadInfo\(\)](#)

[ConfigInfoManager<string, StageMapInfo>.LoadInfoAsync\(CancellationToken\)](#)

[ConfigInfoManager<string, StageMapInfo>.AddValue\(StageMapInfo\)](#)

[ConfigInfoManager<string, StageMapInfo>.TryGetValue\(string, out StageMapInfo\)](#)

[ConfigInfoManager<string, StageMapInfo>.Dispose\(\)](#)

[InfoManager.infoType](#)

[InfoManager.GetAllValue\(\)](#)

[InfoManager.LoadInfo\(\)](#)

[InfoManager.LoadInfoAsync\(CancellationToken\)](#)

[InfoManager.OnLoadCompleted\(\)](#)

[InfoManager.Dispose\(\)](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [CatSweeper.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public class StageMapInfoManager : ConfigInfoManager<string, StageMapInfo>, IInfoManager, IDisposable
```

Constructors

StageMapInfoManager(IValueLoader<StageMapInfo>)

Declaration

```
public StageMapInfoManager(IValueLoader<StageMapInfo> loader)
```

Parameters

TYPE	NAME
IValueLoader<StageMapInfo>	loader

Properties

keyFn

Declaration

```
protected override Func<StageMapInfo, string> keyFn { get; }
```

Property Value

TYPE
Func<StageMapInfo, string>

Overrides

[ConfigInfoManager<string, StageMapInfo>.keyFn](#)

Methods

GetMapBySize(int, int)

Declaration

```
public StageMapInfo GetMapBySize(int width, int height)
```

Parameters

TYPE	NAME
------	------

int	width
-----	-------

int	height
-----	--------

Returns

TYPE

StageMapInfo

Implements

IInfoManager

IDisposable

© Bus Fighter. All rights reserved.

Namespace CatSweeper.Service

Classes

[StageService](#)

[StageServiceModel](#)

Structs

[StageData](#)

© Bus Fighter. All rights reserved.

Struct StageData

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [CatSweeper.Service](#)

Assembly: CatSweeper.dll

Syntax

```
public struct StageData
```

Fields

mapInfo

Declaration

```
public StageMapInfo mapInfo
```

Field Value

TYPE

[StageMapInfo](#)

stageInfo

Declaration

```
public StageInfo stageInfo
```

Field Value

TYPE

[StageInfo](#)

© Bus Fighter. All rights reserved.

Class StageService

Inheritance

[object](#) → StageService

Implements

[IModelService](#)

[IService](#)

[IDisposable](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [CatSweeper.Service](#)

Assembly: CatSweeper.dll

Syntax

```
public class StageService : IModelService, IService, IDisposable
```

Constructors

[StageService\(StageServiceModel, StageInfoManager, StageMapInfoManager\)](#)

Declaration

```
public StageService(StageServiceModel model, StageInfoManager infoManger, StageMapInfoManager
```

Parameters

TYPE	NAME
StageServiceModel	model
StageInfoManager	infoManger
StageMapInfoManager	mapInfoManager

Properties

GetModel

Declaration

```
public IServiceProvider GetModel { get; }
```

Property Value

TYPE

IServiceProvider

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

GetStageData(string)

Declaration

```
public Res<StageData, Exception> GetStageData(string stageId)
```

Parameters

TYPE	NAME
------	------

string	stageId
--------	---------

Returns

TYPE

Res<StageData, Exception>

Implements

[IModelService](#)

[IService](#)

[IDisposable](#)

© Bus Fighter. All rights reserved.

Class StageServiceModel

Inheritance

[object](#) → StageServiceModel

Implements

[IServiceModel](#)

[IRuntimeSavable](#)

[IDisposable](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [CatSweeper.Service](#)

Assembly: CatSweeper.dll

Syntax

```
public class StageServiceModel : IServiceModel, IRuntimeSavable, IDisposable
```

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

Initialize(IUserData)

Declaration

```
public void Initialize(IUserData userData)
```

Parameters

TYPE	NAME
IUserData	userData

SetSaveData(Dictionary<string, object>)

Declaration

```
public void SetSaveData(Dictionary<string, object> dataMap)
```

Parameters

TYPE	NAME
Dictionary<string, object>	dataMap

Implements

[IServiceModel](#)
[IRuntimeSavable](#)
[IDisposable](#)

© Bus Fighter. All rights reserved.

Namespace CatSweeper.Util

Classes

[ExportUtil](#)

[InfoUtil](#)

© Bus Fighter. All rights reserved.

Class ExportUtil

Inheritance

`object` → `ExportUtil`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [CatSweeper.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public static class ExportUtil
```

Methods

MarkReadOnly(Dictionary, StringName)

Declaration

```
public static void MarkReadOnly(this Dictionary property, StringName propertyName)
```

Parameters

TYPE	NAME
Dictionary	property
StringName	propertyName

Class InfoUtil

Inheritance

`object` → `InfoUtil`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [CatSweeper.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public static class InfoUtil
```

Methods

CreateStorage(Type)

Declaration

```
public static IStorage CreateStorage(Type infoType)
```

Parameters

TYPE	NAME
------	------

Type	infoType
------	----------

Returns

© Bus Fighter. All rights reserved.

Namespace CofyDev.Xml.Doc

Classes

[CofyXmlDocParser](#)

[DataContainer](#)

[DataObject](#)

[DataObject.Decoder](#)

[DataObject.ListValueDecoder](#)

[DataObject.StringValueDecoder](#)

[DataObjectEncoder](#)

[DataObjectExtension](#)

Structs

[DataObject.BooleanDecoder](#)

[DataObject.DoubleDecoder](#)

[DataObject.EnumDecoder](#)

[DataObject.FloatDecoder](#)

[DataObject.IntDecoder](#)

[DataObject.StringDecoder](#)

Interfaces

[DataObject.IStringDecoder](#)

[DataObject.IValueDecoder](#)

Class CofyXmlDocParser

Inheritance

[object](#) → CofyXmlDocParser

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [CofyDev.Xml.Doc](#)

Assembly: CatSweeper.dll

Syntax

```
public static class CofyXmlDocParser
```

Methods

ParseExcel(byte[])

Declaration

```
public static DataContainer ParseExcel(byte[] fileBytes)
```

Parameters

TYPE	NAME
------	------

byte[]	fileBytes
--------	-----------

Returns

© Bus Fighter. All rights reserved.

Class DataContainer

Inheritance

object → List<DataObject> → DataContainer

Implements

IList<DataObject>
ICollection<DataObject>
IReadOnlyList<DataObject>
IReadOnlyCollection<DataObject>
IEnumerable<DataObject>
IList
ICollection
IEnumerable

Inherited Members

List<DataObject>.Add(DataObject)
List<DataObject>.AddRange(IEnumerable<DataObject>)
List<DataObject>.AsReadOnly()
List<DataObject>.BinarySearch(int, int, DataObject, IComparer<DataObject>)
List<DataObject>.BinarySearch(DataObject)
List<DataObject>.BinarySearch(DataObject, IComparer<DataObject>)
List<DataObject>.Clear()
List<DataObject>.Contains(DataObject)
List<DataObject>.ConvertAll<TOutput>(Converter<DataObject, TOutput>)
List<DataObject>.CopyTo(int, DataObject[], int, int)
List<DataObject>.CopyTo(DataObject[])
List<DataObject>.CopyTo(DataObject[], int)
List<DataObject>.EnsureCapacity(int)
List<DataObject>.Exists(Predicate<DataObject>)
List<DataObject>.Find(Predicate<DataObject>)
List<DataObject>.FindAll(Predicate<DataObject>)
List<DataObject>.FindIndex(int, int, Predicate<DataObject>)
List<DataObject>.FindIndex(int, Predicate<DataObject>)
List<DataObject>.FindIndex(Predicate<DataObject>)
List<DataObject>.FindLast(Predicate<DataObject>)
List<DataObject>.FindLastIndex(int, int, Predicate<DataObject>)
List<DataObject>.FindLastIndex(int, Predicate<DataObject>)
List<DataObject>.FindLastIndex(Predicate<DataObject>)
List<DataObject>.ForEach(Action<DataObject>)
List<DataObject>.GetEnumerator()

List<DataObject>.GetRange(int, int)
List<DataObject>.IndexOf(DataObject)
List<DataObject>.IndexOf(DataObject, int)
List<DataObject>.IndexOf(DataObject, int, int)
List<DataObject>.Insert(int, DataObject)
List<DataObject>.InsertRange(int, IEnumerable<DataObject>)
List<DataObject>.LastIndexOf(DataObject)
List<DataObject>.LastIndexOf(DataObject, int)
List<DataObject>.LastIndexOf(DataObject, int, int)
List<DataObject>.Remove(DataObject)
List<DataObject>.RemoveAll(Predicate<DataObject>)
List<DataObject>.RemoveAt(int)
List<DataObject>.RemoveRange(int, int)
List<DataObject>.Reverse()
List<DataObject>.Reverse(int, int)
List<DataObject>.Slice(int, int)
List<DataObject>.Sort()
List<DataObject>.Sort(IComparer<DataObject>)
List<DataObject>.Sort(Comparison<DataObject>)
List<DataObject>.Sort(int, int, IComparer<DataObject>)
List<DataObject>.ToArray()
List<DataObject>.TrimExcess()
List<DataObject>.TrueForAll(Predicate<DataObject>)
List<DataObject>.Capacity
List<DataObject>.Count
List<DataObject>.this[int]
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [CofyDev.Xml.Doc](#)

Assembly: [CatSweeper.dll](#)

Syntax

```
public class DataContainer : List<DataObject>, IList<DataObject>, ICollection<DataObject>, IRe
```

Implements

IList<T>
ICollection<T>
IReadOnlyList<T>
IReadOnlyCollection<T>
IEnumerable<T>

IList

ICollection

IEnumerable

Extension Methods

[ListExtension.EnsureCapacity<T>\(List<T>, int\)](#)

© Bus Fighter. All rights reserved.

Class DataObject

Inheritance

object → [Dictionary<string, object>](#) → DataObject

Implements

[IDictionary<string, object>](#)
[ICollection<KeyValuePair<string, object>>](#)
 [IReadOnlyDictionary<string, object>](#)
 [IReadOnlyCollection<KeyValuePair<string, object>>](#)
 [IEnumerable<KeyValuePair<string, object>>](#)
 [IDictionary](#)
 [ICollection](#)
 [IEnumerable](#)
 [IDeserializationCallback](#)
 [ISerializable](#)

Inherited Members

[Dictionary<string, object>.Add\(string, object\)](#)
[Dictionary<string, object>.Clear\(\)](#)
[Dictionary<string, object>.ContainsKey\(string\)](#)
[Dictionary<string, object>.ContainsValue\(object\)](#)
[Dictionary<string, object>.EnsureCapacity\(int\)](#)
[Dictionary<string, object>.GetEnumerator\(\)](#)
[Dictionary<string, object>.OnDeserialization\(object\)](#)
[Dictionary<string, object>.Remove\(string\)](#)
[Dictionary<string, object>.Remove\(string, out object\)](#)
[Dictionary<string, object>.TrimExcess\(\)](#)
[Dictionary<string, object>.TrimExcess\(int\)](#)
[Dictionary<string, object>.TryAdd\(string, object\)](#)
[Dictionary<string, object>.TryGetValue\(string, out object\)](#)
[Dictionary<string, object>.Comparer](#)
[Dictionary<string, object>.Count](#)
[Dictionary<string, object>.this\[string\]](#)
[Dictionary<string, object>.Keys](#)
[Dictionary<string, object>.Values](#)
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)

object.ReferenceEquals(object, object)

object.ToString()

Namespace: [CofyDev.Xml.Doc](#)

Assembly: CatSweeper.dll

Syntax

```
public class DataObject : Dictionary<string, object>, IDictionary<string, object>, ICollection
```

Constructors

DataObject()

Declaration

```
public DataObject()
```

DataObject(int)

Declaration

```
public DataObject(int capacity)
```

Parameters

TYPE	NAME
------	------

int	capacity
-----	----------

Implements

IDictionary<TKey, TValue>

ICollection<T>

IReadOnlyDictionary<TKey, TValue>

IReadOnlyCollection<T>

IEnumerable<T>

IDictionary

ICollection

IEnumerable

[IDeserializationCallback](#)

[ISerializable](#)

© Bus Fighter. All rights reserved.

Struct DataObject.BooleanDecoder

Implements

[DataObject.IStringDecoder](#)

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [CofyDev.Xml.Doc](#)

Assembly: CatSweeper.dll

Syntax

```
public struct DataObject.BooleanDecoder : DataObject.IStringDecoder
```

Properties

propertyType

Declaration

```
public Type propertyType { get; }
```

Property Value

TYPE

[Type](#)

Methods

TryDecode(string, out object)

Declaration

```
public bool TryDecode(string raw, out object decoded)
```

Parameters

TYPE	NAME
string	raw
object	decoded

Returns

TYPE
bool

Implements

[DataObject.IStringDecoder](#)

© Bus Fighter. All rights reserved.

Class DataObject.Decoder

Inheritance

[object](#) → [DataObject.Decoder](#)

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [CofyDev.Xml.Doc](#)

Assembly: [CatSweeper.dll](#)

Syntax

```
public static class DataObject.Decoder
```

Properties

stringDecoders

Declaration

```
public static IReadOnlyDictionary<Type, DataObject.IValueDecoder> stringDecoders { get; }
```

Property Value

TYPE

[IReadOnlyDictionary<Type, DataObject.IValueDecoder>](#)

Methods

RegisterDecoder(IValueDecoder)

Declaration

```
public static void RegisterDecoder(DataObject.IValueDecoder decoder)
```

Parameters

TYPE	NAME
DataObject.IValueDecoder	decoder

TryDecode(object, Type, out object)

Declaration

```
public static bool TryDecode(object raw, Type decodedType, out object decoded)
```

Parameters

TYPE	NAME
object	raw
Type	decodedType
object	decoded

Returns

TYPE
bool

TryGetDecoder(Type, out IValueDecoder)

Declaration

```
public static bool TryGetDecoder(Type type, out DataObject.IValueDecoder decoder)
```

Parameters

TYPE	NAME
Type	type
DataObject.IValueDecoder	decoder

Returns

TYPE

bool

© Bus Fighter. All rights reserved.

Struct DataObject.DoubleDecoder

Implements

[DataObject.IStringDecoder](#)

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [CofyDev.Xml.Doc](#)

Assembly: CatSweeper.dll

Syntax

```
public struct DataObject.DoubleDecoder : DataObject.IStringDecoder
```

Properties

propertyType

Declaration

```
public Type propertyType { get; }
```

Property Value

TYPE

[Type](#)

Methods

TryDecode(string, out object)

Declaration

```
public bool TryDecode(string raw, out object decoded)
```

Parameters

TYPE	NAME
string	raw
object	decoded

Returns

TYPE
bool

Implements

[DataObject.IStringDecoder](#)

© Bus Fighter. All rights reserved.

Struct DataObject.EnumDecoder

Implements

[DataObject.IStringDecoder](#)

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [CofyDev.Xml.Doc](#)

Assembly: CatSweeper.dll

Syntax

```
public struct DataObject.EnumDecoder : DataObject.IStringDecoder
```

Constructors

EnumDecoder(Type)

Declaration

```
public EnumDecoder(Type enumType)
```

Parameters

TYPE	NAME
------	------

Type	enumType
------	----------

Properties

propertyType

Declaration

```
public Type propertyType { get; }
```

Property Value

TYPE

Type

Methods

TryDecode(string, out object)

Declaration

```
public bool TryDecode(string raw, out object decoded)
```

Parameters

TYPE NAME

string raw

object decoded

Returns

TYPE

bool

Implements

[DataObject.IStringDecoder](#)

Struct DataObject.FloatDecoder

Implements

[DataObject.IStringDecoder](#)

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [CofyDev.Xml.Doc](#)

Assembly: CatSweeper.dll

Syntax

```
public struct DataObject.FloatDecoder : DataObject.IStringDecoder
```

Properties

propertyType

Declaration

```
public Type propertyType { get; }
```

Property Value

TYPE

[Type](#)

Methods

TryDecode(string, out object)

Declaration

```
public bool TryDecode(string raw, out object decoded)
```

Parameters

TYPE	NAME
string	raw
object	decoded

Returns

TYPE
bool

Implements

[DataObject.IStringDecoder](#)

© Bus Fighter. All rights reserved.

Interface DataObject.IStringDecoder

Namespace: [CofyDev.Xml.Doc](#)

Assembly: CatSweeper.dll

Syntax

```
public interface DataObject.IStringDecoder
```

Properties

propertyType

Declaration

```
Type propertyType { get; }
```

Property Value

TYPE

Type

Methods

TryDecode(string, out object)

Declaration

```
bool TryDecode(string raw, out object decoded)
```

Parameters

TYPE	NAME
------	------

string	raw
--------	-----

object	decoded
--------	---------

Returns

TYPE

bool

© Bus Fighter. All rights reserved.

Interface DataObject.IValueDecoder

Namespace: [CofyDev.Xml.Doc](#)

Assembly: CatSweeper.dll

Syntax

```
public interface DataObject.IValueDecoder
```

Properties

valueType

Declaration

```
Type valueType { get; }
```

Property Value

TYPE

Type

Methods

TryDecode(object, Type, out object)

Declaration

```
bool TryDecode(object raw, Type decodedType, out object decoded)
```

Parameters

TYPE **NAME**

object	raw
Type	decodedType
object	decoded

Returns**TYPE**

bool

© Bus Fighter. All rights reserved.

Struct DataObject.IntDecoder

Implements

[DataObject.IStringDecoder](#)

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [CofyDev.Xml.Doc](#)

Assembly: CatSweeper.dll

Syntax

```
public struct DataObject.IntDecoder : DataObject.IStringDecoder
```

Properties

propertyType

Declaration

```
public Type propertyType { get; }
```

Property Value

TYPE

[Type](#)

Methods

TryDecode(string, out object)

Declaration

```
public bool TryDecode(string raw, out object decoded)
```

Parameters

TYPE	NAME
string	raw
object	decoded

Returns

TYPE
bool

Implements

[DataObject.IStringDecoder](#)

© Bus Fighter. All rights reserved.

Class DataObject.ListValueDecoder

Inheritance

`object` → `DataObject.ListValueDecoder`

Implements

`DataObject.IValueDecoder`

Inherited Members

`object.Equals(object)`

`object.Equals(object, object)`

`object.GetHashCode()`

`object.GetType()`

`object.MemberwiseClone()`

`object.ReferenceEquals(object, object)`

`object.ToString()`

Namespace: `CofyDev.Xml.Doc`

Assembly: `CatSweeper.dll`

Syntax

```
public class DataObject.ListValueDecoder : DataObject.IValueDecoder
```

Properties

valueType

Declaration

```
public Type valueType { get; }
```

Property Value

TYPE

`Type`

Methods

TryDecode(object, Type, out object)

Declaration

```
public bool TryDecode(object raw, Type decodedType, out object decoded)
```

Parameters

TYPE	NAME
object	raw
Type	decodedType
object	decoded

Returns

TYPE
bool

Implements

DataObject.IValueDecoder

© Bus Fighter. All rights reserved.

Struct DataObject.StringDecoder

Implements

[DataObject.IStringDecoder](#)

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [CofyDev.Xml.Doc](#)

Assembly: CatSweeper.dll

Syntax

```
public struct DataObject.StringDecoder : DataObject.IStringDecoder
```

Properties

propertyType

Declaration

```
public Type propertyType { get; }
```

Property Value

TYPE

[Type](#)

Methods

TryDecode(string, out object)

Declaration

```
public bool TryDecode(string raw, out object decoded)
```

Parameters

TYPE	NAME
string	raw
object	decoded

Returns

TYPE
bool

Implements

[DataObject.IStringDecoder](#)

© Bus Fighter. All rights reserved.

Class DataObject.StringValueDecoder

Inheritance

[object](#) → [DataObject.StringValueDecoder](#)

Implements

[DataObject.IValueDecoder](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [CofyDev.Xml.Doc](#)

Assembly: [CatSweeper.dll](#)

Syntax

```
public class DataObject.StringValueDecoder : DataObject.IValueDecoder
```

Constructors

StringValueDecoder()

Declaration

```
public StringValueDecoder()
```

Properties

stringDecoders

Declaration

```
public static IReadOnlyDictionary<Type, DataObject.IStringDecoder> stringDecoders { get; }
```

Property Value

TYPE

IReadOnlyDictionary<Type, DataObject.IStringDecoder>

valueType

Declaration

```
public Type valueType { get; }
```

Property Value

TYPE

Type

Methods

RegisterStringDecoder(IStringDecoder)

Declaration

```
public static void RegisterStringDecoder(DataObject.IStringDecoder decoder)
```

Parameters

TYPE	NAME
DataObject.IStringDecoder	decoder

TryDecode(object, Type, out object)

Declaration

```
public bool TryDecode(object raw, Type decodedType, out object decoded)
```

Parameters

TYPE	NAME
------	------

object	raw
--------	-----

Type	decodedType
------	-------------

object	decoded
--------	---------

Returns

TYPE

bool

TryGetStringDecoder(Type, out IStringDecoder)

Declaration

```
public static bool TryGetStringDecoder(Type type, out DataObject.IStringDecoder decoder)
```

Parameters

TYPE	NAME
------	------

Type	type
------	------

DataObject.IStringDecoder	decoder
---------------------------	---------

Returns

TYPE

bool

Implements

[DataObject.IValueDecoder](#)

Class DataObjectEncoder

Inheritance

[object](#) → [DataObjectEncoder](#)

Implements

[IDisposable](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [CofyDev.Xml.Doc](#)

Assembly: [CatSweeper.dll](#)

Syntax

```
public class DataObjectEncoder : IDisposable
```

Methods

[DecodeAs<T>\(DataObject, Action< PropertyInfo, object, KeyValuePair<string, object>>\)](#)

Declaration

```
public virtual T DecodeAs<T>(DataObject dataObject, Action< PropertyInfo, object, KeyValuePair<
```

Parameters

TYPE**NAME**

DataObject

dataObject

Action<PropertyInfo, object, KeyValuePair<string, object>>

propertyDecodeSetter

Returns**TYPE**

T

Type Parameters**NAME**

T

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

Encode(object)

Declaration

```
public virtual DataObject Encode(object obj)
```

Parameters**TYPE NAME**

object obj

Returns**TYPE**

DataObject

Implements

[IDisposable](#)

© Bus Fighter. All rights reserved.

Class DataObjectExtension

Inheritance

[object](#) → DataObjectExtension

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [CofyDev.Xml.Doc](#)

Assembly: CatSweeper.dll

Syntax

```
public static class DataObjectExtension
```

Methods

SetDecodePropertyValue(PropertyInfo, object, KeyValuePair<string, object>)

Declaration

```
public static void SetDecodePropertyValue(PropertyInfo propertyInfo, object propertyObject, Ke
```

Parameters

TYPE	NAME
 PropertyInfo	propertyInfo
 object	propertyObject

KeyValuePair<string, object> kvp

© Bus Fighter. All rights reserved.

Namespace cfEngine

Classes

[Optional](#)

[Res](#)

[Validation](#)

Structs

[Failure<T>](#)

[Optional<T>](#)

[Pending<T>](#)

[Res<TOk, TErr>](#)

[Success<T>](#)

Interfaces

[Validation<T>](#)

Enums

[ValidationState](#)

Struct Failure<T>

Implements

[Validation<T>](#)

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfEngine](#)

Assembly: CatSweeper.dll

Syntax

```
public readonly struct Failure<T> : Validation<T>
```

Type Parameters

NAME

T

Constructors

Failure(Exception)

Declaration

```
public Failure(Exception exception)
```

Parameters

TYPE NAME

Exception exception

Properties

state

Declaration

```
public ValidationState state { get; }
```

Property Value

TYPE

ValidationState

Methods

GetException()

Declaration

```
public Exception GetException()
```

Returns

TYPE

Exception

GetValue()

Declaration

```
public T GetValue()
```

Returns

TYPE

T

Implements

[Validation<T>](#)

© Bus Fighter. All rights reserved.

Class Optional

Inheritance

`object` → `Optional`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfEngine`

Assembly: `CatSweeper.dll`

Syntax

```
public static class Optional
```

Methods

`None<T>()`

Declaration

```
public static Optional<T> None<T>() where T : class
```

Returns

TYPE

`Optional<T>`

Type Parameters

Some<T>(T)

Declaration

```
public static Optional<T> Some<T>(T value) where T : class
```

Parameters

TYPE	NAME
------	------

T	value
---	-------

Returns

TYPE

Optional<T>

Type Parameters

NAME

T

© Bus Fighter. All rights reserved.

Struct Optional<T>

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfEngine](#)

Assembly: CatSweeper.dll

Syntax

```
public readonly struct Optional<T>
```

Type Parameters

NAME

T

Methods

None()

Declaration

```
public static Optional<T> None()
```

Returns

TYPE

[Optional<T>](#)

Some(T)

Declaration

```
public static Optional<T> Some(T value)
```

Parameters

TYPE	NAME
------	------

T	value
---	-------

Returns

TYPE

Optional<T>

ToString()

Returns the fully qualified type name of this instance.

Declaration

```
public override string ToString()
```

Returns

TYPE	DESCRIPTION
------	-------------

string	The fully qualified type name.
--------	--------------------------------

Overrides

[ValueType.ToString\(\)](#)

TryGetValue(out T)

Declaration

```
public bool TryGetValue(out T value)
```

Parameters

TYPE NAME

T value

Returns

TYPE

bool

Operators

implicit operator Optional<T>(T)

Declaration

```
public static implicit operator Optional<T>(T value)
```

Parameters

TYPE NAME

T value

Returns

TYPE

Optional<T>

© Bus Fighter. All rights reserved.

Struct Pending<T>

Implements

[Validation<T>](#)

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfEngine](#)

Assembly: CatSweeper.dll

Syntax

```
public readonly struct Pending<T> : Validation<T>
```

Type Parameters

NAME

T

Constructors

Pending(T)

Declaration

```
public Pending(T value)
```

Parameters

TYPE NAME

T value

Properties

state

Declaration

```
public ValidationState state { get; }
```

Property Value

TYPE

ValidationState

Methods

GetException()

Declaration

```
public Exception GetException()
```

Returns

TYPE

Exception

GetValue()

Declaration

```
public T GetValue()
```

Returns

TYPE

T

Implements

[Validation<T>](#)

© Bus Fighter. All rights reserved.

Class Res

Inheritance

object → Res

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine](#)

Assembly: CatSweeper.dll

Syntax

```
public static class Res
```

Methods

Err<T>(Exception)

Declaration

```
public static Res<T, Exception> Err<T>(Exception error)
```

Parameters

TYPE	NAME
Exception	error

Returns

TYPE

Res<T, Exception>

Type Parameters

NAME

T

Ok<T>(T)

Declaration

```
public static Res<T, Exception> Ok<T>(T value)
```

Parameters

TYPE NAME

T value

Returns

TYPE

Res<T, Exception>

Type Parameters

NAME

T

© Bus Fighter. All rights reserved.

Struct Res<T0k, TErr>

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfEngine](#)

Assembly: CatSweeper.dll

Syntax

```
public readonly struct Res<T0k, TErr>
```

Type Parameters

NAME

T0k

TErr

Properties

IsErr

Declaration

```
public bool IsErr { get; }
```

Property Value

TYPE

bool

IsOk

Declaration

```
public bool IsOk { get; }
```

Property Value

TYPE

bool

error

Declaration

```
public TErr error { get; }
```

Property Value

TYPE

TErr

value

Declaration

```
public TOK value { get; }
```

Property Value

TYPE

TOK

Methods

Err(TErr)

Declaration

```
public static Res<T0k, TErr> Err(TErr error)
```

Parameters

TYPE	NAME
------	------

TErr	error
------	-------

Returns

TYPE

Res<T0k, TErr>

MapErr<T>(Func<TErr, T>)

Declaration

```
public Res<T0k, T> MapErr<T>(Func<TErr, T> mapper)
```

Parameters

TYPE	NAME
------	------

Func<TErr, T>	mapper
---------------	--------

Returns

TYPE

Res<T0k, T>

Type Parameters

NAME

T

Map<T>(Func<T0k, T>)

Declaration

```
public Res<T, TErr> Map<T>(Func<T0k, T> mapper)
```

Parameters

TYPE	NAME
Func<T0k, T>	mapper

Returns

TYPE
Res<T, TErr>

Type Parameters

NAME
T

Ok(T0k)

Declaration

```
public static Res<T0k, TErr> Ok(T0k value)
```

Parameters

TYPE	NAME
T0k	value

Returns

TYPE
Res<T0k, TErr>

OnErr(Action<TErr>)

Declaration

```
public Res<T0k, TErr> OnErr(Action<TErr> action)
```

Parameters

TYPE NAME

Action<TErr> action

Returns

TYPE

Res<TOk, TErr>

OnOk(Action<TOk>)

Declaration

```
public Res<TOk, TErr> OnOk(Action<TOk> action)
```

Parameters

TYPE NAME

Action<TOk> action

Returns

TYPE

Res<TOk, TErr>

ToString()

Returns the fully qualified type name of this instance.

Declaration

```
public override string ToString()
```

Returns

TYPE DESCRIPTION

string The fully qualified type name.

Overrides

[ValueType.ToString\(\)](#)

TryGetError(out TErr)

Declaration

```
public bool TryGetError(out TErr error)
```

Parameters

TYPE	NAME
------	------

TERr	error
------	-------

Returns

TYPE

bool

TryGetValue(out TOk)

Declaration

```
public bool TryGetValue(out TOk value)
```

Parameters

TYPE	NAME
------	------

TOk	value
-----	-------

Returns

TYPE

bool

© Bus Fighter. All rights reserved.

Struct Success<T>

Implements

[Validation<T>](#)

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfEngine](#)

Assembly: CatSweeper.dll

Syntax

```
public readonly struct Success<T> : Validation<T>
```

Type Parameters

NAME

T

Constructors

Success(T)

Declaration

```
public Success(T value)
```

Parameters

TYPE NAME

T value

Properties

state

Declaration

```
public ValidationState state { get; }
```

Property Value

TYPE

ValidationState

Methods

GetException()

Declaration

```
public Exception GetException()
```

Returns

TYPE

Exception

GetValue()

Declaration

```
public T GetValue()
```

Returns

TYPE

T

Implements

[Validation<T>](#)

© Bus Fighter. All rights reserved.

Class Validation

Inheritance

`object` → Validation

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfEngine`

Assembly: CatSweeper.dll

Syntax

```
public static class Validation
```

Methods

ContinueWith<T>(Task<Validation<T>>, Action<Validation<T>>)

Declaration

```
public static Task<Validation<T>> ContinueWith<T>(this Task<Validation<T>> task, Action<Validation<T>> continuation)
```

Parameters

TYPE	NAME
<code>Task<Validation<T>></code>	<code>task</code>
<code>Action<Validation<T>></code>	<code>continuation</code>

Returns

TYPE

Task<Validation<T>>

Type Parameters

NAME

T

Failure<T>(Exception)

Declaration

```
public static Validation<T> Failure<T>(Exception exception)
```

Parameters

TYPE	NAME
------	------

Exception	exception
-----------	-----------

Returns

TYPE

Validation<T>

Type Parameters

NAME

T

Pending<T>(T)

Declaration

```
public static Validation<T> Pending<T>(T value)
```

Parameters

TYPE	NAME
------	------

T	value
---	-------

Returns

TYPE

Validation<T>

Type Parameters

NAME

T

Success<T>(T)

Declaration

```
public static Validation<T> Success<T>(T value)
```

Parameters

TYPE NAME

T value

Returns

TYPE

Validation<T>

Type Parameters

NAME

T

© Bus Fighter. All rights reserved.

Enum ValidationState

Namespace: [cfEngine](#)

Assembly: CatSweeper.dll

Syntax

```
public enum ValidationState : byte
```

Fields

NAME
Failure
Pending
Success

Extension Methods

[EnumExtension.hasFlag\(Enum, Enum\)](#)

© Bus Fighter. All rights reserved.

Interface Validation<T>

Namespace: [cfEngine](#)

Assembly: CatSweeper.dll

Syntax

```
public interface Validation<out T>
```

Type Parameters

NAME

T

Properties

state

Declaration

```
ValidationState state { get; }
```

Property Value

TYPE

[ValidationState](#)

Methods

GetException()

Declaration

`Exception GetException()`

Returns

TYPE

Exception

GetValue()

Declaration

`T GetValue()`

Returns

TYPE

T

© Bus Fighter. All rights reserved.

Namespace cfEngine.Asset

Classes

[AssetHandle](#)

[AssetHandle<T>](#)

[AssetManager<TBaseObject>](#)

© Bus Fighter. All rights reserved.

Class AssetHandle

Inheritance

object → AssetHandle → [AssetHandle<T>](#)

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Asset](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class AssetHandle
```

Constructors

AssetHandle(Action)

Declaration

```
public AssetHandle(Action releaseAction)
```

Parameters

TYPE	NAME
Action	releaseAction

Fields

ReleaseAction

Declaration

```
public readonly Action ReleaseAction
```

Field Value

TYPE

Action

© Bus Fighter. All rights reserved.

Class AssetHandle<T>

Inheritance

object → [AssetHandle](#) → AssetHandle<T>

Inherited Members

[AssetHandle.ReleaseAction](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.Asset](#)

Assembly: CatSweeper.dll

Syntax

```
public class AssetHandle<T> : AssetHandle where T : class
```

Type Parameters

NAME

T

Constructors

AssetHandle(T, Action)

Declaration

```
public AssetHandle(T asset, Action releaseAction)
```

Parameters

TYPE**NAME**

T asset

Action releaseAction

Fields

Asset

Declaration

```
public readonly WeakReference<T> Asset
```

Field Value

TYPE

WeakReference<T>

© Bus Fighter. All rights reserved.

Class AssetManager<TBaseObject>

Inheritance

[object](#) → [AssetManager<TBaseObject>](#) → [ResourceAssetManager](#)

Implements

[IService](#)

[IDisposable](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.Asset](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class AssetManager<TBaseObject> : IService, IDisposable where TBaseObject : cl
```

Type Parameters

NAME

TBaseObject

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

LoadAsync<T>(string, CancellationToken)

Declaration

```
public Task<T> LoadAsync<T>(string path, CancellationToken token = default) where T : class, T
```

Parameters

TYPE	NAME
string	path
CancellationToken	token

Returns

TYPE
Task<T>

Type Parameters

NAME
T

Load<T>(string)

Declaration

```
public T Load<T>(string path) where T : class, TBaseObject
```

Parameters

TYPE	NAME
string	path

Returns

TYPE
T

Type Parameters

NAME

T

TryGetAsset<T>(string, out T)

Declaration

```
public bool TryGetAsset<T>(string path, out T asset) where T : class, TBaseObject
```

Parameters

TYPE NAME

string path

T asset

Returns

TYPE

bool

Type Parameters

NAME

T

_LoadAsync<T>(string, CancellationToken)

Declaration

```
protected abstract Task<AssetHandle<T>> _LoadAsync<T>(string path, CancellationToken token = d
```

Parameters

TYPE NAME

string path

CancellationToken token

Returns

TYPE

Task<AssetHandle<T>>

Type Parameters

NAME

T

_Load<T>(string)

Declaration

```
protected abstract AssetHandle<T> _Load<T>(string path) where T : class, TBaseObject
```

Parameters

TYPE NAME

string path

Returns

TYPE

AssetHandle<T>

Type Parameters

NAME

T

Implements

IService

IDisposable

© Bus Fighter. All rights reserved.

Namespace cfEngine.Command

Classes

[CommandService](#)

[CommandService.RegisterOnInitializedAttribute](#)

[ICommand.HintAttribute](#)

Interfaces

[ICommand](#)

© Bus Fighter. All rights reserved.

Class CommandService

Inheritance

[object](#) → CommandService

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Command](#)

Assembly: CatSweeper.dll

Syntax

```
public static class CommandService
```

Fields

commandMap

Declaration

```
public static Dictionary<string[], ICommand> commandMap
```

Field Value

TYPE

[Dictionary<string\[\], ICommand>](#)

Methods

RegisterCommand(ICommand, params string[])

Declaration

```
public static void RegisterCommand(ICommand resolver, params string[] commandKeys)
```

Parameters

TYPE	NAME
ICommand	resolver
string[]	commandKeys

UnregisterCommand(params string[])

Declaration

```
public static void UnregisterCommand(params string[] command)
```

Parameters

TYPE	NAME
string[]	command

© Bus Fighter. All rights reserved.

Class CommandService.RegisterOnInitializedAttribute

Inheritance

object → [Attribute](#) → CommandService.RegisterOnInitializedAttribute

Inherited Members

[Attribute.Equals\(object\)](#)
[Attribute.GetCustomAttribute\(Assembly, Type\)](#)
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#)
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#)
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#)
[Attribute.GetCustomAttribute\(Module, Type\)](#)
[Attribute.GetCustomAttribute\(Module, Type, bool\)](#)
[Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#)
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#)
[Attribute.GetCustomAttributes\(Assembly\)](#)
[Attribute.GetCustomAttributes\(Assembly, bool\)](#)
[Attribute.GetCustomAttributes\(Assembly, Type\)](#)
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#)
[Attribute.GetCustomAttributes\(MemberInfo\)](#)
[Attribute.GetCustomAttributes\(MemberInfo, bool\)](#)
[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#)
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#)
[Attribute.GetCustomAttributes\(Module\)](#)
[Attribute.GetCustomAttributes\(Module, bool\)](#)
[Attribute.GetCustomAttributes\(Module, Type\)](#)
[Attribute.GetCustomAttributes\(Module, Type, bool\)](#)
[Attribute.GetCustomAttributes\(ParameterInfo\)](#)
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#)
[Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#)
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#)
[Attribute.GetHashCode\(\)](#)
[Attribute.IsDefaultAttribute\(\)](#)
[Attribute.IsDefined\(Assembly, Type\)](#)
[Attribute.IsDefined\(Assembly, Type, bool\)](#)
[Attribute.IsDefined\(MemberInfo, Type\)](#)
[Attribute.IsDefined\(MemberInfo, Type, bool\)](#)
[Attribute.IsDefined\(Module, Type\)](#)
[Attribute.IsDefined\(Module, Type, bool\)](#)
[Attribute.IsDefined\(ParameterInfo, Type\)](#)

Attribute.IsDefined(ParameterInfo, Type, bool)

Attribute.Match(object)

Attribute.TypeId

object.Equals(object, object)

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: **cfEngine.Command**

Assembly: CatSweeper.dll

Syntax

```
public class CommandService.RegisterOnInitializedAttribute : Attribute
```

Constructors

RegisterOnInitializedAttribute(string)

Declaration

```
public RegisterOnInitializedAttribute(string registerMethodName)
```

Parameters

TYPE	NAME
------	------

string	registerMethodName
--------	--------------------

Fields

registerMethodName

Declaration

```
public readonly string registerMethodName
```

Field Value

© Bus Fighter. All rights reserved.

Interface ICommand

Namespace: [cfEngine.Command](#)

Assembly: CatSweeper.dll

Syntax

```
public interface ICommand
```

Methods

Execute(IReadOnlyDictionary<string, string>)

Declaration

```
void Execute(IReadOnlyDictionary<string, string> args)
```

Parameters

TYPE	NAME
IReadOnlyDictionary<string, string>	args

© Bus Fighter. All rights reserved.

Class ICommand.HintAttribute

Inheritance

object → [Attribute](#) → ICommand.HintAttribute

Inherited Members

[Attribute.Equals\(object\)](#)
[Attribute.GetCustomAttribute\(Assembly, Type\)](#)
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#)
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#)
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#)
[Attribute.GetCustomAttribute\(Module, Type\)](#)
[Attribute.GetCustomAttribute\(Module, Type, bool\)](#)
[Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#)
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#)
[Attribute.GetCustomAttributes\(Assembly\)](#)
[Attribute.GetCustomAttributes\(Assembly, bool\)](#)
[Attribute.GetCustomAttributes\(Assembly, Type\)](#)
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#)
[Attribute.GetCustomAttributes\(MemberInfo\)](#)
[Attribute.GetCustomAttributes\(MemberInfo, bool\)](#)
[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#)
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#)
[Attribute.GetCustomAttributes\(Module\)](#)
[Attribute.GetCustomAttributes\(Module, bool\)](#)
[Attribute.GetCustomAttributes\(Module, Type\)](#)
[Attribute.GetCustomAttributes\(Module, Type, bool\)](#)
[Attribute.GetCustomAttributes\(ParameterInfo\)](#)
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#)
[Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#)
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#)
[Attribute.GetHashCode\(\)](#)
[Attribute.IsDefaultAttribute\(\)](#)
[Attribute.IsDefined\(Assembly, Type\)](#)
[Attribute.IsDefined\(Assembly, Type, bool\)](#)
[Attribute.IsDefined\(MemberInfo, Type\)](#)
[Attribute.IsDefined\(MemberInfo, Type, bool\)](#)
[Attribute.IsDefined\(Module, Type\)](#)
[Attribute.IsDefined\(Module, Type, bool\)](#)
[Attribute.IsDefined\(ParameterInfo, Type\)](#)
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#)
[Attribute.Match\(object\)](#)

`Attribute.TypeId`
`object.Equals(object, object)`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfEngine.Command`

Assembly: `CatSweeper.dll`

Syntax

```
public class ICommand.HintAttribute : Attribute
```

Constructors

HintAttribute(string)

Declaration

```
public HintAttribute(string description)
```

Parameters

TYPE	NAME
------	------

<code>string</code>	<code>description</code>
---------------------	--------------------------

Fields

description

Declaration

```
public readonly string description
```

Field Value

TYPE

<code>string</code>

Namespace cfEngine.Core

Classes

[Game](#)

[GameExtension](#)

[ServiceName](#)

[UserDataKey](#)

[UserDataManager](#)

[UserDataManager.JsonContextMap](#)

Structs

[UserDataManager.DeleteSaveCommand](#)

Interfaces

[IRuntimeSavable](#)

[IUserData](#)

Class Game

Inheritance

object → ServiceLocator → Game

Implements

IServiceLocator
IEnumerable<IService>
IEnumerable
IDisposable

Inherited Members

ServiceLocator.Register<T>(T, string)
ServiceLocator.Unregister<T>(T)
ServiceLocator.Unregister(string)
ServiceLocator.GetService<T>()
ServiceLocator.GetService<T>(string)
ServiceLocator.Dispose()
ServiceLocator.GetEnumerator()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfEngine.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class Game : ServiceLocator, IServiceLocator, IEnumerable<IService>, IEnumerable, IDisp
```

Properties

Current

Declaration

```
public static Game Current { get; }
```

Property Value

TYPE

Game

TaskToken

Declaration

```
public static CancellationToken TaskToken { get; }
```

Property Value

TYPE

CancellationToken

Methods

SetCurrent(Game)

Declaration

```
public static void SetCurrent(Game game)
```

Parameters

TYPE NAME

Game game

Implements

IServiceLocator

IEnumerable<T>

Extension Methods

```
GameExtension.GetStageService(Game)
GameExtension.WithStage(Game, StageService)
GameExtension.GetAsset(Game)
GameExtension.GetAuth(Game)
GameExtension.GetGameStateMachine(Game)
GameExtension.GetInfo(Game)
GameExtension.GetInventory(Game)
GameExtension.GetPoolManager(Game)
GameExtension.GetPool<T>(Game, string)
GameExtension.GetSceneManager<TScene>(Game)
GameExtension.GetUserData(Game)
GameExtension.WithAsset(Game, ResourceAssetManager)
GameExtension.WithAuthService(Game, IAuthService)
GameExtension.WithGameStateMachine(Game, GameStateMachine)
GameExtension.WithInfo(Game, InfoLayer)
GameExtension.WithInventory(Game, IInventoryService)
GameExtension.WithPoolManager(Game, PoolManager)
GameExtension.WithSceneManager<TScene>(Game, ISceneManager<TScene>)
GameExtension.WithUserData(Game, UserManager)
```

© Bus Fighter. All rights reserved.

Class GameExtension

Inheritance

`object` → GameExtension

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfEngine.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public static class GameExtension
```

Methods

GetAsset(Game)

Declaration

```
public static AssetManager<Resource> GetAsset(this Game game)
```

Parameters

TYPE	NAME
<code>Game</code>	<code>game</code>

Returns

TYPE

AssetManager<Resource>

GetAuth(Game)

Declaration

```
public static IAuthService GetAuth(this Game game)
```

Parameters

TYPE **NAME**

Game game

Returns

TYPE

IAuthService

GetGameStateMachine(Game)

Declaration

```
public static GameStateMachine GetGameStateMachine(this Game game)
```

Parameters

TYPE **NAME**

Game game

Returns

TYPE

GameStateMachine

GetInfo(Game)

Declaration

```
public static InfoLayer GetInfo(this Game game)
```

Parameters

TYPE	NAME
------	------

Game	game
------	------

Returns

TYPE

InfoLayer

GetInventory(Game)

Declaration

```
public static InventoryService GetInventory(this Game game)
```

Parameters

TYPE	NAME
------	------

Game	game
------	------

Returns

TYPE

InventoryService

GetPoolManager(Game)

Declaration

```
public static PoolManager GetPoolManager(this Game game)
```

Parameters

TYPE	NAME
------	------

Game	game
------	------

Returns

TYPE

PoolManager

GetPool<T>(Game, string)

Declaration

```
public static Res<T, Exception> GetPool<T>(this Game game, string poolKey) where T : IObjectPo
```

Parameters

TYPE	NAME
------	------

Game	game
------	------

string	poolKey
--------	---------

Returns

TYPE

Res<T, Exception>

Type Parameters

NAME

T

GetSceneManager<TScene>(Game)

Declaration

```
public static ISceneManager<TScene> GetSceneManager<TScene>(this Game game)
```

Parameters

TYPE	NAME
------	------

Game	game
------	------

Returns

TYPE

ISceneManager<TScene>

Type Parameters**NAME**

TScene

GetUserData(Game)

Declaration

```
public static UserManager GetUserData(this Game game)
```

Parameters**TYPE NAME**

Game game

Returns**TYPE**

UserManager

WithAsset(Game, ResourceAssetManager)

Declaration

```
public static Game WithAsset(this Game game, ResourceAssetManager assetManager)
```

Parameters**TYPE****NAME**

Game game

ResourceAssetManager assetManager

Returns

TYPE

Game

WithAuthService(Game, IAuthService)

Declaration

```
public static Game WithAuthService(this Game game, IAuthService service)
```

Parameters

TYPE	NAME
Game	game
IAuthService	service

Returns

TYPE
Game

WithGameStateMachine(Game, GameStateMachine)

Declaration

```
public static Game WithGameStateMachine(this Game game, GameStateMachine service)
```

Parameters

TYPE	NAME
Game	game
GameStateMachine	service

Returns

TYPE
Game

WithInfo(Game, InfoLayer)

Declaration

```
public static Game WithInfo(this Game game, InfoLayer service)
```

Parameters

TYPE	NAME
Game	game
InfoLayer	service

Returns

TYPE
Game

WithInventory(Game, IInventoryService)

Declaration

```
public static Game WithInventory(this Game game, IInventoryService service)
```

Parameters

TYPE	NAME
Game	game
IInventoryService	service

Returns

TYPE
Game

WithPoolManager(Game, PoolManager)

Declaration

```
public static Game WithPoolManager(this Game game, PoolManager service)
```

Parameters

TYPE	NAME
Game	game
PoolManager	service

Returns

TYPE
Game

WithSceneManager<TScene>(Game, ISceneManager<TScene>)

Declaration

```
public static Game WithSceneManager<TScene>(this Game game, ISceneManager<TScene> sceneManager)
```

Parameters

TYPE	NAME
Game	game
ISceneManager<TScene>	sceneManager

Returns

TYPE
Game

Type Parameters

NAME
TScene

WithData(Game, UserDataManager)

Declaration

```
public static Game WithUserData(this Game game, UserDataManager service)
```

Parameters

TYPE	NAME
Game	game
UserDataManager	service

Returns

TYPE
Game

© Bus Fighter. All rights reserved.

Interface IRuntimeSavable

Inherited Members

[IDisposable.Dispose\(\)](#)

Namespace: [cfEngine.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IRuntimeSavable : IDisposable
```

Methods

Initialize(IUserData)

Declaration

```
void Initialize(IUserData userData)
```

Parameters

TYPE	NAME
IUserData	userData

SetSaveData(Dictionary<string, object>)

Declaration

```
void SetSaveData(Dictionary<string, object> dataMap)
```

Parameters

TYPE

NAME

Dictionary<string, object> dataMap

© Bus Fighter. All rights reserved.

Interface IUserData

Namespace: [cfEngine.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IUserData
```

Methods

GetContext<T>(string)

Declaration

```
T GetContext<T>(string contextKey)
```

Parameters

TYPE	NAME
------	------

string	contextKey
--------	------------

Returns

TYPE

T

Type Parameters

NAME

T

TryGetContext<T>(string, out T)

Declaration

```
bool TryGetContext<T>(string contextKey, out T context)
```

Parameters

TYPE	NAME
------	------

string	contextKey
--------	------------

T	context
---	---------

Returns

TYPE

bool

Type Parameters

NAME

T

© Bus Fighter. All rights reserved.

Class ServiceName

Inheritance

[object](#) → [ServiceName](#)

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public static class ServiceName
```

Fields

Auth

Declaration

```
public const string Auth = "Auth"
```

Field Value

TYPE

[string](#)

Info

Declaration

```
public const string Info = "Info"
```

Field Value

TYPE

string

Inventory

Declaration

```
public const string Inventory = "Inventory"
```

Field Value

TYPE

string

Pool

Declaration

```
public const string Pool = "Pool"
```

Field Value

TYPE

string

UserDataManager

Declaration

```
public const string UserDataManager = "UserDataManager"
```

Field Value

TYPE

string

© Bus Fighter. All rights reserved.

Class UserDataKey

Inheritance

`object` → `UserDataKey`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfEngine.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class UserDataKey
```

Fields

Inventory

Declaration

```
public const string Inventory = "Inventory"
```

Field Value

TYPE

`string`

SaveVersion

Declaration

```
public const string SaveVersion = "SaveVersion"
```

Field Value

TYPE

string

Statistic

Declaration

```
public const string Statistic = "Statistic"
```

Field Value

TYPE

string

© Bus Fighter. All rights reserved.

UserDataManager

Inheritance

`object` → `UserDataManager`

Implements

`IService`

`IDisposable`

Inherited Members

`object.Equals(object)`

`object.Equals(object, object)`

`object.GetHashCode()`

`object.GetType()`

`object.MemberwiseClone()`

`object.ReferenceEquals(object, object)`

`object.ToString()`

Namespace: `cfEngine.Core`

Assembly: `CatSweeper.dll`

Syntax

```
public class UserDataManager : IService, IDisposable
```

Constructors

UserDataManager(IStorage, ISerializer)

Declaration

```
public UserDataManager(IStorage storage, ISerializer serializer)
```

Parameters

TYPE	NAME
IStorage	storage
ISerializer	serializer

Methods

DeleteSave()

Declaration

```
public void DeleteSave()
```

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

InitializeSavables()

Declaration

```
public void InitializeSavables()
```

LoadDataMap(CancellationToken)

Declaration

```
public Task<IUserData> LoadDataMap(CancellationToken token = default)
```

Parameters

TYPE**NAME**

CancellationToken	token
-------------------	-------

Returns

TYPE

Task<IUserData>

Register(IRuntimeSavable)

Declaration

```
public void Register(IRuntimeSavable savable)
```

Parameters

TYPE**NAME**

IRuntimeSavable	savable
-----------------	---------

SaveAsync(IReadOnlyDictionary<string, object>, CancellationToken)

Declaration

```
public Task SaveAsync(IReadOnlyDictionary<string, object> dataMap, CancellationToken token = d
```

Parameters

TYPE**NAME**

IReadOnlyDictionary<string, object>	dataMap
-------------------------------------	---------

CancellationToken	token
-------------------	-------

Returns

TYPE

Task

SaveAsync(CancellationToken)

Declaration

```
public Task SaveAsync(CancellationToken token = default)
```

Parameters

TYPE	NAME
CancellationToken	token

Returns

TYPE
Task

TriggerSave()

Declaration

```
public void TriggerSave()
```

Implements

IService
IDisposable

© Bus Fighter. All rights reserved.

Struct UserDataManager.DeleteSaveCommand

Implements

[ICommand](#)

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfEngine.Core](#)

Assembly: CatSweeper.dll

Syntax

```
[CommandService.RegisterOnInitialized("Register")]
public struct UserDataManager.DeleteSaveCommand : ICommand
```

Methods

Execute(IReadOnlyDictionary<string, string>)

Declaration

```
public void Execute(IReadOnlyDictionary<string, string> args)
```

Parameters

TYPE	NAME
IReadOnlyDictionary<string, string>	args

Register()

Declaration

```
public static void Register()
```

Implements

ICommand

© Bus Fighter. All rights reserved.

Class UserDataManager.JsonContextMap

Inheritance

[object](#) → [UserDataManager.JsonContextMap](#)

Implements

[IUserData](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class UserDataManager.JsonContextMap : IUserData
```

Constructors

[JsonContextMap\(IReadOnlyDictionary<string, JsonObject>\)](#)

Declaration

```
public JsonContextMap(IReadOnlyDictionary<string, JsonObject> dataMap)
```

Parameters

TYPE	NAME
IReadOnlyDictionary<string, JsonObject>	dataMap

Fields

Empty

Declaration

```
public static UserDataManager.JsonContextMap Empty
```

Field Value

TYPE

UserDataManager.JsonContextMap

Methods

GetContext<T>(string)

Declaration

```
public T GetContext<T>(string contextKey)
```

Parameters

TYPE NAME

string contextKey

Returns

TYPE

T

Type Parameters

NAME

T

TryGetContext<T>(string, out T)

Declaration

```
public bool TryGetContext<T>(string contextKey, out T context)
```

Parameters

TYPE	NAME
------	------

string	contextKey
--------	------------

T	context
---	---------

Returns

TYPE

bool

Type Parameters

NAME

T

Implements

IUserData

© Bus Fighter. All rights reserved.

Namespace cfEngine.DataStructure

Classes

[MemoryDictionary<TMemoryKey, TValue>](#)

[ReadOnlyMemoryComparer<T>](#)

[WeakReferenceListPool<T>](#)

[WeakReferenceList<T>](#)

© Bus Fighter. All rights reserved.

Class MemoryDictionary<TMemoryKey, TValue>

Inheritance

object → Dictionary<ReadOnlyMemory<TMemoryKey>, TValue> → MemoryDictionary<TMemoryKey, TValue>

Implements

IDictionary<ReadOnlyMemory<TMemoryKey>, TValue>
ICollection<KeyValuePair<ReadOnlyMemory<TMemoryKey>, TValue>>
IReadOnlyDictionary<ReadOnlyMemory<TMemoryKey>, TValue>
IReadOnlyCollection<KeyValuePair<ReadOnlyMemory<TMemoryKey>, TValue>>
IEnumerable<KeyValuePair<ReadOnlyMemory<TMemoryKey>, TValue>>
IDictionary
ICollection
IEnumerable
IDeserializationCallback
ISerializable

Inherited Members

Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.Add(ReadOnlyMemory<TMemoryKey>, TValue)
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.Clear()
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.ContainsKey(ReadOnlyMemory<TMemoryKey>)
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.ContainsValue(TValue)
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.EnsureCapacity(int)
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.GetEnumerator()
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.OnDeserialization(object)
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.Remove(ReadOnlyMemory<TMemoryKey>)
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.Remove(ReadOnlyMemory<TMemoryKey>, out TValue)
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.TrimExcess()
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.TrimExcess(int)
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.TryAdd(ReadOnlyMemory<TMemoryKey>, TValue)
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.TryGetValue(ReadOnlyMemory<TMemoryKey>, out TValue)
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.Comparer
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.Count
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.this[ReadOnlyMemory<TMemoryKey>]
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.Keys
Dictionary<ReadOnlyMemory<TMemoryKey>, TValue>.Values
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()

`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfEngine.DataStructure](#)
Assembly: CatSweeper.dll

Syntax

```
public class MemoryDictionary<TMemoryKey, TValue> : Dictionary<ReadOnlyMemory<TMemoryKey>, TVa
```

Type Parameters

NAME

TMemoryKey

TValue

Constructors

MemoryDictionary()

Declaration

```
public MemoryDictionary()
```

Methods

Create()

Declaration

```
public static MemoryDictionary<TMemoryKey, TValue> Create()
```

Returns

TYPE

[MemoryDictionary](#)<TMemoryKey, TValue>

Implements

`IDictionary< TKey, TValue >`

`ICollection< T >`

`IReadOnlyDictionary< TKey, TValue >`

`IReadOnlyCollection< T >`

`IEnumerable< T >`

`IDictionary`

`ICollection`

`IEnumerable`

`IDeserializationCallback`

`ISerializable`

© Bus Fighter. All rights reserved.

Class ReadOnlyMemoryComparer<T>

Inheritance

[object](#) → [ReadOnlyMemoryComparer<T>](#)

Implements

[IEqualityComparer<ReadOnlyMemory<T>>](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.DataStructure](#)

Assembly: CatSweeper.dll

Syntax

```
public class ReadOnlyMemoryComparer<T> : IEqualityComparer<ReadOnlyMemory<T>>
```

Type Parameters

NAME

T

Properties

Instance

Declaration

```
public static ReadOnlyMemoryComparer<T> Instance { get; }
```

TYPE

`ReadOnlyMemoryComparer<T>`

Methods

Equals(ReadOnlyMemory<T>, ReadOnlyMemory<T>)

Determines whether the specified objects are equal.

Declaration

```
public bool Equals(ReadOnlyMemory<T> x, ReadOnlyMemory<T> y)
```

Parameters

TYPE	NAME	DESCRIPTION
<code>ReadOnlyMemory<T></code>	<code>x</code>	The first object of type <code>T</code> to compare.
<code>ReadOnlyMemory<T></code>	<code>y</code>	The second object of type <code>T</code> to compare.

Returns

TYPE	DESCRIPTION
<code>bool</code>	true if the specified objects are equal; otherwise, <code>false</code> .

GetHashCode(ReadOnlyMemory<T>)

Returns a hash code for the specified object.

Declaration

```
public int GetHashCode(ReadOnlyMemory<T> obj)
```

Parameters

TYPE	NAME	DESCRIPTION
<code>ReadOnlyMemory<T></code>	<code>obj</code>	The <code>object</code> for which a hash code is to be returned.

Returns

TYPE DESCRIPTION

int	A hash code for the specified object.
-----	---------------------------------------

Exceptions

TYPE

CONDITION

ArgumentNullException	The type of obj is a reference type and obj is null .
-----------------------	--

Implements

IEqualityComparer<T>

© Bus Fighter. All rights reserved.

Class WeakReferenceListPool<T>

Inheritance

object → [ObjectPool<WeakReferenceList<T>>](#) → [WeakReferenceListPool<T>](#)

Implements

[IObjectPool](#)

[IDisposable](#)

Inherited Members

[ObjectPool<WeakReferenceList<T>>.Queue](#)

[ObjectPool<WeakReferenceList<T>>.Get\(\)](#)

[ObjectPool<WeakReferenceList<T>>.Get\(out WeakReferenceList<T>\)](#)

[ObjectPool<WeakReferenceList<T>>.Release\(WeakReferenceList<T>\)](#)

[ObjectPool<WeakReferenceList<T>>.Dispose\(\)](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.DataStructure](#)

Assembly: CatSweeper.dll

Syntax

```
public class WeakReferenceListPool<T> : ObjectPool<WeakReferenceList<T>>, IObjectPool, IDisposable
```

Type Parameters

NAME

T

Constructors

Declaration

```
public WeakReferenceListPool()
```

Implements

IObjectPool

IDisposable

© Bus Fighter. All rights reserved.

Class WeakReferenceList<T>

Inheritance

object → [List<WeakReference<T>>](#) → [WeakReferenceList<T>](#)

Implements

[IList<WeakReference<T>>](#)
[ICollection<WeakReference<T>>](#)
 [IReadOnlyList<WeakReference<T>>](#)
 [IReadOnlyCollection<WeakReference<T>>](#)
[IEnumerable<WeakReference<T>>](#)
[IList](#)
[ICollection](#)
[IEnumerable](#)

Inherited Members

[List<WeakReference<T>>.Add\(WeakReference<T>\)](#)
[List<WeakReference<T>>.AddRange\(IEnumerable<WeakReference<T>>\)](#)
[List<WeakReference<T>>.AsReadOnly\(\)](#)
[List<WeakReference<T>>.BinarySearch\(int, int, WeakReference<T>, IComparer<WeakReference<T>>\)](#)
[List<WeakReference<T>>.BinarySearch\(WeakReference<T>\)](#)
[List<WeakReference<T>>.BinarySearch\(WeakReference<T>, IComparer<WeakReference<T>>\)](#)
[List<WeakReference<T>>.Clear\(\)](#)
[List<WeakReference<T>>.Contains\(WeakReference<T>\)](#)
[List<WeakReference<T>>.ConvertAll<TOoutput>\(Converter<WeakReference<T>, TOoutput>\)](#)
[List<WeakReference<T>>.CopyTo\(int, WeakReference<T>\[\], int, int\)](#)
[List<WeakReference<T>>.CopyTo\(WeakReference<T>\[\]\)](#)
[List<WeakReference<T>>.CopyTo\(WeakReference<T>\[\], int\)](#)
[List<WeakReference<T>>.EnsureCapacity\(int\)](#)
[List<WeakReference<T>>.Exists\(Predicate<WeakReference<T>>\)](#)
[List<WeakReference<T>>.Find\(Predicate<WeakReference<T>>\)](#)
[List<WeakReference<T>>.FindAll\(Predicate<WeakReference<T>>\)](#)
[List<WeakReference<T>>.FindIndex\(int, int, Predicate<WeakReference<T>>\)](#)
[List<WeakReference<T>>.FindIndex\(int, Predicate<WeakReference<T>>\)](#)
[List<WeakReference<T>>.FindIndex\(Predicate<WeakReference<T>>\)](#)
[List<WeakReference<T>>.FindLast\(Predicate<WeakReference<T>>\)](#)
[List<WeakReference<T>>.FindLastIndex\(int, int, Predicate<WeakReference<T>>\)](#)
[List<WeakReference<T>>.FindLastIndex\(int, Predicate<WeakReference<T>>\)](#)
[List<WeakReference<T>>.FindLastIndex\(Predicate<WeakReference<T>>\)](#)
[List<WeakReference<T>>.ForEach\(Action<WeakReference<T>>\)](#)
[List<WeakReference<T>>.GetEnumerator\(\)](#)

List<WeakReference<T>>.GetRange(int, int)
List<WeakReference<T>>.IndexOf(WeakReference<T>)
List<WeakReference<T>>.IndexOf(WeakReference<T>, int)
List<WeakReference<T>>.IndexOf(WeakReference<T>, int, int)
List<WeakReference<T>>.Insert(int, WeakReference<T>)
List<WeakReference<T>>.InsertRange(int, IEnumerable<WeakReference<T>>)
List<WeakReference<T>>.LastIndexOf(WeakReference<T>)
List<WeakReference<T>>.LastIndexOf(WeakReference<T>, int)
List<WeakReference<T>>.LastIndexOf(WeakReference<T>, int, int)
List<WeakReference<T>>.Remove(WeakReference<T>)
List<WeakReference<T>>.RemoveAll(Predicate<WeakReference<T>>)
List<WeakReference<T>>.RemoveAt(int)
List<WeakReference<T>>.RemoveRange(int, int)
List<WeakReference<T>>.Reverse()
List<WeakReference<T>>.Reverse(int, int)
List<WeakReference<T>>.Slice(int, int)
List<WeakReference<T>>.Sort()
List<WeakReference<T>>.Sort(IComparer<WeakReference<T>>)
List<WeakReference<T>>.Sort(Comparison<WeakReference<T>>)
List<WeakReference<T>>.Sort(int, int, IComparer<WeakReference<T>>)
List<WeakReference<T>>.ToArray()
List<WeakReference<T>>.TrimExcess()
List<WeakReference<T>>.TrueForAll(Predicate<WeakReference<T>>)
List<WeakReference<T>>.Capacity
List<WeakReference<T>>.Count
List<WeakReference<T>>.this[int]
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfEngine.DataStructure](#)

Assembly: CatSweeper.dll

Syntax

```
public class WeakReferenceList<T> : List<WeakReference<T>>, IList<WeakReference<T>>, ICollecti
```

Type Parameters

NAME

T

Methods

Add(T)

Declaration

```
public void Add(T item)
```

Parameters

TYPE	NAME
------	------

T	item
---	------

Create()

Declaration

```
public static WeakReferenceList<T> Create()
```

Returns

TYPE

WeakReferenceList<T>

Remove(T)

Declaration

```
public void Remove(T item)
```

Parameters

TYPE	NAME
------	------

T	item
---	------

Implements

[IList<T>](#)

[ICollection<T>](#)

[IReadOnlyList<T>](#)

IReadOnlyCollection<T>

IEnumerable<T>

IList

ICollection

IEnumerable

Extension Methods

ListExtension.EnsureCapacity<T>(List<T>, int)

© Bus Fighter. All rights reserved.

Namespace cfEngine.DataStructure.test

Classes

[TestWeakReferenceList](#)

[TestWeakReferenceList.TestObject](#)

© Bus Fighter. All rights reserved.

Class TestWeakReferenceList

Inheritance

`object` → `TestWeakReferenceList`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfEngine.DataStructure.test`

Assembly: `CatSweeper.dll`

Syntax

```
[TestFixture]
public class TestWeakReferenceList
```

Methods

Test_Add()

Declaration

```
[Test]
public void Test_Add()
```

Test_Create()

Declaration

```
[Test]
```

```
public void Test_Create()
```

© Bus Fighter. All rights reserved.

Class TestWeakReferenceList.TestObject

Inheritance

[object](#) → TestWeakReferenceList.TestObject

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.DataStructure.test](#)

Assembly: CatSweeper.dll

Syntax

```
public class TestWeakReferenceList.TestObject
```

© Bus Fighter. All rights reserved.

Namespace cfEngine.Extension

Classes

[EnumExtension](#)

[ListExtension](#)

[TaskExtension](#)

© Bus Fighter. All rights reserved.

Class EnumExtension

Inheritance

`object` → `EnumExtension`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfEngine.Extension](#)

Assembly: CatSweeper.dll

Syntax

```
public static class EnumExtension
```

Methods

hasFlag(Enum, Enum)

Declaration

```
public static bool hasFlag(this Enum target, Enum flag)
```

Parameters

TYPE	NAME
<code>Enum</code>	<code>target</code>
<code>Enum</code>	<code>flag</code>

Returns

© Bus Fighter. All rights reserved.

Class ListExtension

Inheritance

`object` → `ListExtension`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfEngine.Extension](#)

Assembly: CatSweeper.dll

Syntax

```
public static class ListExtension
```

Methods

EnsureCapacity<T>(List<T>, int)

Declaration

```
public static void EnsureCapacity<T>(this List<T> list, int capacity)
```

Parameters

TYPE	NAME
<code>List<T></code>	<code>list</code>
<code>int</code>	<code>capacity</code>

Type Parameters

NAME

T

© Bus Fighter. All rights reserved.

Class TaskExtension

Inheritance

`object` → `TaskExtension`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfEngine.Extension](#)

Assembly: CatSweeper.dll

Syntax

```
public static class TaskExtension
```

Methods

ContinueWithSynchronized(Task, Action<Task>, CancellationToken)

Declaration

```
public static Task ContinueWithSynchronized(this Task t, Action<Task> action, CancellationTokenToke
```

Parameters

TYPE	NAME
<code>Task</code>	<code>t</code>
<code>Action<Task></code>	<code>action</code>
<code>CancellationToken</code>	<code>token</code>

Returns

TYPE

Task

ContinueWithSynchronized<T>(Task<T>, Action<Task<T>>, CancellationToken)

Declaration

```
public static Task ContinueWithSynchronized<T>(this Task<T> t, Action<Task<T>> action, Cancell
```

Parameters

TYPE	NAME
Task<T>	t
Action<Task<T>>	action
CancellationToken	token

Returns

TYPE

Task

Type Parameters

NAME
T

ContinueWithSynchronized<T>(Task<T>, Func<Task<T>, T>, CancellationToken)

Declaration

```
public static Task<T> ContinueWithSynchronized<T>(this Task<T> t, Func<Task<T>, T> action, Can
```

Parameters

TYPE	NAME
Task<T>	t
Func<Task<T>, T>	action
CancellationToken	token

Returns

TYPE
Task<T>

Type Parameters

NAME
T

DisposeIfCompleted(Task)

Declaration

```
public static void DisposeIfCompleted(this Task t)
```

Parameters

TYPE	NAME
Task	t

LogAggregateException(AggregateException)

Declaration

```
public static void LogAggregateException(this AggregateException ex)
```

Parameters

TYPE	NAME
AggregateException	ex

LogIfFaulted(Task)

Declaration

```
public static Task LogIfFaulted(this Task t)
```

Parameters

TYPE	NAME
------	------

Task	t
------	---

Returns

TYPE

Task

LogIfFaulted<T>(Task<T>)

Declaration

```
public static Task<T> LogIfFaulted<T>(this Task<T> t)
```

Parameters

TYPE	NAME
------	------

Task<T>	t
---------	---

Returns

TYPE

Task<T>

Type Parameters

NAME

T

Namespace cfEngine.IO

Classes

[LocalFileStorage](#)

Interfaces

[IStorage](#)

© Bus Fighter. All rights reserved.

Interface IStorage

Inherited Members

[IDisposable.Dispose\(\)](#)

Namespace: [cfEngine.IO](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IStorage : IDisposable
```

Methods

[CopyFile\(string, string, bool\)](#)

Declaration

```
void CopyFile(string relativeFrom, string relativeTo, bool overwrite = false)
```

Parameters

TYPE	NAME
string	relativeFrom
string	relativeTo
bool	overwrite

[CreateStream\(string, bool\)](#)

Declaration

```
Stream CreateStream(string relativePath, bool useAsync)
```

Parameters

TYPE	NAME
string	relativePath
bool	useAsync

Returns

TYPE
Stream

DeleteFile(string)

Declaration

```
void DeleteFile(string relativePath)
```

Parameters

TYPE	NAME
string	relativePath

GetFiles(string)

Declaration

```
string[] GetFiles(string searchPattern)
```

Parameters

TYPE	NAME
string	searchPattern

Returns

TYPE
string[]

GetFiles(string, string)

Declaration

```
string[] GetFiles(string directory, string searchPattern)
```

Parameters

TYPE	NAME
------	------

string	directory
--------	-----------

string	searchPattern
--------	---------------

Returns

TYPE

string[]

IsFileExist(string)

Declaration

```
bool IsFileExist(string relativePath)
```

Parameters

TYPE	NAME
------	------

string	relativePath
--------	--------------

Returns

TYPE

bool

IsStorageExist()

Declaration

```
bool IsStorageExist()
```

Returns

TYPE

bool

LoadBytes(string)

Declaration

```
byte[] LoadBytes(string relativePath)
```

Parameters

TYPE NAME

string relativePath

Returns

TYPE

byte[]

LoadBytesAsync(string, CancellationToken)

Declaration

```
Task<byte[]> LoadBytesAsync(string relativePath, CancellationToken token = default)
```

Parameters

TYPE NAME

string relativePath

CancellationToken token

Returns

TYPE

Task<byte[]>

Save(string, byte[])

Declaration

```
void Save(string relativeFilePath, byte[] data)
```

Parameters

TYPE	NAME
------	------

string	relativeFilePath
--------	------------------

byte[]	data
--------	------

Save(string, string)

Declaration

```
void Save(string relativeFilePath, string data)
```

Parameters

TYPE	NAME
------	------

string	relativeFilePath
--------	------------------

string	data
--------	------

SaveAsync(string, byte[], CancellationToken)

Declaration

```
Task SaveAsync(string relativeFilePath, byte[] data, CancellationToken token = default)
```

Parameters

TYPE	NAME
------	------

string	relativeFilePath
--------	------------------

byte[]	data
--------	------

CancellationToken	token
-------------------	-------

Returns

SaveAsync(string, string, CancellationToken)

Declaration

```
Task SaveAsync(string relativeFilePath, string data, CancellationToken token = default)
```

Parameters

TYPE	NAME
string	relativeFilePath
string	data
CancellationToken	token

Returns

© Bus Fighter. All rights reserved.

Class LocalFileStorage

Inheritance

[object](#) → LocalFileStorage

Implements

[IStorage](#)

[IDisposable](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.IO](#)

Assembly: CatSweeper.dll

Syntax

```
public class LocalFileStorage : IStorage, IDisposable
```

Constructors

LocalFileStorage(string)

Declaration

```
public LocalFileStorage(string storagePath)
```

Parameters

TYPE NAME

string storagePath

Methods

CopyFile(string, string, bool)

Declaration

```
public void CopyFile(string relativeFrom, string relativeTo, bool overwrite = false)
```

Parameters

TYPE NAME

string relativeFrom

string relativeTo

bool overwrite

CreateStream(string, bool)

Declaration

```
public Stream CreateStream(string relativePath, bool useAsync)
```

Parameters

TYPE NAME

string relativePath

bool useAsync

Returns

TYPE

Stream

DeleteFile(string)

Declaration

```
public void DeleteFile(string relativePath)
```

Parameters

TYPE	NAME
------	------

string	relativePath
--------	--------------

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

GetFiles(string)

Declaration

```
public string[] GetFiles(string searchPattern)
```

Parameters

TYPE	NAME
------	------

string	searchPattern
--------	---------------

Returns

TYPE

string[]

GetFiles(string, string)

Declaration

```
public string[] GetFiles(string directory, string searchPattern)
```

Parameters

TYPE	NAME
string	directory
string	searchPattern

Returns

TYPE
string[]

IsFileExist(string)

Declaration

```
public bool IsFileExist(string relativePath)
```

Parameters

TYPE	NAME
string	relativePath

Returns

TYPE
bool

IsStorageExist()

Declaration

```
public bool IsStorageExist()
```

Returns

TYPE
bool

LoadBytes(string)

Declaration

```
public byte[] LoadBytes(string relativePath)
```

Parameters

TYPE	NAME
------	------

string	relativePath
--------	--------------

Returns

TYPE

byte[]

LoadBytesAsync(string, CancellationToken)

Declaration

```
public Task<byte[]> LoadBytesAsync(string relativePath, CancellationToken cancellationToken =
```

Parameters

TYPE	NAME
------	------

string	relativePath
--------	--------------

CancellationToken	cancellationToken
-------------------	-------------------

Returns

TYPE

Task<byte[]>

Save(string, byte[])

Declaration

```
public void Save(string relativeFilePath, byte[] data)
```

Parameters

TYPE	NAME
string	relativeFilePath
byte[]	data

Save(string, string)

Declaration

```
public void Save(string relativeFilePath, string data)
```

Parameters

TYPE	NAME
string	relativeFilePath
string	data

SaveAsync(string, byte[], CancellationToken)

Declaration

```
public Task SaveAsync(string relativeFilePath, byte[] data, CancellationToken token = default)
```

Parameters

TYPE	NAME
string	relativeFilePath
byte[]	data
CancellationToken	token

Returns

TYPE
Task

SaveAsync(string, string, CancellationToken)

Declaration

```
public Task SaveAsync(string relativeFilePath, string data, CancellationToken token = default)
```

Parameters

TYPE	NAME
string	relativeFilePath
string	data
CancellationToken	token

Returns

TYPE
Task

Implements

IStorage

IDisposable

© Bus Fighter. All rights reserved.

Namespace cfEngine.Info

Classes

[ConfigInfoManager<TKey, TInfo>](#)

[ExcelJsonLoader<TInfo>](#)

[InfoLayer](#)

[InfoManager](#)

[InfoUtil](#)

[JsonElementDecoder](#)

[SerializationLoader<TInfo>](#)

Interfaces

[IInfoManager](#)

[IValueLoader<TInfo>](#)

© Bus Fighter. All rights reserved.

Class ConfigInfoManager<TKey, TInfo>

Inheritance

object → [InfoManager](#) → [ConfigInfoManager<TKey, TInfo>](#) → [GameSettingInfoManager](#) → [StageInfoManager](#) → [StageMapInfoManager](#) → [InventoryInfoManager](#)

Implements

[IInfoManager](#)

[IDisposable](#)

Inherited Members

[InfoManager.OnLoadCompleted\(\)](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class ConfigInfoManager<TKey, TInfo> : InfoManager, IInfoManager, IDisposable
```

Type Parameters

NAME

TKey

TInfo

Constructors

[ConfigInfoManager\(IValueLoader<TInfo>\)](#)

Declaration

```
protected ConfigInfoManager(IValueLoader<TInfo> loader)
```

Parameters

TYPE	NAME
IValueLoader<TInfo>	loader

Properties

infoType

Declaration

```
public override Type infoType { get; }
```

Property Value

TYPE
Type

Overrides

[InfoManager.infoType](#)

keyFn

Declaration

```
protected abstract Func<TInfo, TKey> keyFn { get; }
```

Property Value

TYPE
Func<TInfo, TKey>

valueMap

Declaration

```
public IReadOnlyDictionary<TKey, TInfo> valueMap { get; }
```

Property Value

TYPE

IReadOnlyDictionary<TKey, TInfo>

Methods

AddValue(TInfo)

Declaration

```
public void AddValue(TInfo value)
```

Parameters

TYPE NAME

TInfo value

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public override void Dispose()
```

Overrides

[InfoManager.Dispose\(\)](#)

GetAllValue()

Declaration

```
public override IEnumerable<object> GetAllValue()
```

Returns

TYPE

IEnumerable<object>

Overrides

[InfoManager.GetAllValue\(\)](#)

LoadInfo()

Declaration

```
public override void LoadInfo()
```

Overrides

[InfoManager.LoadInfo\(\)](#)

LoadInfoAsync(CancellationToken)

Declaration

```
public override Task LoadInfoAsync(CancellationToken cancellationToken)
```

Parameters

TYPE	NAME
CancellationToken	cancellationToken

Returns

TYPE

Task

Overrides

[InfoManager.LoadInfoAsync\(CancellationToken\)](#)

TryGetValue(TKey, out TInfo)

Declaration

```
public bool TryGetValue(TKey key, out TInfo value)
```

Parameters

TYPE	NAME
------	------

TKey	key
------	-----

TInfo	value
-------	-------

Returns

TYPE

bool

Implements

[IInfoManager](#)

[IDisposable](#)

© Bus Fighter. All rights reserved.

Class ExcelJsonLoader<TInfo>

Inheritance

[object](#) → [ExcelJsonLoader<TInfo>](#)

Implements

[IValueLoader<TInfo>](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public class ExcelJsonLoader<TInfo> : IValueLoader<TInfo>
```

Type Parameters

NAME

TInfo

Constructors

ExcelJsonLoader(IStorage, DataObjectEncoder)

Declaration

```
public ExcelJsonLoader(IStorage storage, DataObjectEncoder encoder)
```

Parameters

TYPE	NAME
IStorage	storage
DataObjectEncoder	encoder

Methods

Load(out List<TInfo>)

Declaration

```
public ObjectPool<List<TInfo>>.Handle Load(out List<TInfo> values)
```

Parameters

TYPE	NAME
List<TInfo>	values

Returns

TYPE
ObjectPool<List<TInfo>>.Handle

LoadAsync(CancellationToken)

Declaration

```
public Task<List<TInfo>> LoadAsync(CancellationToken cancellationToken)
```

Parameters

TYPE	NAME
CancellationToken	cancellationToken

Returns

Implements

IValueLoader<TInfo>

© Bus Fighter. All rights reserved.

Interface IInfoManager

Inherited Members

[IDisposable.Dispose\(\)](#)

Namespace: [cfEngine.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IInfoManager : IDisposable
```

Properties

infoType

Declaration

```
Type infoType { get; }
```

Property Value

TYPE

[Type](#)

Methods

GetAllValue()

Declaration

```
IEnumerable<object> GetAllValue()
```

Returns

TYPE

```
IEnumerable<object>
```

LoadInfo()

Declaration

```
void LoadInfo()
```

LoadInfoAsync(CancellationToken)

Declaration

```
Task LoadInfoAsync(CancellationToken cancellationToken)
```

Parameters

TYPE	NAME
CancellationToken	cancellationToken

Returns

TYPE

```
Task
```

© Bus Fighter. All rights reserved.

Interface IValueLoader<TInfo>

Namespace: [cfEngine.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IValueLoader<TInfo>
```

Type Parameters

NAME

TInfo

Methods

Load([out List<TInfo>](#))

Declaration

```
ObjectPool<List<TInfo>>.Handle Load(out List<TInfo> values)
```

Parameters

TYPE	NAME
------	------

List<TInfo>	values
-------------	--------

Returns

TYPE

ObjectPool<List<TInfo>>.Handle

LoadAsync(CancellationToken)

Declaration

```
Task<List<TInfo>> LoadAsync(CancellationToken cancellationToken)
```

Parameters

TYPE	NAME
CancellationToken	cancellationToken

Returns

TYPE
Task<List<TInfo>>

© Bus Fighter. All rights reserved.

Class InfoLayer

Inheritance

`object` → `InfoLayer`

Implements

`IService`

`IDisposable`

Inherited Members

`object.Equals(object)`

`object.Equals(object, object)`

`object.GetHashCode()`

`object.GetType()`

`object.MemberwiseClone()`

`object.ReferenceEquals(object, object)`

`object.ToString()`

Namespace: `cfEngine.Info`

Assembly: `CatSweeper.dll`

Syntax

```
public class InfoLayer : IService, IDisposable
```

Properties

InfoMap

Declaration

```
public IReadOnlyDictionary<Type, IIInfoManager> InfoMap { get; }
```

Property Value

TYPE

IReadOnlyDictionary<Type, IInfoManager>

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

Get<TInfo>()

Declaration

```
public TInfo Get<TInfo>() where TInfo : InfoManager
```

Returns

TYPE

TInfo

Type Parameters

NAME

TInfo

LoadInfoAsync()

Declaration

```
public IEnumerable<Task> LoadInfoAsync()
```

Returns

TYPE

IEnumerable<Task>

RegisterInfo(IInfoManager)

Declaration

```
public void RegisterInfo(IInfoManager infoManager)
```

Parameters

TYPE	NAME
------	------

IInfoManager	infoManager
--------------	-------------

TryGetInfo<TInfo>(out TInfo)

Declaration

```
public bool TryGetInfo<TInfo>(out TInfo infoManager) where TInfo : InfoManager
```

Parameters

TYPE	NAME
------	------

TInfo	infoManager
-------	-------------

Returns

TYPE

bool

Type Parameters

NAME

TInfo

Implements

IService

© Bus Fighter. All rights reserved.

Class InfoManager

Inheritance

object → InfoManager → ConfigInfoManager<TKey, TInfo>

Implements

IInfoManager

IDisposable

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: [cfEngine.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class InfoManager : IInfoManager, IDisposable
```

Properties

infoType

Declaration

```
public abstract Type infoType { get; }
```

Property Value

TYPE

Type

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public virtual void Dispose()
```

GetAllValue()

Declaration

```
public abstract IEnumerable<object> GetAllValue()
```

Returns

TYPE

IEnumerable<object>

LoadInfo()

Declaration

```
public abstract void LoadInfo()
```

LoadInfoAsync(CancellationToken)

Declaration

```
public abstract Task LoadInfoAsync(CancellationToken cancellationToken)
```

Parameters

TYPE	NAME
CancellationToken	cancellationToken

Returns

TYPE
Task

OnLoadCompleted()

Declaration

```
protected virtual void OnLoadCompleted()
```

Implements

IInfoManager

IDisposable

© Bus Fighter. All rights reserved.

Class InfoUtil

Inheritance

`object` → `InfoUtil`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfEngine.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public static class InfoUtil
```

Methods

`ParseCommand(ReadOnlySpan<char>, out IReadOnlyList<string>, char, char, char)`

Declaration

```
public static string ParseCommand(ReadOnlySpan<char> command, out IReadOnlyList<string> args,
```

Parameters

TYPE	NAME
<code>ReadOnlySpan<char></code>	<code>command</code>
<code>IReadOnlyList<string></code>	<code>args</code>

TYPE**NAME**

char	separator
char	open
char	close

Returns**TYPE**

string

ParseTypeCommand(ReadOnlySpan<char>, out IReadOnlyList<string>, char, char, char)

Declaration

```
public static Type ParseTypeCommand(ReadOnlySpan<char> command, out IReadOnlyList<string> args
```

Parameters**TYPE****NAME**

ReadOnlySpan<char>	command
IReadOnlyList<string>	args
char	separator
char	open
char	close

Returns**TYPE**

Type

Class JsonElementDecoder

Inheritance

[object](#) → JsonElementDecoder

Implements

[DataObject.IValueDecoder](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public class JsonElementDecoder : DataObject.IValueDecoder
```

Properties

valueType

Declaration

```
public Type valueType { get; }
```

Property Value

TYPE

Type

Methods

TryDecode(object, Type, out object)

Declaration

```
public bool TryDecode(object raw, Type decodedType, out object decoded)
```

Parameters

TYPE	NAME
object	raw
Type	decodedType
object	decoded

Returns

TYPE
bool

Implements

DataObject.IValueDecoder

© Bus Fighter. All rights reserved.

Class SerializationLoader<TInfo>

Inheritance

`object` → `SerializationLoader<TInfo>`

Implements

`IValueLoader<TInfo>`

Inherited Members

`object.Equals(object)`

`object.Equals(object, object)`

`object.GetHashCode()`

`object.GetType()`

`object.MemberwiseClone()`

`object.ReferenceEquals(object, object)`

`object.ToString()`

Namespace: `cfEngine.Info`

Assembly: `CatSweeper.dll`

Syntax

```
public class SerializationLoader<TInfo> : IValueLoader<TInfo>
```

Type Parameters

NAME

`TInfo`

Constructors

`SerializationLoader(IStorage, ISerializer)`

Declaration

```
public SerializationLoader(IStorage storage, ISerializer serializer)
```

Parameters

TYPE	NAME
IStorage	storage
ISerializer	serializer

Methods

Load(out List<TInfo>)

Declaration

```
public ObjectPool<List<TInfo>>.Handle Load(out List<TInfo> values)
```

Parameters

TYPE	NAME
List<TInfo>	values

Returns

TYPE
ObjectPool<List<TInfo>>.Handle

LoadAsync(CancellationToken)

Declaration

```
public Task<List<TInfo>> LoadAsync(CancellationToken cancellationToken)
```

Parameters

TYPE	NAME
CancellationToken	cancellationToken

Returns

Implements

IValueLoader<TInfo>

© Bus Fighter. All rights reserved.

Namespace cfEngine.Logging

Classes

[Log](#)

Interfaces

[ILogger](#)

Enums

[LogLevel](#)

[LogType](#)

© Bus Fighter. All rights reserved.

Interface ILogger

Namespace: [cfEngine.Logging](#)

Assembly: CatSweeper.dll

Syntax

```
public interface ILogger
```

Methods

Asset(bool, object)

Declaration

```
void Asset(bool condition, object context = null)
```

Parameters

TYPE	NAME
bool	condition
object	context

LogDebug(string, object)

Declaration

```
void LogDebug(string message, object context = null)
```

Parameters

TYPE	NAME
------	------

string	message
--------	---------

object	context
--------	---------

.LogError(string, object)

Declaration

```
void LogError(string message, object context = null)
```

Parameters

TYPE	NAME
------	------

string	message
--------	---------

object	context
--------	---------

LogException(Exception, object)

Declaration

```
void LogException(Exception ex, object message = null)
```

Parameters

TYPE	NAME
------	------

Exception	ex
-----------	----

object	message
--------	---------

LogInfo(string, object)

Declaration

```
void LogInfo(string message, object context = null)
```

Parameters

TYPE	NAME
------	------

string	message
--------	---------

object	context
--------	---------

LogWarning(string, object)

Declaration

```
void LogWarning(string message, object context = null)
```

Parameters

TYPE	NAME
------	------

string	message
--------	---------

object	context
--------	---------

© Bus Fighter. All rights reserved.

Class Log

Inheritance

`object` → Log

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfEngine.Logging](#)

Assembly: CatSweeper.dll

Syntax

```
public static class Log
```

Methods

Asset(bool, object)

Declaration

```
public static void Asset(bool condition, object context = null)
```

Parameters

TYPE	NAME
<code>bool</code>	<code>condition</code>
<code>object</code>	<code>context</code>

LogDebug(string, object)

Declaration

```
public static void LogDebug(string message, object context = null)
```

Parameters

TYPE	NAME
string	message
object	context

.LogError(string, object)

Declaration

```
public static void LogError(string message, object context = null)
```

Parameters

TYPE	NAME
string	message
object	context

LogException(Exception, string)

Declaration

```
public static void LogException(Exception ex, string message = null)
```

Parameters

TYPE	NAME
Exception	ex
string	message

LogInfo(string, object)

Declaration

```
public static void LogInfo(string message, object context = null)
```

Parameters

TYPE	NAME
string	message
object	context

LogWarning(string, object)

Declaration

```
public static void LogWarning(string message, object context = null)
```

Parameters

TYPE	NAME
string	message
object	context

SetLogLevel(LogLevel)

Declaration

```
public static void SetLogLevel(LogLevel logLevel)
```

Parameters

TYPE	NAME
LogLevel	logLevel

SetLogger(ILLogger)

Declaration

```
public static void SetLogger(ILogger logger)
```

Parameters

TYPE	NAME
ILogger	logger

© Bus Fighter. All rights reserved.

Enum LogLevel

Namespace: [cfEngine.Logging](#)

Assembly: CatSweeper.dll

Syntax

```
public enum LogLevel
```

Fields

NAME
Debug
Error
Info
Verbose
Warn

Extension Methods

[EnumExtension.hasFlag\(Enum, Enum\)](#)

© Bus Fighter. All rights reserved.

Enum LogType

Namespace: [cfEngine.Logging](#)

Assembly: CatSweeper.dll

Syntax

```
[Flags]  
public enum LogType
```

Fields

NAME

Assert

Debug

Error

Exception

Info

Warning

Extension Methods

[EnumExtension.hasFlag\(Enum, Enum\)](#)

Namespace cfEngine.Pooling

Classes

[ListPool<T>](#)

[ObjectPool<T>](#)

[PoolManager](#)

[StringBuilderPool](#)

Structs

[ObjectPool<T>.Handle](#)

Interfaces

[IObjectPool](#)

© Bus Fighter. All rights reserved.

Interface IObjectPool

Inherited Members

[IDisposable.Dispose\(\)](#)

Namespace: [cfEngine.Pooling](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IObjectPool : IDisposable
```

© Bus Fighter. All rights reserved.

Class ListPool<T>

Inheritance

object → ObjectPool<List<T>> → ListPool<T>

Implements

IObjectPool

IDisposable

Inherited Members

ObjectPool<List<T>>.Queue

ObjectPool<List<T>>.Get()

ObjectPool<List<T>>.Get(out List<T>)

ObjectPool<List<T>>.Release(List<T>)

ObjectPool<List<T>>.Dispose()

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: [cfEngine.Pooling](#)

Assembly: CatSweeper.dll

Syntax

```
public class ListPool<T> : ObjectPool<List<T>>, IObjectPool, IDisposable
```

Type Parameters

NAME

T

Constructors

ListPool()

Declaration

```
public ListPool()
```

Properties

Default

Declaration

```
public static ListPool<T> Default { get; }
```

Property Value

TYPE

ListPool<T>

Implements

IObjectPool

IDisposable

© Bus Fighter. All rights reserved.

Class ObjectPool<T>

Inheritance

object → ObjectPool<T> → WeakReferenceListPool<T> → ListPool<T> → StringBuilderPool

Implements

IObjectPool

IDisposable

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: [cfEngine.Pooling](#)

Assembly: CatSweeper.dll

Syntax

```
public class ObjectPool<T> : IObjectPool, IDisposable where T : class
```

Type Parameters

NAME

T

Constructors

[ObjectPool\(Func<T>, Action<T>, Action<T>, int\)](#)

Declaration

```
public ObjectPool<Func<T> createMethod, Action<T> getAction, Action<T> releaseAction, int warm
```

Parameters

TYPE	NAME
Func<T>	createMethod
Action<T>	getAction
Action<T>	releaseAction
int	warmupSize

Fields

Queue

Declaration

```
protected readonly Queue<T> Queue
```

Field Value

TYPE
Queue<T>

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public virtual void Dispose()
```

Get()

Declaration

```
public virtual T Get()
```

Returns

TYPE

T

Get(out T)

Declaration

```
public virtual ObjectPool<T>.Handle Get(out T value)
```

Parameters

TYPE NAME

T value

Returns

TYPE

ObjectPool<T>.Handle

Release(T)

Declaration

```
public virtual void Release(T obj)
```

Parameters

TYPE NAME

T obj

Implements

IObjectPool

IDisposable

© Bus Fighter. All rights reserved.

Struct ObjectPool<T>.Handle

Implements

[IDisposable](#)

Inherited Members

[ValueType.Equals\(object\)](#)

[ValueType.GetHashCode\(\)](#)

[ValueType.ToString\(\)](#)

[object.Equals\(object, object\)](#)

[object.GetType\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfEngine.Pooling](#)

Assembly: CatSweeper.dll

Syntax

```
public struct ObjectPool<T>.Handle : IDisposable
```

Constructors

Handle(Action<T>, T)

Declaration

```
public Handle(Action<T> releaseAction, T obj)
```

Parameters

TYPE	NAME
Action<T>	releaseAction
T	obj

Properties

Empty

Declaration

```
public static ObjectPool<T>.Handle Empty { get; }
```

Property Value

TYPE

[ObjectPool<T>.Handle](#)

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

Implements

[IDisposable](#)

© Bus Fighter. All rights reserved.

Class PoolManager

Inheritance

[object](#) → [PoolManager](#)

Implements

[IService](#)

[IDisposable](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.Pooling](#)

Assembly: CatSweeper.dll

Syntax

```
public class PoolManager : IService, IDisposable
```

Methods

AddPool<T>(string, T)

Declaration

```
public void AddPool<T>(string key, T pool) where T : IObjectPool
```

Parameters

TYPE	NAME
------	------

string	key
--------	-----

T	pool
---	------

Type Parameters

NAME

T

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

GetOrCreatePool<T>(string, Func<T>)

Declaration

```
public T GetOrCreatePool<T>(string key, Func<T> createFunc) where T : class, IObjectPool
```

Parameters

TYPE	NAME
------	------

string	key
--------	-----

Func<T>	createFunc
---------	------------

Returns

TYPE

T

Type Parameters

NAME

T

TryGetPool(string, out IObjectPool)

Declaration

```
public bool TryGetPool(string key, out IObjectPool pool)
```

Parameters

TYPE	NAME
string	key
IObjectPool	pool

Returns

TYPE
bool

TryGetPool<T>(string, out T)

Declaration

```
public bool TryGetPool<T>(string key, out T pool) where T : class, IObjectPool
```

Parameters

TYPE	NAME
string	key
T	pool

Returns

TYPE
bool

Type Parameters

NAME
T

Implements

© Bus Fighter. All rights reserved.

Class StringBuilderPool

Inheritance

object → [ObjectPool<StringBuilder>](#) → [StringBuilderPool](#)

Implements

[IObjectPool](#)

[IDisposable](#)

Inherited Members

[ObjectPool<StringBuilder>.Queue](#)

[ObjectPool<StringBuilder>.Get\(\)](#)

[ObjectPool<StringBuilder>.Get\(out StringBuilder\)](#)

[ObjectPool<StringBuilder>.Release\(StringBuilder\)](#)

[ObjectPool<StringBuilder>.Dispose\(\)](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.Pooling](#)

Assembly: CatSweeper.dll

Syntax

```
public class StringBuilderPool : ObjectPool<StringBuilder>, IObjectPool, IDisposable
```

Constructors

[StringBuilderPool\(\)](#)

Declaration

```
public StringBuilderPool()
```

Properties

Default

Declaration

```
public static StringBuilderPool Default { get; }
```

Property Value

TYPE

StringBuilderPool

Implements

[IObjectPool](#)

[IDisposable](#)

© Bus Fighter. All rights reserved.

Namespace cfEngine.Rx

Classes

[CollectionEventExtension](#)

[CollectionEvents<T>](#)

[Linq](#)

[Relay](#)

[RelayBase<TDelegate>](#)

[Relay<T>](#)

[Relay<T1, T2>](#)

[RtCollection<TEventArgs>](#)

[RtCount<T>](#)

[RtDictionary< TKey, TValue >](#)

[RtFilteredDictionary< TKey, TValue >](#)

Represents a dictionary that filters its elements based on a provided function.

[RtGroup< TKey, TValue >](#)

[RtList<T>](#)

[RtMutatedDictionaryBase< TSourceKey, TSourceValue, TKey, TValue >](#)

Represents a base class for dictionaries that support mutation and event dispatching.

[RtMutatedListBase< TOrig, TNew >](#)

[RtMutatedLocalListBase< TOrig, TNew >](#)

[RtMutatedSingleBase< TOrig, TNew >](#)

[RtObserverList<T>](#)

[RtReadOnlyDictionary< TKey, TValue >](#)

[RtReadOnlyList<T>](#)

[RtSelectKeyDictionary< TOrigKey, TSelectKey, TValue >](#)

Represents a dictionary that selects keys based on a provided function.

RtSelectList<TOrig, TNew>

Represents a list that projects each element of a source list into a new form.

RtSelectLocalList<T, TSelect>

RtSelectValueDictionary< TKey, TOrigValue, TValue>

Represents a dictionary that selects values based on a provided function.

Rt<T>

Subscription

SubscriptionBinding<TDelegate>

SubscriptionGroup

SubscriptionHandleExtension

Interfaces

ICollectionEvents<T>

IRelay<TDelegate>

© Bus Fighter. All rights reserved.

Class CollectionEventExtension

Inheritance

[object](#) → CollectionEventExtension

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public static class CollectionEventExtension
```

Methods

OnChange<T>(ICollectionEvents<T>, Action)

Declaration

```
public static Subscription OnChange<T>(this ICollectionEvents<T> collectionEvents, Action acti
```

Parameters

TYPE	NAME
ICollectionEvents<T>	collectionEvents
Action	action

Returns

TYPE

Subscription

Type Parameters

NAME

T

Subscribe<T>(ICollectionEvents<T>, Action<T>, Action<T>, Action<T, T>, Action)

Declaration

```
public static Subscription Subscribe<T>(this ICollectionEvents<T> collectionEvents, Action<T>
```

Parameters

TYPE	NAME
ICollectionEvents<T>	collectionEvents
Action<T>	onAdd
Action<T>	onRemove
Action<T, T>	onUpdate
Action	onDispose

Returns

TYPE

Subscription

Type Parameters

NAME

T

Class CollectionEvents<T>

Inheritance

`object` → `CollectionEvents<T>`

Implements

`ICollectionEvents<T>`

Inherited Members

`object.Equals(object)`

`object.Equals(object, object)`

`object.GetHashCode()`

`object.GetType()`

`object.MemberwiseClone()`

`object.ReferenceEquals(object, object)`

`object.ToString()`

Namespace: `cfEngine.Rx`

Assembly: `CatSweeper.dll`

Syntax

```
public class CollectionEvents<T> : ICollectionEvents<T>
```

Type Parameters

NAME

T

Constructors

CollectionEvents(object)

Declaration

```
public CollectionEvents(object owner)
```

Parameters

TYPE	NAME
------	------

object	owner
--------	-------

Fields

OnAddRelay

Declaration

```
public readonly Relay<T> OnAddRelay
```

Field Value

TYPE

Relay<T>

OnDisposeRelay

Declaration

```
public readonly Relay OnDisposeRelay
```

Field Value

TYPE

Relay

OnRemoveRelay

Declaration

```
public readonly Relay<T> OnRemoveRelay
```

Field Value

TYPE

Relay<T>

OnUpdateRelay

Declaration

```
public readonly Relay<T, T> OnUpdateRelay
```

Field Value

TYPE

Relay<T, T>

Methods

Dispose()

Declaration

```
public void Dispose()
```

SubscribeOnAdd(Action<T>)

Declaration

```
public Subscription SubscribeOnAdd(Action<T> onAdd)
```

Parameters

TYPE	NAME
------	------

Action<T>	onAdd
-----------	-------

Returns

TYPE

Subscription

SubscribeOnDispose(Action)

Declaration

```
public Subscription SubscribeOnDispose(Action onDispose)
```

Parameters

TYPE	NAME
------	------

Action	onDispose
--------	-----------

Returns

TYPE

Subscription

SubscribeOnRemove(Action<T>)

Declaration

```
public Subscription SubscribeOnRemove(Action<T> onRemove)
```

Parameters

TYPE	NAME
------	------

Action<T>	onRemove
-----------	----------

Returns

TYPE

Subscription

SubscribeOnUpdate(Action<T, T>)

Declaration

```
public Subscription SubscribeOnUpdate(Action<T, T> onUpdate)
```

Parameters

TYPE	NAME
Action<T, T>	onUpdate

Returns

TYPE
Subscription

Implements

ICollectionEvents<T>

Extension Methods

CollectionEventExtension.OnChange<T>(ICollectionEvents<T>, Action)

CollectionEventExtension.Subscribe<T>(ICollectionEvents<T>, Action<T>, Action<T>, Action<T, T>, Action)

© Bus Fighter. All rights reserved.

Interface ICollectionEvents<T>

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public interface ICollectionEvents<out T>
```

Type Parameters

NAME

T

Methods

Dispose()

Declaration

```
void Dispose()
```

SubscribeOnAdd(Action<T>)

Declaration

```
Subscription SubscribeOnAdd(Action<out T> onAdd)
```

Parameters

TYPE	NAME
------	------

Action<T>	onAdd
-----------	-------

Returns

TYPE

Subscription

SubscribeOnDispose(Action)

Declaration

Subscription SubscribeOnDispose(Action onDispose)

Parameters

TYPE NAME

Action onDispose

Returns

TYPE

Subscription

SubscribeOnRemove(Action<T>)

Declaration

Subscription SubscribeOnRemove(Action<out T> onRemove)

Parameters

TYPE NAME

Action<T> onRemove

Returns

TYPE

Subscription

SubscribeOnUpdate(Action<T, T>)

Declaration

```
Subscription SubscribeOnUpdate(Action<out T, out T> onUpdate)
```

Parameters

TYPE	NAME
Action<T, T>	onUpdate

Returns

TYPE
Subscription

Extension Methods

[CollectionEventExtension.OnChange<T>\(ICollectionEvents<T>, Action\)](#)

[CollectionEventExtension.Subscribe<T>\(ICollectionEvents<T>, Action<T>, Action<T>, Action<T, T>, Action\)](#)

© Bus Fighter. All rights reserved.

Interface IRelay<TDelegate>

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IRelay<TDelegate> where TDelegate : class
```

Type Parameters

NAME

TDelegate

Methods

AddListener(TDelegate)

Declaration

```
Subscription AddListener(TDelegate listener)
```

Parameters

TYPE	NAME
------	------

TDelegate listener

Returns

TYPE

Subscription

Contains(TDelegate)

Declaration

```
bool Contains(TDelegate d)
```

Parameters

TYPE	NAME
TDelegate	d

Returns

TYPE
bool

RemoveAll()

Declaration

```
void RemoveAll()
```

RemoveListener(TDelegate)

Declaration

```
bool RemoveListener(TDelegate listener)
```

Parameters

TYPE	NAME
TDelegate	listener

Returns

TYPE
bool

Class Linq

Inheritance

`object` → `Linq`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public static class Linq
```

Methods

count<T>(RtReadOnlyList<T>)

Declaration

```
public static RtCount<T> count<T>(this RtReadOnlyList<T> source)
```

Parameters

TYPE	NAME
<code>RtReadOnlyList<T></code>	<code>source</code>

Returns

TYPE

RtCount<T>

Type Parameters

NAME

T

groupBy< TKey, TValue >(RtReadOnlyList< TValue >, Func< TValue, TKey >)

Declaration

```
public static RtGroup< TKey, TValue > groupBy< TKey, TValue >(this RtReadOnlyList< TValue > rtList,
```

Parameters

TYPE	NAME
RtReadOnlyList< TValue >	rtList
Func< TValue, TKey >	keyFn

Returns

TYPE

RtGroup< TKey, TValue >

Type Parameters

NAME

TKey

TValue

selectKey< TOrigKey, TSelectedKey, TValue >
(RtReadOnlyDictionary< TOrigKey, TValue >, Func< TOrigKey, TSelectedKey >)

Declaration

```
public static RtSelectKeyDictionary<TOrigKey, TSelectedKey, TValue> selectKey<TOrigKey, TSelectedKey, TValue>(this RtReadOnlyDictionary<TOrigKey, TValue> source, Func<TOrigKey, TSelectedKey> selectFn)
```

Parameters

TYPE	NAME
RtReadOnlyDictionary<TOrigKey, TValue>	source
Func<TOrigKey, TSelectedKey>	selectFn

Returns

TYPE
RtSelectKeyDictionary<TOrigKey, TSelectedKey, TValue>

Type Parameters

NAME
TOrigKey
TSelectedKey
TValue

selectNew<T, TSelect>(RtReadOnlyList<T>, Func<T, TSelect>)

Declaration

```
public static RtSelectList<T, TSelect> selectNew<T, TSelect>(this RtReadOnlyList<T> source, Func<T, TSelect> selectFn)
```

Parameters

TYPE	NAME
RtReadOnlyList<T>	source
Func<T, TSelect>	selectFn

Returns

TYPE
RtSelectList<T, TSelect>

Type Parameters

NAME

T

TSelect

selectValue<TKey, TValue, TSelectValue> (RtReadOnlyDictionary<TKey, TValue>, Func<TValue, TSelectValue>)

Declaration

```
public static RtSelectValueDictionary<TKey, TValue, TSelectValue> selectValue<TKey, TValue, TS
```

Parameters

TYPE	NAME
RtReadOnlyDictionary<TKey, TValue>	source
Func<TValue, TSelectValue>	selectFn

Returns

TYPE
RtSelectValueDictionary<TKey, TValue, TSelectValue>

Type Parameters

NAME
TKey
TValue
TSelectValue

select<T, TSelect>(RtReadOnlyList<T>, Func<T, TSelect>)

Declaration

```
public static RtSelectLocalList<T, TSelect> select<T, TSelect>(this RtReadOnlyList<T> source,
```

Parameters

TYPE**NAME**

RtReadOnlyList<T> source

Func<T, TSelect> selectFn

Returns**TYPE**

RtSelectLocalList<T, TSelect>

Type Parameters**NAME**

T

TSelect

where<TKey, TValue>(RtReadOnlyDictionary<TKey, TValue>, Func<KeyValuePair<TKey, TValue>, bool>)

Declaration

```
public static RtFilteredDictionary<TKey, TValue> where<TKey, TValue>(this RtReadOnlyDictionary
```

Parameters**TYPE****NAME**

RtReadOnlyDictionary<TKey, TValue> source

Func<KeyValuePair<TKey, TValue>, bool> filterFn

Returns**TYPE**

RtFilteredDictionary<TKey, TValue>

Type Parameters**NAME**

TKey

TValue

Class Relay

Inheritance

object → [RelayBase<Action>](#) → Relay

Implements

[IRelay<Action>](#)

Inherited Members

[RelayBase<Action>._subscriptionRefList](#)

[RelayBase<Action>._cap](#)

[RelayBase<Action>._count](#)

[RelayBase<Action>.listenerCount](#)

[RelayBase<Action>.AddListener\(Action\)](#)

[RelayBase<Action>.RemoveListener\(Action\)](#)

[RelayBase<Action>.RemoveAll\(\)](#)

[RelayBase<Action>.Contains\(Action\)](#)

[RelayBase<Action>.Expand\(ref WeakReference<SubscriptionBinding<Action>>\[\]\)](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class Relay : RelayBase<Action>, IRelay<Action>
```

Constructors

[Relay\(object, int\)](#)

```
public Relay(object owner, int defaultSize = 1)
```

Parameters

TYPE	NAME
object	owner
int	defaultSize

Methods

Dispatch()

Declaration

```
public void Dispatch()
```

Implements

[IRelay<TDelegate>](#)

© Bus Fighter. All rights reserved.

Class RelayBase<TDelegate>

Inheritance

object → RelayBase<TDelegate> → [Relay](#) → [Relay<T>](#) → [Relay<T1, T2>](#)

Implements

[IRelay<TDelegate>](#)

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class RelayBase<TDelegate> : IRelay<TDelegate> where TDelegate : class
```

Type Parameters

NAME

TDelegate

Constructors

RelayBase(object, int)

Declaration

```
public RelayBase(object owner, int defaultSize = 1)
```

Parameters

TYPE	NAME
------	------

object	owner
--------	-------

int	defaultSize
-----	-------------

Fields

_cap

Declaration

```
protected int _cap
```

Field Value

TYPE

int

_count

Declaration

```
protected int _count
```

Field Value

TYPE

int

_subscriptionRefList

Declaration

```
protected WeakReference<SubscriptionBinding<TDelegate>>[] _subscriptionRefList
```

Field Value

TYPE

WeakReference<SubscriptionBinding<TDelegate>>[]

Properties

listenerCount

Declaration

```
public int listenerCount { get; }
```

Property Value

TYPE

int

Methods

AddListener(TDelegate)

Declaration

```
public Subscription AddListener(TDelegate listener)
```

Parameters

TYPE	NAME
------	------

TDelegate	listener
-----------	----------

Returns

TYPE

Subscription

Contains(TDelegate)

Declaration

```
public bool Contains(TDelegate d)
```

Parameters

TYPE	NAME
TDelegate	d

Returns

TYPE
bool

Expand(ref WeakReference<SubscriptionBinding<TDelegate>>[])

Declaration

```
public int Expand(ref WeakReference<SubscriptionBinding<TDelegate>>[] bindings)
```

Parameters

TYPE	NAME
WeakReference<SubscriptionBinding<TDelegate>>[]	bindings

Returns

TYPE
int

RemoveAll()

Declaration

```
public void RemoveAll()
```

RemoveListener(TDelegate)

Declaration

```
public bool RemoveListener(TDelegate listener)
```

Parameters

TYPE	NAME
TDelegate	listener

Returns

TYPE
bool

Implements

[IRelay<TDelegate>](#)

© Bus Fighter. All rights reserved.

Class Relay<T>

Inheritance

object → RelayBase<Action<T>> → Relay<T>

Implements

IRelay<Action<T>>

Inherited Members

RelayBase<Action<T>>._subscriptionRefList

RelayBase<Action<T>>._cap

RelayBase<Action<T>>._count

RelayBase<Action<T>>.listenerCount

RelayBase<Action<T>>.AddListener(Action<T>)

RelayBase<Action<T>>.RemoveListener(Action<T>)

RelayBase<Action<T>>.RemoveAll()

RelayBase<Action<T>>.Contains(Action<T>)

RelayBase<Action<T>>.Expand(ref WeakReference<SubscriptionBinding<Action<T>>>[])

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class Relay<T> : RelayBase<Action<T>>, IRelay<Action<T>>
```

Type Parameters

NAME

T

Constructors

Relay(object, int)

Declaration

```
public Relay(object owner, int defaultSize = 1)
```

Parameters

TYPE	NAME
object	owner
int	defaultSize

Methods

Dispatch(T)

Declaration

```
public void Dispatch(T value1)
```

Parameters

TYPE	NAME
T	value1

Implements

[IRelay<TDelegate>](#)

© Bus Fighter. All rights reserved.

Class Relay<T1, T2>

Inheritance

object → RelayBase<Action<T1, T2>> → Relay<T1, T2>

Implements

IRelay<Action<T1, T2>>

Inherited Members

RelayBase<Action<T1, T2>>._subscriptionRefList

RelayBase<Action<T1, T2>>._cap

RelayBase<Action<T1, T2>>._count

RelayBase<Action<T1, T2>>.listenerCount

RelayBase<Action<T1, T2>>.AddListener(Action<T1, T2>)

RelayBase<Action<T1, T2>>.RemoveListener(Action<T1, T2>)

RelayBase<Action<T1, T2>>.RemoveAll()

RelayBase<Action<T1, T2>>.Contains(Action<T1, T2>)

RelayBase<Action<T1, T2>>.Expand(ref WeakReference<SubscriptionBinding<Action<T1, T2>>>[])

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class Relay<T1, T2> : RelayBase<Action<T1, T2>>, IRelay<Action<T1, T2>>
```

Type Parameters

NAME

T1

T2

Constructors

Relay(object, int)

Declaration

```
public Relay(object owner, int defaultSize = 1)
```

Parameters

TYPE	NAME
object	owner
int	defaultSize

Methods

Dispatch(T1, T2)

Declaration

```
public void Dispatch(T1 value1, T2 value2)
```

Parameters

TYPE	NAME
T1	value1
T2	value2

Implements

[IRelay<TDelegate>](#)

Class RtCollection<TEventArgs>

Inheritance

object → RtCollection<TEventArgs> → [RtReadOnlyDictionary< TKey, TValue >](#) → [RtReadOnlyList< T >](#)

Implements

[IDisposable](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class RtCollection<TEventArgs> : IDisposable
```

Type Parameters

NAME

TEventArgs

Constructors

RtCollection()

Declaration

```
protected RtCollection()
```

Properties

CollectionEvents

Declaration

```
protected CollectionEvents<TEventArgs> CollectionEvents { get; }
```

Property Value

TYPE

CollectionEvents<TEventArgs>

Events

Declaration

```
public ICollectionEvents<TEventArgs> Events { get; }
```

Property Value

TYPE

ICollectionEvents<TEventArgs>

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public virtual void Dispose()
```

Implements

© Bus Fighter. All rights reserved.

Class RtCount<T>

Inheritance

object → RtCollection<(int index, int item)> → RtReadOnlyList<int> → RtMutatedLocalListBase<T, int> → RtMutatedSingleBase<T, int> → RtCount<T>

Implements

IDisposable
IReadOnlyList<int>
IReadOnlyCollection<int>
IEnumerable<int>
IEnumerable

Inherited Members

RtMutatedSingleBase<T, int>.Value
RtMutatedSingleBase<T, int>.Dispose()
RtMutatedSingleBase<T, int>.GetEnumerator()
RtMutatedSingleBase<T, int>.Count
RtMutatedSingleBase<T, int>.this[int]
RtMutatedLocalListBase<T, int>.Dispose()
RtMutatedLocalListBase<T, int>._OnSourceUpdate((int index, T item), (int index, T item))
RtMutatedLocalListBase<T, int>._OnSourceRemove((int index, T item))
RtMutatedLocalListBase<T, int>._OnSourceAdd((int index, T item))
RtReadOnlyList<int>.GetEnumerator()
RtReadOnlyList<int>.Count
RtReadOnlyList<int>.this[int]
RtCollection<(int index, int item)>.CollectionEvents
RtCollection<(int index, int item)>.Events
RtCollection<(int index, int item)>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class RtCount<T> : RtMutatedSingleBase<T, int>, IDisposable, IReadOnlyList<int>, IReadOnly
```

Type Parameters

NAME

T

Constructors

RtCount(RtReadOnlyList<T>)

Declaration

```
public RtCount(RtReadOnlyList<T> source)
```

Parameters

TYPE	NAME
RtReadOnlyList<T>	source

Methods

_OnSourceAdd((int index, T item))

Declaration

```
protected override void _OnSourceAdd((int index, T item) item)
```

Parameters

TYPE	NAME
(int index, T item)	item

Overrides

[RtMutatedLocalListBase<T, int>._OnSourceAdd\(\(int index, T item\)\)](#)

_OnSourceRemove((int index, T item))

Declaration

```
protected override void _OnSourceRemove((int index, T item) item)
```

Parameters

TYPE	NAME
(int index, T item)	item

Overrides

[RtMutatedLocalListBase<T, int>._OnSourceRemove\(\(int index, T item\)\)](#)

_OnSourceUpdate((int index, T item), (int index, T item))

Declaration

```
protected override void _OnSourceUpdate((int index, T item) oldItem, (int index, T item) newItem)
```

Parameters

TYPE	NAME
(int index, T item)	oldItem
(int index, T item)	newItem

Overrides

[RtMutatedLocalListBase<T, int>._OnSourceUpdate\(\(int index, T item\), \(int index, T item\)\)](#)

Implements

[IDisposable](#)
 [IReadOnlyList<T>](#)
 [IReadOnlyCollection<T>](#)
 [IEnumerable<T>](#)
 [IEnumerable](#)

Extension Methods

```
Linq.count<T>(RtReadOnlyList<T>)
Linq.groupBy< TKey, TValue>(RtReadOnlyList< TValue>, Func< TValue, TKey>)
Linq.selectNew< T, TSelect>(RtReadOnlyList< T>, Func< T, TSelect>)
Linq.select< T, TSelect>(RtReadOnlyList< T>, Func< T, TSelect>)
```

© Bus Fighter. All rights reserved.

Class RtDictionary<TKey, TValue>

Inheritance

object → RtCollection<KeyValuePair<TKey, TValue>> → RtReadOnlyDictionary<TKey, TValue> → RtDictionary<TKey, TValue>

Implements

IDisposable

IReadOnlyDictionary<TKey, TValue>

IReadOnlyCollection<KeyValuePair<TKey, TValue>>

IEnumerable<KeyValuePair<TKey, TValue>>

IEnumerable

Inherited Members

RtReadOnlyDictionary<TKey, TValue>.Count

RtReadOnlyDictionary<TKey, TValue>.ContainsKey(TKey)

RtReadOnlyDictionary<TKey, TValue>.TryGetValue(TKey, out TValue)

RtReadOnlyDictionary<TKey, TValue>.this[TKey]

RtReadOnlyDictionary<TKey, TValue>.Keys

RtReadOnlyDictionary<TKey, TValue>.Values

RtReadOnlyDictionary<TKey, TValue>.RtPairs

RtReadOnlyDictionary<TKey, TValue>.RtKeys

RtReadOnlyDictionary<TKey, TValue>.RtValues

RtReadOnlyDictionary<TKey, TValue>.GetEnumerator()

RtCollection<KeyValuePair<TKey, TValue>>.CollectionEvents

RtCollection<KeyValuePair<TKey, TValue>>.Events

RtCollection<KeyValuePair<TKey, TValue>>.Dispose()

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class RtDictionary<TKey, TValue> : RtReadOnlyDictionary<TKey, TValue>, IDisposable, IRe
```

Type Parameters

NAME

TKey

TValue

Properties

Count

Gets the number of elements in the collection.

Declaration

```
public override int Count { get; }
```

Property Value

TYPE	DESCRIPTION
------	-------------

int	The number of elements in the collection.
-----	---

Overrides

[RtReadOnlyDictionary<TKey, TValue>.Count](#)

IsReadOnly

Declaration

```
public bool IsReadOnly { get; }
```

Property Value

TYPE

bool

this[TKey]

Gets the element that has the specified key in the read-only dictionary.

Declaration

```
public override TValue this[TKey key] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
TKey	key	The key to locate.

Property Value

TYPE	DESCRIPTION
TValue	The element that has the specified key in the read-only dictionary.

Overrides

[RtReadOnlyDictionary<TKey, TValue>.this\[TKey\]](#)

Exceptions

TYPE	CONDITION
ArgumentNullException	key is null.
KeyNotFoundException	The property is retrieved and key is not found.

Keys

Gets an enumerable collection that contains the keys in the read-only dictionary.

Declaration

```
public override IEnumerable<TKey> Keys { get; }
```

Property Value

TYPE	DESCRIPTION
IEnumerable<TKey>	An enumerable collection that contains the keys in the read-only dictionary.

Overrides

[RtReadOnlyDictionary<TKey, TValue>.Keys](#)

Values

Gets an enumerable collection that contains the values in the read-only dictionary.

Declaration

```
public override IEnumerable<TValue> Values { get; }
```

Property Value

TYPE	DESCRIPTION
IEnumerable<TValue>	An enumerable collection that contains the values in the read-only dictionary.

Overrides

[RtReadOnlyDictionary< TKey, TValue >.Values](#)

Methods

Add(KeyValuePair< TKey, TValue >)

Adds a key-value pair to the dictionary.

Declaration

```
public void Add(KeyValuePair< TKey, TValue > kvp)
```

Parameters

TYPE	NAME	DESCRIPTION
KeyValuePair< TKey, TValue >	kvp	The key-value pair to add.

Exceptions

TYPE	CONDITION
ArgumentException	Thrown when the key already exists.

Add(TKey, TValue)

Declaration

```
public void Add(TKey key, TValue value)
```

Parameters

TYPE	NAME
TKey	key
TValue	value

Clear()

Declaration

```
public void Clear()
```

Contains(KeyValuePair<TKey, TValue>)

Declaration

```
public bool Contains(KeyValuePair<TKey, TValue> kvp)
```

Parameters

TYPE	NAME
KeyValuePair<TKey, TValue>	kvp

Returns

TYPE
bool

ContainsKey(TKey)

Determines whether the read-only dictionary contains an element that has the specified key.

Declaration

```
public override bool ContainsKey(TKey key)
```

Parameters

TYPE	NAME	DESCRIPTION
------	------	-------------

TKey	key	The key to locate.
------	-----	--------------------

Returns

TYPE	DESCRIPTION
------	-------------

bool	true if the read-only dictionary contains an element that has the specified key; otherwise, false.
------	--

Overrides

[RtReadOnlyDictionary<TKey, TValue>.ContainsKey\(TKey\)](#)

Exceptions

TYPE	CONDITION
------	-----------

ArgumentNullException	key is null.
-----------------------	--------------

CopyTo(KeyValuePair<TKey, TValue>[], int)

Declaration

```
public void CopyTo(KeyValuePair<TKey, TValue>[] array, int arrayIndex)
```

Parameters

TYPE	NAME
KeyValuePair<TKey, TValue>[]	array
int	arrayIndex

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public override void Dispose()
```

Overrides

[RtCollection<KeyValuePair<TKey, TValue>>.Dispose\(\)](#)

EnsureCapacity(int)

Declaration

```
public void EnsureCapacity(int capacity)
```

Parameters

TYPE	NAME
------	------

int	capacity
-----	----------

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public override IEnumerator<KeyValuePair<TKey, TValue>> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
IEnumerator<KeyValuePair<TKey, TValue>>	An enumerator that can be used to iterate through the collection.

Overrides

[RtReadOnlyDictionary< TKey, TValue >.GetEnumerator\(\)](#)

Remove(KeyValuePair< TKey, TValue >)

Declaration

```
public bool Remove(KeyValuePair<TKey, TValue> kvp)
```

Parameters

TYPE	NAME
------	------

KeyValuePair< TKey, TValue >	kvp
------------------------------	-----

Returns

TYPE

bool

Remove(TKey)

Declaration

```
public bool Remove(TKey key)
```

Parameters

TYPE NAME

TKey key

Returns

TYPE

bool

TryGetValue(TKey, out TValue)

Gets the value that is associated with the specified key.

Declaration

```
public override bool TryGetValue(TKey key, out TValue value)
```

Parameters

TYPE NAME DESCRIPTION

TKey key The key to locate.

TValue value When this method returns, the value associated with the specified key, if the key is found; otherwise, the default value for the type of the **value** parameter. This parameter is passed uninitialized.

Returns

TYPE DESCRIPTION

bool true if the object that implements the **IReadOnlyDictionary<TKey, TValue>** interface contains an element that has the specified key; otherwise, false.

Overrides

RtReadOnlyDictionary< TKey, TValue>.TryGetValue(TKey, out TValue)

Exceptions

TYPE	CONDITION
ArgumentNullException	key is null.

Upsert(TKey, TValue)

Declaration

```
public void Upsert(TKey key, TValue value)
```

Parameters

TYPE	NAME
TKey	key
TValue	value

Implements

IDisposable

IReadOnlyDictionary< TKey, TValue>

IReadOnlyCollection< T >

IEnumerable< T >

IEnumerable

Extension Methods

Linq.selectKey< TOrigKey, TSelectedKey, TValue>(RtReadOnlyDictionary< TOrigKey, TValue>, Func< TOrigKey, TSelectedKey>)

Linq.selectValue< TKey, TValue, TSelectValue>(RtReadOnlyDictionary< TKey, TValue>, Func< TValue, TSelectValue>)

Linq.where< TKey, TValue>(RtReadOnlyDictionary< TKey, TValue>, Func< KeyValuePair< TKey, TValue>, bool>)

Class RtFilteredDictionary<TKey, TValue>

Represents a dictionary that filters its elements based on a provided function.

Inheritance

object → RtCollection<KeyValuePair<TKey, TValue>> → RtReadOnlyDictionary<TKey, TValue> → RtMutatedDictionaryBase<TKey, TValue, TKey, TValue> → RtFilteredDictionary<TKey, TValue>

Implements

IDisposable

IReadOnlyDictionary<TKey, TValue>

IReadOnlyCollection<KeyValuePair<TKey, TValue>>

IEnumerable<KeyValuePair<TKey, TValue>>

IEnumerable

Inherited Members

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>._mutated

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>.Dispose()

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>._OnSourceUpdate(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TValue>, KeyValuePair<TKey, TValue>)

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>._OnSourceRemove(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TValue>)

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>._OnSourceAdd(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TValue>)

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>.GetEnumerator()

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>.Count

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>.ContainsKey(TKey)

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>.TryGetValue(TKey, out TValue)

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>.this[TKey]

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>.Keys

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>.Values

RtReadOnlyDictionary<TKey, TValue>.Count

RtReadOnlyDictionary<TKey, TValue>.ContainsKey(TKey)

RtReadOnlyDictionary<TKey, TValue>.TryGetValue(TKey, out TValue)

RtReadOnlyDictionary<TKey, TValue>.this[TKey]

RtReadOnlyDictionary<TKey, TValue>.Keys

RtReadOnlyDictionary<TKey, TValue>.Values

RtReadOnlyDictionary<TKey, TValue>.RtPairs

RtReadOnlyDictionary<TKey, TValue>.RtKeys

RtReadOnlyDictionary<TKey, TValue>.RtValues

RtReadOnlyDictionary<TKey, TValue>.GetEnumerator()

RtCollection<KeyValuePair<TKey, TValue>>.CollectionEvents

RtCollection<KeyValuePair< TKey, TValue >>.Events
RtCollection<KeyValuePair< TKey, TValue >>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class RtFilteredDictionary<TKey, TValue> : RtMutatedDictionaryBase<TKey, TValue, TKey,
```

Type Parameters

NAME	DESCRIPTION
TKey	The type of keys in the dictionary.
TValue	The type of values in the dictionary.

Constructors

RtFilteredDictionary(RtReadOnlyDictionary< TKey, TValue >, Func<KeyValuePair< TKey, TValue >, bool>)

Declaration

```
public RtFilteredDictionary(RtReadOnlyDictionary< TKey, TValue > source, Func<KeyValuePair< TKey,
```

Parameters

TYPE	NAME
RtReadOnlyDictionary< TKey, TValue >	source
Func<KeyValuePair< TKey, TValue >, bool>	filterFn

Methods

_OnSourceAdd(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TValue>)

Declaration

```
protected override void _OnSourceAdd(in Dictionary<TKey, TValue> mutated, KeyValuePair<TKey, TValue> kvp)
```

Parameters

TYPE	NAME
Dictionary<TKey, TValue>	mutated
KeyValuePair<TKey, TValue>	kvp

Overrides

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>._OnSourceAdd(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TValue>)

_OnSourceRemove(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TValue>)

Declaration

```
protected override void _OnSourceRemove(in Dictionary<TKey, TValue> mutated, KeyValuePair<TKey, TValue> kvp)
```

Parameters

TYPE	NAME
Dictionary<TKey, TValue>	mutated
KeyValuePair<TKey, TValue>	kvp

Overrides

RtMutatedDictionaryBase<TKey, TValue, TKey, TValue>._OnSourceRemove(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TValue>)

_OnSourceUpdate(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TValue>, KeyValuePair<TKey, TValue>)

Declaration

```
protected override void _OnSourceUpdate(in Dictionary< TKey, TValue> mutated, KeyValuePair< TKey
```

Parameters

TYPE	NAME
Dictionary< TKey, TValue>	mutated
KeyValuePair< TKey, TValue>	oldPair
KeyValuePair< TKey, TValue>	newPair

Overrides

RtMutatedDictionaryBase< TKey, TValue, TKey, TValue>._OnSourceUpdate(in Dictionary< TKey, TValue>, KeyValuePair< TKey, TValue>, KeyValuePair< TKey, TValue>)

Implements

IDisposable

IReadOnlyDictionary< TKey, TValue>

IReadOnlyCollection< T>

IEnumerable< T>

IEnumerable

Extension Methods

Linq.selectKey< TOrigKey, TSelectedKey, TValue>(RtReadOnlyDictionary< TOrigKey, TValue>, Func< TOrigKey, TSelectedKey>)

Linq.selectValue< TKey, TValue, TSelectValue>(RtReadOnlyDictionary< TKey, TValue>, Func< TValue, TSelectValue>)

Linq.where< TKey, TValue>(RtReadOnlyDictionary< TKey, TValue>, Func< KeyValuePair< TKey, TValue>, bool>)

© Bus Fighter. All rights reserved.

Class RtGroup<TKey, TValue>

Inheritance

object → RtCollection<KeyValuePair<TKey, RtReadOnlyList<TValue>>> →
RtReadOnlyDictionary<TKey, RtReadOnlyList<TValue>> → RtGroup<TKey, TValue>

Implements

IDisposable
IReadOnlyDictionary<TKey, RtReadOnlyList<TValue>>
IReadOnlyCollection<KeyValuePair<TKey, RtReadOnlyList<TValue>>>
IEnumerable<KeyValuePair<TKey, RtReadOnlyList<TValue>>>
IEnumerable

Inherited Members

RtReadOnlyDictionary<TKey, RtReadOnlyList<TValue>>.Count
RtReadOnlyDictionary<TKey, RtReadOnlyList<TValue>>.ContainsKey(TKey)
RtReadOnlyDictionary<TKey, RtReadOnlyList<TValue>>.TryGetValue(TKey, out RtReadOnlyList<TValue>)
RtReadOnlyDictionary<TKey, RtReadOnlyList<TValue>>.this[TKey]
RtReadOnlyDictionary<TKey, RtReadOnlyList<TValue>>.Keys
RtReadOnlyDictionary<TKey, RtReadOnlyList<TValue>>.Values
RtReadOnlyDictionary<TKey, RtReadOnlyList<TValue>>.RtPairs
RtReadOnlyDictionary<TKey, RtReadOnlyList<TValue>>.RtKeys
RtReadOnlyDictionary<TKey, RtReadOnlyList<TValue>>.RtValues
RtReadOnlyDictionary<TKey, RtReadOnlyList<TValue>>.GetEnumerator()
RtCollection<KeyValuePair<TKey, RtReadOnlyList<TValue>>>.CollectionEvents
RtCollection<KeyValuePair<TKey, RtReadOnlyList<TValue>>>.Events
RtCollection<KeyValuePair<TKey, RtReadOnlyList<TValue>>>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class RtGroup<TKey, TValue> : RtReadOnlyDictionary<TKey, RtReadOnlyList<TValue>>, IDisp
```

Type Parameters

NAME

TKey

TValue

Constructors

RtGroup(RtReadOnlyList<TValue>, Func<TValue, TKey>)

Declaration

```
public RtGroup(RtReadOnlyList<TValue> source, Func<TValue, TKey> keyFn)
```

Parameters

TYPE	NAME
RtReadOnlyList<TValue>	source
Func<TValue, TKey>	keyFn

Properties

Count

Gets the number of elements in the collection.

Declaration

```
public override int Count { get; }
```

Property Value

TYPE DESCRIPTION

int The number of elements in the collection.

Overrides

[RtReadOnlyDictionary< TKey, RtReadOnlyList< TValue > >.Count](#)

this[TKey]

Gets the element that has the specified key in the read-only dictionary.

Declaration

```
public override RtReadOnlyList< TValue > this[TKey key] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
TKey	key	The key to locate.

Property Value

TYPE	DESCRIPTION
RtReadOnlyList< TValue >	The element that has the specified key in the read-only dictionary.

Overrides

[RtReadOnlyDictionary< TKey, RtReadOnlyList< TValue > >.this\[TKey\]](#)

Exceptions

TYPE	CONDITION
ArgumentNullException	key is null.
KeyNotFoundException	The property is retrieved and key is not found.

Keys

Gets an enumerable collection that contains the keys in the read-only dictionary.

Declaration

```
public override IEnumerable< TKey > Keys { get; }
```

Property Value

TYPE	DESCRIPTION
IEnumerable< TKey >	An enumerable collection that contains the keys in the read-only dictionary.

Overrides

[RtReadOnlyDictionary< TKey, RtReadOnlyList< TValue > >.Keys](#)

Values

Gets an enumerable collection that contains the values in the read-only dictionary.

Declaration

```
public override IEnumerable<RtReadOnlyList<TValue>> Values { get; }
```

Property Value

TYPE	DESCRIPTION
IEnumerable<RtReadOnlyList<TValue>>	An enumerable collection that contains the values in the read-only dictionary.

Overrides

[RtReadOnlyDictionary< TKey, RtReadOnlyList< TValue > >.Values](#)

Methods

ContainsKey(TKey)

Determines whether the read-only dictionary contains an element that has the specified key.

Declaration

```
public override bool ContainsKey(TKey key)
```

Parameters

TYPE	NAME	DESCRIPTION
TKey	key	The key to locate.

Returns

TYPE	DESCRIPTION
bool	true if the read-only dictionary contains an element that has the specified key; otherwise, false.

Overrides

[RtReadOnlyDictionary< TKey, RtReadOnlyList< TValue > >.ContainsKey\(TKey\)](#)

Exceptions

TYPE**CONDITION**

ArgumentNullException	key is null.
-----------------------	--------------

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public override void Dispose()
```

Overrides

RtCollection<KeyValuePair< TKey, RtReadOnlyList< TValue>>>.Dispose()

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public override IEnumerator<KeyValuePair< TKey, RtReadOnlyList< TValue>>> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
IEnumerator<KeyValuePair< TKey, RtReadOnlyList< TValue>>>	An enumerator that can be used to iterate through the collection.

Overrides

RtReadOnlyDictionary< TKey, RtReadOnlyList< TValue>>.GetEnumerator()

TryGetValue(TKey, out RtReadOnlyList< TValue>)

Gets the value that is associated with the specified key.

Declaration

```
public override bool TryGetValue(TKey key, out RtReadOnlyList< TValue> value)
```

Parameters

TYPE	NAME	DESCRIPTION
TKey	key	The key to locate.
RtReadOnlyList< TValue >	value	When this method returns, the value associated with the specified key, if the key is found; otherwise, the default value for the type of the value parameter. This parameter is passed uninitialized.

Returns

TYPE	DESCRIPTION
bool	true if the object that implements the IReadOnlyDictionary< TKey, TValue > interface contains an element that has the specified key; otherwise, false.

Overrides

[RtReadOnlyDictionary< TKey, RtReadOnlyList< TValue > >.TryGetValue\(TKey, out RtReadOnlyList< TValue >\)](#)

Exceptions

TYPE	CONDITION
ArgumentNullException	key is null.

Implements

[IDisposable](#)
[IReadOnlyDictionary< TKey, TValue >](#)
[IReadOnlyCollection< T >](#)
[IEnumerable< T >](#)
[IEnumerable](#)

Extension Methods

[Linq.selectKey< TOrigKey, TSelectedKey, TValue >\(RtReadOnlyDictionary< TOrigKey, TValue >, Func< TOrigKey, TSelectedKey >\)](#)
[Linq.selectValue< TKey, TValue, TSelectValue >\(RtReadOnlyDictionary< TKey, TValue >, Func< TValue, TSelectValue >\)](#)
[Linq.where< TKey, TValue >\(RtReadOnlyDictionary< TKey, TValue >, Func< KeyValuePair< TKey, TValue >, bool >\)](#)

Class RtList<T>

Inheritance

object → [RtCollection<\(int index, T item\)>](#) → [RtReadOnlyList<T>](#) → [RtList<T>](#)

Implements

[IDisposable](#)

[IReadOnlyList<T>](#)

[IReadOnlyCollection<T>](#)

[IEnumerable<T>](#)

[IEnumerable](#)

Inherited Members

[RtCollection<\(int index, T item\)>.CollectionEvents](#)

[RtCollection<\(int index, T item\)>.Events](#)

[RtCollection<\(int index, T item\)>.Dispose\(\)](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class RtList<T> : RtReadOnlyList<T>, IDisposable, IReadOnlyList<T>, IReadOnlyCollection
```

Type Parameters

NAME

T

Constructors

RtList()

Declaration

```
public RtList()
```

RtList(IEnumerable<T>)

Declaration

```
public RtList(IEnumerable<T> defaultItems)
```

Parameters

TYPE	NAME
IEnumerable<T>	defaultItems

RtList(int)

Declaration

```
public RtList(int capacity)
```

Parameters

TYPE	NAME
int	capacity

Properties

Count

Gets the number of elements in the collection.

Declaration

```
public override int Count { get; }
```

Property Value

TYPE	DESCRIPTION
------	-------------

int	The number of elements in the collection.
-----	---

Overrides

[RtReadOnlyList<T>.Count](#)

IsReadOnly

Declaration

```
public bool IsReadOnly { get; }
```

Property Value

TYPE

bool

this[int]

Gets the element at the specified index in the read-only list.

Declaration

```
public override T this[int index] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
------	------	-------------

int	index	The zero-based index of the element to get.
-----	-------	---

Property Value

TYPE	DESCRIPTION
------	-------------

T	The element at the specified index in the read-only list.
---	---

Overrides

Methods

Add(T)

Declaration

```
public void Add(T item)
```

Parameters

TYPE	NAME
------	------

T	item
---	------

Clear()

Declaration

```
public void Clear()
```

Contains(T)

Declaration

```
public bool Contains(T item)
```

Parameters

TYPE	NAME
------	------

T	item
---	------

Returns

TYPE

bool

CopyTo(T[], int)

Declaration

```
public void CopyTo(T[] array, int arrayIndex)
```

Parameters

TYPE	NAME
T[]	array
int	arrayIndex

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public override void Dispose()
```

Overrides

[RtCollection<\(int index, T item\)>.Dispose\(\)](#)

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public override IEnumerator<T> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
IEnumerator<T>	An enumerator that can be used to iterate through the collection.

Overrides

[RtReadOnlyList<T>.GetEnumerator\(\)](#)

IndexOf(T)

Declaration

```
public int IndexOf(T item)
```

Parameters

TYPE	NAME
------	------

T	item
---	------

Returns

TYPE

int

Insert(int, T)

Declaration

```
public void Insert(int index, T item)
```

Parameters

TYPE	NAME
------	------

int	index
-----	-------

T	item
---	------

Remove(T)

Declaration

```
public bool Remove(T item)
```

Parameters

TYPE	NAME
------	------

T	item
---	------

Returns

TYPE

bool

RemoveAt(int)

Declaration

```
public void RemoveAt(int index)
```

Parameters

TYPE NAME

int index

Update(int, T)

Declaration

```
public void Update(int index, T item)
```

Parameters

TYPE NAME

int index

T item

Implements

IDisposable

IReadOnlyList<T>

IReadOnlyCollection<T>

IEnumerable<T>

IEnumerable

Extension Methods

```
Linq.count<T>(RtReadOnlyList<T>)
Linq.groupBy< TKey, TValue>(RtReadOnlyList< TValue>, Func< TValue, TKey>)
Linq.selectNew< T, TSelect>(RtReadOnlyList< T>, Func< T, TSelect>)
Linq.select< T, TSelect>(RtReadOnlyList< T>, Func< T, TSelect>)
```

© Bus Fighter. All rights reserved.

Class RtMutatedDictionaryBase<TSourceKey, TValue>

Represents a base class for dictionaries that support mutation and event dispatching.

Inheritance

```
object → RtCollection<KeyValuePair<TKey, TValue>> → RtReadOnlyDictionary<TKey, TValue> →  
RtMutatedDictionaryBase<TSourceKey, TValue> → RtFilteredDictionary<TKey, TValue> →  
RtSelectKeyDictionary<TOrigKey, TSelectKey, TValue> → RtSelectValueDictionary<TKey, TOrigValue, TValue>
```

Implements

```
IDisposable  
IReadOnlyDictionary<TKey, TValue>  
IReadOnlyCollection<KeyValuePair<TKey, TValue>>  
IEnumerable<KeyValuePair<TKey, TValue>>  
IEnumerable
```

Inherited Members

```
RtReadOnlyDictionary<TKey, TValue>.Count  
RtReadOnlyDictionary<TKey, TValue>.ContainsKey(TKey)  
RtReadOnlyDictionary<TKey, TValue>.TryGetValue(TKey, out TValue)  
RtReadOnlyDictionary<TKey, TValue>.this[TKey]  
RtReadOnlyDictionary<TKey, TValue>.Keys  
RtReadOnlyDictionary<TKey, TValue>.Values  
RtReadOnlyDictionary<TKey, TValue>.RtPairs  
RtReadOnlyDictionary<TKey, TValue>.RtKeys  
RtReadOnlyDictionary<TKey, TValue>.RtValues  
RtReadOnlyDictionary<TKey, TValue>.GetEnumerator()  
RtCollection<KeyValuePair<TKey, TValue>>.CollectionEvents  
RtCollection<KeyValuePair<TKey, TValue>>.Events  
RtCollection<KeyValuePair<TKey, TValue>>.Dispose()  
object.Equals(object)  
object.Equals(object, object)  
object.GetHashCode()  
object.GetType()  
object.MemberwiseClone()  
object.ReferenceEquals(object, object)  
object.ToString()
```

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class RtMutatedDictionaryBase<TSourceKey, TValue, TKey, TValue> : RtRead
```

Type Parameters

NAME	DESCRIPTION
TSourceKey	The type of keys in the source dictionary.
TSourceValue	The type of values in the source dictionary.
TKey	The type of keys in the mutated dictionary.
TValue	The type of values in the mutated dictionary.

Constructors

RtMutatedDictionaryBase(ICollectionEvents<KeyValuePair<TSourceKey, TValue>> sourceEvents)

Declaration

```
protected RtMutatedDictionaryBase(ICollectionEvents<KeyValuePair<TSourceKey, TValue>> sourceEvents)
```

Parameters

TYPE	NAME
ICollectionEvents<KeyValuePair<TSourceKey, TValue>>	sourceEvents

Fields

_mutated

Declaration

```
protected readonly Dictionary<TKey, TValue> _mutated
```

Field Value

TYPE

[Dictionary<TKey, TValue>](#)

Properties

Count

Gets the number of elements in the collection.

Declaration

```
public override int Count { get; }
```

Property Value

TYPE	DESCRIPTION
------	-------------

int	The number of elements in the collection.
-----	---

Overrides

[RtReadOnlyDictionary<TKey, TValue>.Count](#)

this[TKey]

Gets the element that has the specified key in the read-only dictionary.

Declaration

```
public override TValue this[TKey key] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
------	------	-------------

TKey	key	The key to locate.
------	-----	--------------------

Property Value

TYPE	DESCRIPTION
------	-------------

TValue	The element that has the specified key in the read-only dictionary.
--------	---

Overrides

Exceptions

TYPE	CONDITION
ArgumentNullException	key is null.
KeyNotFoundException	The property is retrieved and key is not found.

Keys

Gets an enumerable collection that contains the keys in the read-only dictionary.

Declaration

```
public override IEnumerable< TKey> Keys { get; }
```

Property Value

TYPE	DESCRIPTION
IEnumerable< TKey>	An enumerable collection that contains the keys in the read-only dictionary.

Overrides

RtReadOnlyDictionary< TKey, TValue>.Keys

Values

Gets an enumerable collection that contains the values in the read-only dictionary.

Declaration

```
public override IEnumerable< TValue> Values { get; }
```

Property Value

TYPE	DESCRIPTION
IEnumerable< TValue>	An enumerable collection that contains the values in the read-only dictionary.

Overrides

RtReadOnlyDictionary< TKey, TValue>.Values

Methods

ContainsKey(TKey)

Determines whether the read-only dictionary contains an element that has the specified key.

Declaration

```
public override bool ContainsKey(TKey key)
```

Parameters

TYPE	NAME	DESCRIPTION
------	------	-------------

TKey	key	The key to locate.
------	-----	--------------------

Returns

TYPE	DESCRIPTION
------	-------------

bool	true if the read-only dictionary contains an element that has the specified key; otherwise, false.
------	--

Overrides

[RtReadOnlyDictionary<TKey, TValue>.ContainsKey\(TKey\)](#)

Exceptions

TYPE	CONDITION
------	-----------

ArgumentNullException	key is null.
-----------------------	--------------

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public override void Dispose()
```

Overrides

[RtCollection<KeyValuePair<TKey, TValue>>.Dispose\(\)](#)

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public override IEnumarator<KeyValuePair<TKey, TValue>> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
IEnumerator<KeyValuePair<TKey, TValue>>	An enumerator that can be used to iterate through the collection.

Overrides

[RtReadOnlyDictionary<TKey, TValue>.GetEnumerator\(\)](#)

TryGetValue(TKey, out TValue)

Gets the value that is associated with the specified key.

Declaration

```
public override bool TryGetValue(TKey key, out TValue value)
```

Parameters

TYPE	NAME	DESCRIPTION
TKey	key	The key to locate.
TValue	value	When this method returns, the value associated with the specified key, if the key is found; otherwise, the default value for the type of the value parameter. This parameter is passed uninitialized.

Returns

TYPE	DESCRIPTION
bool	true if the object that implements the IReadOnlyDictionary<TKey, TValue> interface contains an element that has the specified key; otherwise, false.

Overrides

[RtReadOnlyDictionary<TKey, TValue>.TryGetValue\(TKey, out TValue\)](#)

Exceptions

TYPE**CONDITION**

ArgumentNullException key is null.

_OnSourceAdd(in Dictionary<TKey, TValue>, KeyValuePair<TSourceKey, TSourceValue>)

Declaration

```
protected abstract void _OnSourceAdd(in Dictionary<TKey, TValue> mutated, KeyValuePair<TSource
```

Parameters

TYPE	NAME
Dictionary<TKey, TValue>	mutated
KeyValuePair<TSourceKey, TSourceValue>	kvp

_OnSourceRemove(in Dictionary<TKey, TValue>, KeyValuePair<TSourceKey, TSourceValue>)

Declaration

```
protected abstract void _OnSourceRemove(in Dictionary<TKey, TValue> mutated, KeyValuePair<TSource
```

Parameters

TYPE	NAME
Dictionary<TKey, TValue>	mutated
KeyValuePair<TSourceKey, TSourceValue>	kvp

_OnSourceUpdate(in Dictionary<TKey, TValue>, KeyValuePair<TSourceKey, TSourceValue>, KeyValuePair<TSourceKey, TSourceValue>)

Declaration

```
protected abstract void _OnSourceUpdate(in Dictionary< TKey, TValue> mutated, KeyValuePair< TSou
```

Parameters

TYPE	NAME
Dictionary< TKey, TValue>	mutated
KeyValuePair< TSourceKey, TValue>	oldPair
KeyValuePair< TSourceKey, TValue>	newPair

Implements

IDisposable
IReadOnlyDictionary< TKey, TValue>
IReadOnlyCollection< T>
IEnumerable< T>
IEnumerable

Extension Methods

Linq.selectKey< TOrigKey, TSelectedKey, TValue>(RtReadOnlyDictionary< TOrigKey, TValue>, Func< TOrigKey, TSelectedKey>)
Linq.selectValue< TKey, TValue, TSelectValue>(RtReadOnlyDictionary< TKey, TValue>, Func< TValue, TSelectValue>)
Linq.where< TKey, TValue>(RtReadOnlyDictionary< TKey, TValue>, Func< KeyValuePair< TKey, TValue>, bool>)

© Bus Fighter. All rights reserved.

Class RtMutatedListBase<TOrig, TNew>

Inheritance

object → RtCollection<(int index, TNew item)> → RtReadOnlyList<TNew> → RtMutatedListBase<TOrig, TNew> → RtSelectList<TOrig, TNew>

Implements

IDisposable
IReadOnlyList<TNew>
IReadOnlyCollection<TNew>
IEnumerable<TNew>
IEnumerable

Inherited Members

RtCollection<(int index, TNew item)>.CollectionEvents
RtCollection<(int index, TNew item)>.Events
RtCollection<(int index, TNew item)>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()
Namespace: [cfEngine.Rx](#)
Assembly: CatSweeper.dll

Syntax

```
public abstract class RtMutatedListBase<TOrig, TNew> : RtReadOnlyList<TNew>, IDisposable, IRea
```

Type Parameters

NAME

TOrig

TNew

Constructors

RtMutatedListBase<ICollectionEvents<(int index, TOrig item)>>

Declaration

```
protected RtMutatedListBase(ICollectionEvents<(int index, TOrig item)> sourceEvents)
```

Parameters

TYPE	NAME
ICollectionEvents<(int index, TOrig item)>	sourceEvents

Fields

_mutated

Declaration

```
protected readonly List<TNew> _mutated
```

Field Value

TYPE
List<TNew>

Properties

Count

Gets the number of elements in the collection.

Declaration

```
public override int Count { get; }
```

Property Value

TYPE	DESCRIPTION
------	-------------

int	The number of elements in the collection.
-----	---

Overrides

[RtReadOnlyList<TNew>.Count](#)

this[int]

Gets the element at the specified index in the read-only list.

Declaration

```
public override TNew this[int index] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
------	------	-------------

int	index	The zero-based index of the element to get.
-----	-------	---

Property Value

TYPE	DESCRIPTION
------	-------------

TNew	The element at the specified index in the read-only list.
------	---

Overrides

[RtReadOnlyList<TNew>.this\[int\]](#)

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public override void Dispose()
```

Overrides

[RtCollection<\(int index, TNew item\)>.Dispose\(\)](#)

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public override IEnumerator<TNew> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
IEnumerator<TNew>	An enumerator that can be used to iterate through the collection.

Overrides

[RtReadOnlyList<TNew>.GetEnumerator\(\)](#)

_OnSourceAdd(List<TNew>, (int index, TOrig item))

Declaration

```
protected abstract void _OnSourceAdd(List<TNew> mutated, (int index, TOrig item) item)
```

Parameters

TYPE	NAME
List<TNew>	mutated
(int index, TOrig item)	item

_OnSourceRemove(List<TNew>, (int index, TOrig item))

Declaration

```
protected abstract void _OnSourceRemove(List<TNew> mutated, (int index, TOrig item) item)
```

Parameters

TYPE	NAME
List<TNew>	mutated
(int index, TOrig item)	item

_OnSourceUpdate(List<TNew>, (int index, TOrig item), (int index, TOrig item))

Declaration

```
protected abstract void _OnSourceUpdate(List<TNew> mutated, (int index, TOrig item) oldItem, (
```

Parameters

TYPE	NAME
List<TNew>	mutated
(int index, TOrig item)	oldItem
(int index, TOrig item)	newItem

Implements

IDisposable
IReadOnlyList<T>
IReadOnlyCollection<T>
IEnumerable<T>
IEnumerable

Extension Methods

Linq.count<T>(RtReadOnlyList<T>)
Linq.groupBy<TKey, TValue>(RtReadOnlyList<TValue>, Func<TValue, TKey>)
Linq.selectNew<T, TSelect>(RtReadOnlyList<T>, Func<T, TSelect>)
Linq.select<T, TSelect>(RtReadOnlyList<T>, Func<T, TSelect>)

© Bus Fighter. All rights reserved.

Class RtMutatedLocalListBase<TOrig, TNew>

Inheritance

object → RtCollection<(int index, TNew item)> → RtReadOnlyList<TNew> → RtMutatedLocalListBase<TOrig, TNew> → RtMutatedSingleBase<TOrig, TNew> → RtSelectLocalList<T, TSelect>

Implements

IDisposable
IReadOnlyList<TNew>
IReadOnlyCollection<TNew>
IEnumerable<TNew>
IEnumerable

Inherited Members

RtReadOnlyList<TNew>.GetEnumerator()
RtReadOnlyList<TNew>.Count
RtReadOnlyList<TNew>.this[int]
RtCollection<(int index, TNew item)>.CollectionEvents
RtCollection<(int index, TNew item)>.Events
RtCollection<(int index, TNew item)>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: **cfEngine.Rx**

Assembly: CatSweeper.dll

Syntax

```
public abstract class RtMutatedLocalListBase<TOrig, TNew> : RtReadOnlyList<TNew>, IDisposable,
```

Type Parameters

NAME

TOrig

TNew

Constructors

RtMutatedLocalListBase(ICollectionEvents<(int index, TOrig item)>)

Declaration

```
protected RtMutatedLocalListBase(ICollectionEvents<(int index, TOrig item)> sourceEvents)
```

Parameters

TYPE	NAME
ICollectionEvents<(int index, TOrig item)>	sourceEvents

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public override void Dispose()
```

Overrides

RtCollection<(int index, TNew item)>.Dispose()

_OnSourceAdd((int index, TOrig item))

Declaration

```
protected abstract void _OnSourceAdd((int index, TOrig item) item)
```

Parameters

TYPE	NAME
(int index, TOrig item)	item

_OnSourceRemove((int index, TOrig item))

Declaration

```
protected abstract void _OnSourceRemove((int index, TOrig item) item)
```

Parameters

TYPE	NAME
(int index, TOrig item)	item

_OnSourceUpdate((int index, TOrig item), (int index, TOrig item))

Declaration

```
protected abstract void _OnSourceUpdate((int index, TOrig item) oldItem, (int index, TOrig item) newItem)
```

Parameters

TYPE	NAME
(int index, TOrig item)	oldItem
(int index, TOrig item)	newItem

Implements

[IDisposable](#)
 [IReadOnlyList<T>](#)
 [IReadOnlyCollection<T>](#)
 [IEnumerable<T>](#)
 [IEnumerable](#)

Extension Methods

[Linq.count<T>\(RtReadOnlyList<T>\)](#)
[Linq.groupBy< TKey, TValue >\(RtReadOnlyList< TValue >, Func< TValue, TKey >\)](#)
[Linq.selectNew< T, TSelect >\(RtReadOnlyList< T >, Func< T, TSelect >\)](#)
[Linq.select< T, TSelect >\(RtReadOnlyList< T >, Func< T, TSelect >\)](#)

Class RtMutatedSingleBase<TOrig, TNew>

Inheritance

object → RtCollection<(int index, TNew item)> → RtReadOnlyList<TNew> → RtMutatedLocalListBase<TOrig, TNew> → RtMutatedSingleBase<TOrig, TNew> → RtCount<T>

Implements

IDisposable
IReadOnlyList<TNew>
IReadOnlyCollection<TNew>
IEnumerable<TNew>
IEnumerable

Inherited Members

RtMutatedLocalListBase<TOrig, TNew>.Dispose()
RtMutatedLocalListBase<TOrig, TNew>._OnSourceUpdate((int index, TOrig item), (int index, TOrig item))
RtMutatedLocalListBase<TOrig, TNew>._OnSourceRemove((int index, TOrig item))
RtMutatedLocalListBase<TOrig, TNew>._OnSourceAdd((int index, TOrig item))
RtCollection<(int index, TNew item)>.CollectionEvents
RtCollection<(int index, TNew item)>.Events
RtCollection<(int index, TNew item)>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class RtMutatedSingleBase<TOrig, TNew> : RtMutatedLocalListBase<TOrig, TNew>,
```

Type Parameters

NAME

TOrig

NAME

TNew

Constructors

RtMutatedSingleBase(ICollectionEvents<(int index, TOrig item)>)

Declaration

```
public RtMutatedSingleBase(ICollectionEvents<(int index, TOrig item)> sourceEvents)
```

Parameters

TYPE	NAME
ICollectionEvents<(int index, TOrig item)>	sourceEvents

Properties

Count

Gets the number of elements in the collection.

Declaration

```
public override int Count { get; }
```

Property Value

TYPE	DESCRIPTION
int	The number of elements in the collection.

Overrides

[RtReadOnlyList<TNew>.Count](#)

this[int]

Gets the element at the specified index in the read-only list.

Declaration

```
public override TNew this[int index] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
int	index	The zero-based index of the element to get.

Property Value

TYPE	DESCRIPTION
TNew	The element at the specified index in the read-only list.

Overrides

[RtReadOnlyList<TNew>.this\[int\]](#)

Value

Declaration

```
public TNew Value { get; protected set; }
```

Property Value

TYPE
TNew

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public override void Dispose()
```

Overrides

RtMutatedLocalListBase<TOrig, TNew>.Dispose()

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public override IEnumerator<TNew> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
IEnumerator<TNew>	An enumerator that can be used to iterate through the collection.

Overrides

RtReadOnlyList<TNew>.GetEnumerator()

Implements

IDisposable

IReadOnlyList<T>

IReadOnlyCollection<T>

IEnumerable<T>

IEnumerable

Extension Methods

Linq.count<T>(RtReadOnlyList<T>)

Linq.groupBy< TKey, TValue >(RtReadOnlyList< TValue >, Func< TValue, TKey >)

Linq.selectNew< T, TSelect >(RtReadOnlyList< T >, Func< T, TSelect >)

Linq.select< T, TSelect >(RtReadOnlyList< T >, Func< T, TSelect >)

Class RtObserverList<T>

Inheritance

object → [RtCollection<\(int index, T item\)>](#) → [RtReadOnlyList<T>](#) → [RtObserverList<T>](#)

Implements

[IDisposable](#)

[IReadOnlyList<T>](#)

[IReadOnlyCollection<T>](#)

[IEnumerable<T>](#)

[IEnumerable](#)

Inherited Members

[RtCollection<\(int index, T item\)>.CollectionEvents](#)

[RtCollection<\(int index, T item\)>.Events](#)

[RtCollection<\(int index, T item\)>.Dispose\(\)](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class RtObserverList<T> : RtReadOnlyList<T>, IDisposable, IReadOnlyList<T>, IReadOnlyCo
```

Type Parameters

NAME

T

Constructors

RtObserverList(IEnumerable<T>, ICollectionEvents<T>)

Declaration

```
public RtObserverList(IEnumerable<T> sourceItems, ICollectionEvents<T> sourceEvents)
```

Parameters

TYPE	NAME
IEnumerable<T>	sourceItems
ICollectionEvents<T>	sourceEvents

Properties

Count

Gets the number of elements in the collection.

Declaration

```
public override int Count { get; }
```

Property Value

TYPE	DESCRIPTION
int	The number of elements in the collection.

Overrides

[RtReadOnlyList<T>.Count](#)

this[int]

Gets the element at the specified index in the read-only list.

Declaration

```
public override T this[int index] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
------	------	-------------

int	index	The zero-based index of the element to get.
-----	-------	---

Property Value

TYPE	DESCRIPTION
------	-------------

T	The element at the specified index in the read-only list.
---	---

Overrides

[RtReadOnlyList<T>.this\[int\]](#)

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public override void Dispose()
```

Overrides

[RtCollection<\(int index, T item\)>.Dispose\(\)](#)

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public override IEnumerator<T> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
------	-------------

IEnumerator<T>	An enumerator that can be used to iterate through the collection.
----------------	---

Overrides

[RtReadOnlyList<T>.GetEnumerator\(\)](#)

Implements

IDisposable

IReadOnlyList<T>

IReadOnlyCollection<T>

IEnumerable<T>

IEnumerable

Extension Methods

Linq.count<T>(RtReadOnlyList<T>)

Linq.groupBy< TKey, TValue >(RtReadOnlyList< TValue >, Func< TValue, TKey >)

Linq.selectNew< T, TSelect >(RtReadOnlyList< T >, Func< T, TSelect >)

Linq.select< T, TSelect >(RtReadOnlyList< T >, Func< T, TSelect >)

© Bus Fighter. All rights reserved.

Class RtReadOnlyDictionary<TKey, TValue>

Inheritance

object → RtCollection<KeyValuePair<TKey, TValue>> → RtReadOnlyDictionary<TKey, TValue> →
RtDictionary<TKey, TValue> → RtGroup<TKey, TValue> →
RtMutatedDictionaryBase<TSourceKey, TValue, TKey, TValue>

Implements

IDisposable
IReadOnlyDictionary<TKey, TValue>
IReadOnlyCollection<KeyValuePair<TKey, TValue>>
IEnumerable<KeyValuePair<TKey, TValue>>
IEnumerable

Inherited Members

RtCollection<KeyValuePair<TKey, TValue>>.CollectionEvents
RtCollection<KeyValuePair<TKey, TValue>>.Events
RtCollection<KeyValuePair<TKey, TValue>>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class RtReadOnlyDictionary<TKey, TValue> : RtCollection<KeyValuePair<TKey, TValue>>
```

Type Parameters

NAME

TKey

TValue

Properties

Count

Gets the number of elements in the collection.

Declaration

```
public abstract int Count { get; }
```

Property Value

TYPE	DESCRIPTION
------	-------------

int	The number of elements in the collection.
-----	---

this[TKey]

Gets the element that has the specified key in the read-only dictionary.

Declaration

```
public abstract TValue this[TKey key] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
------	------	-------------

TKey	key	The key to locate.
------	-----	--------------------

Property Value

TYPE	DESCRIPTION
------	-------------

TValue	The element that has the specified key in the read-only dictionary.
--------	---

Exceptions

TYPE	CONDITION
------	-----------

ArgumentNullException	key is null.
-----------------------	--------------

KeyNotFoundException	The property is retrieved and key is not found.
----------------------	---

Keys

Gets an enumerable collection that contains the keys in the read-only dictionary.

Declaration

```
public abstract IEnumerable<TKey> Keys { get; }
```

Property Value

TYPE	DESCRIPTION
IEnumerable<TKey>	An enumerable collection that contains the keys in the read-only dictionary.

RtKeys

Declaration

```
public RtReadOnlyList<TKey> RtKeys { get; }
```

Property Value

TYPE
RtReadOnlyList<TKey>

RtPairs

Declaration

```
public RtReadOnlyList<KeyValuePair<TKey, TValue>> RtPairs { get; }
```

Property Value

TYPE
RtReadOnlyList<KeyValuePair<TKey, TValue>>

RtValues

Declaration

```
public RtReadOnlyList<TValue> RtValues { get; }
```

Property Value

TYPE

RtReadOnlyList<TValue>

Values

Gets an enumerable collection that contains the values in the read-only dictionary.

Declaration

```
public abstract IEnumerable<TValue> Values { get; }
```

Property Value

TYPE

DESCRIPTION

IEnumerable<TValue> An enumerable collection that contains the values in the read-only dictionary.

Methods

ContainsKey(TKey)

Determines whether the read-only dictionary contains an element that has the specified key.

Declaration

```
public abstract bool ContainsKey(TKey key)
```

Parameters

TYPE NAME DESCRIPTION

TKey key The key to locate.

Returns

TYPE DESCRIPTION

bool true if the read-only dictionary contains an element that has the specified key; otherwise, false.

Exceptions

TYPE	CONDITION
ArgumentNullException	key is null.

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public abstract IEnumerator<KeyValuePair<TKey, TValue>> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
IEnumerator<KeyValuePair<TKey, TValue>>	An enumerator that can be used to iterate through the collection.

TryGetValue(TKey, out TValue)

Gets the value that is associated with the specified key.

Declaration

```
public abstract bool TryGetValue(TKey key, out TValue value)
```

Parameters

TYPE	NAME	DESCRIPTION
TKey	key	The key to locate.
TValue	value	When this method returns, the value associated with the specified key, if the key is found; otherwise, the default value for the type of the value parameter. This parameter is passed uninitialized.

Returns

TYPE	DESCRIPTION
bool	true if the object that implements the IReadOnlyDictionary<TKey, TValue> interface contains an element that has the specified key; otherwise, false.

Exceptions

ArgumentNullException	key is null.
-----------------------	--------------

Implements

IDisposable

IReadOnlyDictionary<TKey, TValue>

IReadOnlyCollection<T>

IEnumerable<T>

IEnumerable

Extension Methods

Linq.selectKey<TOrigKey, TSelectedKey, TValue>(RtReadOnlyDictionary<TOrigKey, TValue>, Func<TOrigKey, TSelectedKey>)

Linq.selectValue< TKey, TValue, TSelectValue>(RtReadOnlyDictionary< TKey, TValue>, Func< TValue, TSelectValue>)

Linq.where< TKey, TValue>(RtReadOnlyDictionary< TKey, TValue>, Func< KeyValuePair< TKey, TValue>, bool>)

© Bus Fighter. All rights reserved.

Class RtReadOnlyList<T>

Inheritance

object → RtCollection<(int index, T item)> → RtReadOnlyList<T> → RtList<T> → RtMutatedListBase<TOrig, TNew> → RtMutatedLocalListBase<TOrig, TNew> → RtObserverList<T> → Rt<T>

Implements

IDisposable

IReadOnlyList<T>

IReadOnlyCollection<T>

IEnumerable<T>

IEnumerable

Inherited Members

RtCollection<(int index, T item)>.CollectionEvents

RtCollection<(int index, T item)>.Events

RtCollection<(int index, T item)>.Dispose()

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class RtReadOnlyList<T> : RtCollection<(int index, T item)>, IDisposable, IReadOnlyList<T>, IReadOnlyCollection<T>, IEnumerable<T>, IEnumerable
```

Type Parameters

NAME

T

Properties

Count

Gets the number of elements in the collection.

Declaration

```
public abstract int Count { get; }
```

Property Value

TYPE	DESCRIPTION
------	-------------

int	The number of elements in the collection.
-----	---

this[int]

Gets the element at the specified index in the read-only list.

Declaration

```
public abstract T this[int index] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
------	------	-------------

int	index	The zero-based index of the element to get.
-----	-------	---

Property Value

TYPE	DESCRIPTION
------	-------------

T	The element at the specified index in the read-only list.
---	---

Methods

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public abstract IEnumerator<T> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
IEnumerator<T>	An enumerator that can be used to iterate through the collection.

Operators

implicit operator RtReadOnlyList<object>(RtReadOnlyList<T>)

Declaration

```
public static implicit operator RtReadOnlyList<object>(RtReadOnlyList<T> list)
```

Parameters

TYPE	NAME
RtReadOnlyList<T>	list

Returns

TYPE
RtReadOnlyList<object>

Implements

IDisposable
IReadOnlyList<T>
IReadOnlyCollection<T>
IEnumerable<T>
IEnumerable

Extension Methods

Linq.count<T>(RtReadOnlyList<T>)
Linq.groupBy< TKey, TValue >(RtReadOnlyList< TValue >, Func< TValue, TKey >)
Linq.selectNew< T, TSelect >(RtReadOnlyList< T >, Func< T, TSelect >)

© Bus Fighter. All rights reserved.

Class RtSelectKeyDictionary<TOrigKey, TSelectKey, TValue>

Represents a dictionary that selects keys based on a provided function.

Inheritance

object → RtCollection<KeyValuePair<TSelectKey, TValue>> → RtReadOnlyDictionary<TSelectKey, TValue> → RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue> → RtSelectKeyDictionary<TOrigKey, TSelectKey, TValue>

Implements

IDisposable

IReadOnlyDictionary<TSelectKey, TValue>

IReadOnlyCollection<KeyValuePair<TSelectKey, TValue>>

IEnumerable<KeyValuePair<TSelectKey, TValue>>

IEnumerable

Inherited Members

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>._mutated

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>.Dispose()

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>._OnSourceUpdate(in Dictionary<TSelectKey, TValue>, KeyValuePair<TOrigKey, TValue>, KeyValuePair<TOrigKey, TValue>)

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>._OnSourceRemove(in Dictionary<TSelectKey, TValue>, KeyValuePair<TOrigKey, TValue>)

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>._OnSourceAdd(in Dictionary<TSelectKey, TValue>, KeyValuePair<TOrigKey, TValue>)

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>.GetEnumerator()

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>.Count

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>.ContainsKey(TSelectKey)

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>.TryGetValue(TSelectKey, out TValue)

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>.this[TSelectKey]

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>.Keys

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>.Values

RtReadOnlyDictionary<TSelectKey, TValue>.Count

RtReadOnlyDictionary<TSelectKey, TValue>.ContainsKey(TSelectKey)

RtReadOnlyDictionary<TSelectKey, TValue>.TryGetValue(TSelectKey, out TValue)

RtReadOnlyDictionary<TSelectKey, TValue>.this[TSelectKey]

RtReadOnlyDictionary<TSelectKey, TValue>.Keys

RtReadOnlyDictionary<TSelectKey, TValue>.Values

RtReadOnlyDictionary<TSelectKey, TValue>.RtPairs

RtReadOnlyDictionary<TSelectKey, TValue>.RtKeys

RtReadOnlyDictionary<TSelectKey, TValue>.RtValues

RtReadOnlyDictionary<TSelectKey, TValue>.GetEnumerator()
RtCollection<KeyValuePair<TSelectKey, TValue>>.CollectionEvents
RtCollection<KeyValuePair<TSelectKey, TValue>>.Events
RtCollection<KeyValuePair<TSelectKey, TValue>>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class RtSelectKeyDictionary<TOrigKey, TSelectKey, TValue> : RtMutatedDictionaryBase<TOr
```

Type Parameters

NAME	DESCRIPTION
TOrigKey	The type of the original keys in the source dictionary.
TSelectKey	The type of the selected keys in the mutated dictionary.
TValue	The type of values in the dictionary.

Constructors

RtSelectKeyDictionary(RtReadOnlyDictionary<TOrigKey, TValue>, Func<TOrigKey, TSelectKey>)

Initializes a new instance of the `RtSelectKeyDictionary<TOrigKey, TValue>` class.

Declaration

```
public RtSelectKeyDictionary(RtReadOnlyDictionary<TOrigKey, TValue> source, Func<TOrigKey, TSe
```

Parameters

TYPE	NAME	DESCRIPTION
<code>RtReadOnlyDictionary<TOrigKey, TValue></code>	<code>source</code>	The source read-only dictionary.
<code>Func<TOrigKey, TSelectKey></code>	<code>selectFn</code>	The function to select keys.

Methods

_OnSourceAdd(in Dictionary<TSelectKey, TValue>, KeyValuePair<TOrigKey, TValue>)

Declaration

```
protected override void _OnSourceAdd(in Dictionary<TSelectKey, TValue> mutated, KeyValuePair<T
```

Parameters

TYPE	NAME
Dictionary<TSelectKey, TValue>	mutated
KeyValuePair<TOrigKey, TValue>	kvp

Overrides

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>._OnSourceAdd(in Dictionary<TSelectKey, TValue>, KeyValuePair<TOrigKey, TValue>)

_OnSourceRemove(in Dictionary<TSelectKey, TValue>, KeyValuePair<TOrigKey, TValue>)

Declaration

```
protected override void _OnSourceRemove(in Dictionary<TSelectKey, TValue> mutated, KeyValuePair<T
```

Parameters

TYPE	NAME
Dictionary<TSelectKey, TValue>	mutated
KeyValuePair<TOrigKey, TValue>	kvp

Overrides

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>._OnSourceRemove(in Dictionary<TSelectKey, TValue>, KeyValuePair<TOrigKey, TValue>)

_OnSourceUpdate(in Dictionary<TSelectKey, TValue>, KeyValuePair<TOrigKey, TValue>, KeyValuePair<TOrigKey, TValue>)

Declaration

```
protected override void _OnSourceUpdate(in Dictionary<TSelectKey, TValue> mutated, KeyValuePair<TOrigKey, TValue> oldPair, KeyValuePair<TOrigKey, TValue> newPair)
```

Parameters

TYPE	NAME
Dictionary<TSelectKey, TValue>	mutated
KeyValuePair<TOrigKey, TValue>	oldPair
KeyValuePair<TOrigKey, TValue>	newPair

Overrides

RtMutatedDictionaryBase<TOrigKey, TValue, TSelectKey, TValue>._OnSourceUpdate(in Dictionary<TSelectKey, TValue>, KeyValuePair<TOrigKey, TValue>, KeyValuePair<TOrigKey, TValue>)

Implements

IDisposable
 IReadOnlyDictionary<TKey, TValue>
 IReadOnlyCollection<T>
 IEnumerable<T>
 IEnumerable

Extension Methods

Linq.selectKey<TOrigKey, TSelectedKey, TValue>(RtReadOnlyDictionary<TOrigKey, TValue>, Func<TOrigKey, TSelectedKey>)
Linq.selectValue<TKey, TValue, TSelectValue>(RtReadOnlyDictionary<TKey, TValue>, Func<TValue, TSelectValue>)
Linq.where<TKey, TValue>(RtReadOnlyDictionary<TKey, TValue>, Func<KeyValuePair<TKey, TValue>, bool>)

Class RtSelectList<TOrig, TNew>

Represents a list that projects each element of a source list into a new form.

Inheritance

object → RtCollection<(int index, TNew item)> → RtReadOnlyList<TNew> → RtMutatedListBase<TOrig, TNew> → RtSelectList<TOrig, TNew>

Implements

IDisposable
IReadOnlyList<TNew>
IReadOnlyCollection<TNew>
IEnumerable<TNew>
IEnumerable

Inherited Members

RtMutatedListBase<TOrig, TNew>._mutated
RtMutatedListBase<TOrig, TNew>.Dispose()
RtMutatedListBase<TOrig, TNew>._OnSourceUpdate(List<TNew>, (int index, TOrig item), (int index, TOrig item))
RtMutatedListBase<TOrig, TNew>._OnSourceRemove(List<TNew>, (int index, TOrig item))
RtMutatedListBase<TOrig, TNew>._OnSourceAdd(List<TNew>, (int index, TOrig item))
RtMutatedListBase<TOrig, TNew>.GetEnumerator()
RtMutatedListBase<TOrig, TNew>.Count
RtMutatedListBase<TOrig, TNew>.this[int]
RtReadOnlyList<TNew>.GetEnumerator()
RtReadOnlyList<TNew>.Count
RtReadOnlyList<TNew>.this[int]
RtCollection<(int index, TNew item)>.CollectionEvents
RtCollection<(int index, TNew item)>.Events
RtCollection<(int index, TNew item)>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class RtSelectList<TOrig, TNew> : RtMutatedListBase<TOrig, TNew>, IDisposable, IReadOnl
```

Type Parameters

NAME	DESCRIPTION
TOrig	The type of elements in the source list.
TNew	The type of elements in the projected list.

Constructors

RtSelectList(RtReadOnlyList<TOrig>, Func<TOrig, TNew>)

Declaration

```
public RtSelectList(RtReadOnlyList<TOrig> source, Func<TOrig, TNew> selectFn)
```

Parameters

TYPE	NAME
RtReadOnlyList<TOrig>	source
Func<TOrig, TNew>	selectFn

Methods

_OnSourceAdd(List<TNew>, (int index, TOrig item))

Declaration

```
protected override void _OnSourceAdd(List<TNew> mutated, (int index, TOrig item) item)
```

Parameters

TYPE	NAME
List<TNew>	mutated
(int index, TOrig item)	item

Overrides

[RtMutatedListBase<TOrig, TNew>._OnSourceAdd\(List<TNew>, \(int index, TOrig item\)\)](#)

_OnSourceRemove(List<TNew>, (int index, TOrig item))

Declaration

```
protected override void _OnSourceRemove(List<TNew> mutated, (int index, TOrig item) item)
```

Parameters

TYPE	NAME
List<TNew>	mutated
(int index, TOrig item)	item

Overrides

[RtMutatedListBase<TOrig, TNew>._OnSourceRemove\(List<TNew>, \(int index, TOrig item\)\)](#)

_OnSourceUpdate(List<TNew>, (int index, TOrig item), (int index, TOrig item))

Declaration

```
protected override void _OnSourceUpdate(List<TNew> mutated, (int index, TOrig item) oldItem, (
```

Parameters

TYPE	NAME
List<TNew>	mutated
(int index, TOrig item)	oldItem
(int index, TOrig item)	newItem

Overrides

[RtMutatedListBase<TOrig, TNew>._OnSourceUpdate\(List<TNew>, \(int index, TOrig item\), \(int index, TOrig item\)\)](#)

Implements

IDisposable
 IReadOnlyList<T>
 IReadOnlyCollection<T>
 IEnumerable<T>
 IEnumerable

Extension Methods

Linq.count<T>(RtReadOnlyList<T>)
Linq.groupBy< TKey, TValue>(RtReadOnlyList< TValue>, Func< TValue, TKey>)
Linq.selectNew< T, TSelect>(RtReadOnlyList< T>, Func< T, TSelect>)
Linq.select< T, TSelect>(RtReadOnlyList< T>, Func< T, TSelect>)

© Bus Fighter. All rights reserved.

Class RtSelectLocalList<T, TSelect>

Inheritance

object → RtCollection<(int index, TSelect item)> → RtReadOnlyList<TSelect> → RtMutatedLocalListBase<T, TSelect> → RtSelectLocalList<T, TSelect>

Implements

IDisposable
IReadOnlyList<TSelect>
IReadOnlyCollection<TSelect>
IEnumerable<TSelect>
IEnumerable

Inherited Members

RtMutatedLocalListBase<T, TSelect>.Dispose()
RtMutatedLocalListBase<T, TSelect>._OnSourceUpdate((int index, T item), (int index, T item))
RtMutatedLocalListBase<T, TSelect>._OnSourceRemove((int index, T item))
RtMutatedLocalListBase<T, TSelect>._OnSourceAdd((int index, T item))
RtCollection<(int index, TSelect item)>.CollectionEvents
RtCollection<(int index, TSelect item)>.Events
RtCollection<(int index, TSelect item)>.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class RtSelectLocalList<T, TSelect> : RtMutatedLocalListBase<T, TSelect>, IDisposable,
```

Type Parameters

NAME

T

NAME

TSelect

Constructors

RtSelectLocalList(RtReadOnlyList<T>, Func<T, TSelect>)

Declaration

```
public RtSelectLocalList(RtReadOnlyList<T> source, Func<T, TSelect> selectFn)
```

Parameters

TYPE	NAME
RtReadOnlyList<T>	source
Func<T, TSelect>	selectFn

Properties

Count

Gets the number of elements in the collection.

Declaration

```
public override int Count { get; }
```

Property Value

TYPE	DESCRIPTION
------	-------------

int The number of elements in the collection.

Overrides

[RtReadOnlyList<TSelect>.Count](#)

this[int]

Gets the element at the specified index in the read-only list.

Declaration

```
public override TSelect this[int index] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
int	index	The zero-based index of the element to get.

Property Value

TYPE	DESCRIPTION
TSelect	The element at the specified index in the read-only list.

Overrides

[RtReadOnlyList<TSelect>.this\[int\]](#)

Methods

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public override IEnumarator<TSelect> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
IEnumarator<TSelect>	An enumerator that can be used to iterate through the collection.

Overrides

[RtReadOnlyList<TSelect>.GetEnumerator\(\)](#)

_OnSourceAdd((int index, T item))

Declaration

```
protected override void _OnSourceAdd(int index, T item) listItem
```

Parameters

TYPE	NAME
(int index, T item)	listItem

Overrides

[RtMutatedLocalListBase<T, TSelect>._OnSourceAdd\(\(int index, T item\)\)](#)

_OnSourceRemove((int index, T item))

Declaration

```
protected override void _OnSourceRemove(int index, T item) listItem
```

Parameters

TYPE	NAME
(int index, T item)	listItem

Overrides

[RtMutatedLocalListBase<T, TSelect>._OnSourceRemove\(\(int index, T item\)\)](#)

_OnSourceUpdate((int index, T item), (int index, T item))

Declaration

```
protected override void _OnSourceUpdate(int index, T item) oldItem, int index, T item) newItem
```

Parameters

TYPE	NAME
(int index, T item)	oldItem
(int index, T item)	newItem

```
RtMutatedLocalListBase<T, TSelect>._OnSourceUpdate((int index, T item), (int index, T item))
```

Implements

IDisposable
IReadOnlyList<T>
IReadOnlyCollection<T>
IEnumerable<T>
IEnumerable

Extension Methods

Linq.count<T>(RtReadOnlyList<T>)
Linq.groupBy< TKey, TValue >(RtReadOnlyList< TValue >, Func< TValue, TKey >)
Linq.selectNew< T, TSelect >(RtReadOnlyList< T >, Func< T, TSelect >)
Linq.select< T, TSelect >(RtReadOnlyList< T >, Func< T, TSelect >)

© Bus Fighter. All rights reserved.

Class RtSelectValueDictionary<TKey, TOrigValue, TValue>

Represents a dictionary that selects values based on a provided function.

Inheritance

object → RtCollection<KeyValuePair<TKey, TValue>> → RtReadOnlyDictionary<TKey, TValue> → RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue> → RtSelectValueDictionary<TKey, TOrigValue, TValue>

Implements

IDisposable

IReadOnlyDictionary<TKey, TValue>

IReadOnlyCollection<KeyValuePair<TKey, TValue>>

IEnumerable<KeyValuePair<TKey, TValue>>

IEnumerable

Inherited Members

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>._mutated

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>.Dispose()

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>._OnSourceUpdate(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TOrigValue>, KeyValuePair<TKey, TOrigValue>)

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>._OnSourceRemove(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TOrigValue>)

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>._OnSourceAdd(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TOrigValue>)

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>.GetEnumerator()

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>.Count

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>.ContainsKey(TKey)

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>.TryGetValue(TKey, out TValue)

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>.this[TKey]

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>.Keys

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>.Values

RtReadOnlyDictionary<TKey, TValue>.Count

RtReadOnlyDictionary<TKey, TValue>.ContainsKey(TKey)

RtReadOnlyDictionary<TKey, TValue>.TryGetValue(TKey, out TValue)

RtReadOnlyDictionary<TKey, TValue>.this[TKey]

RtReadOnlyDictionary<TKey, TValue>.Keys

RtReadOnlyDictionary<TKey, TValue>.Values

RtReadOnlyDictionary<TKey, TValue>.RtPairs

RtReadOnlyDictionary<TKey, TValue>.RtKeys

RtReadOnlyDictionary<TKey, TValue>.RtValues

RtReadOnlyDictionary< TKey, TValue >.GetEnumerator()
RtCollection< KeyValuePair< TKey, TValue > >.CollectionEvents
RtCollection< KeyValuePair< TKey, TValue > >.Events
RtCollection< KeyValuePair< TKey, TValue > >.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class RtSelectValueDictionary< TKey, TOrigValue, TValue > : RtMutatedDictionaryBase< TKey,
```

Type Parameters

NAME	DESCRIPTION
TKey	The type of keys in the dictionary.
TOrigValue	The type of the original values in the source dictionary.
TValue	The type of the selected values in the mutated dictionary.

Constructors

RtSelectValueDictionary(RtReadOnlyDictionary< TKey, TOrigValue >, Func< TOrigValue, TValue >)

Initializes a new instance of the `RtSelectValueDictionary< TKey, TOrigValue, TValue >` class.

Declaration

```
public RtSelectValueDictionary(RtReadOnlyDictionary< TKey, TOrigValue > source, Func< TOrigValue,
```

Parameters

TYPE	NAME	DESCRIPTION
<code>RtReadOnlyDictionary< TKey, TOrigValue ></code>	<code>source</code>	The source read-only dictionary.
<code>Func< TOrigValue, TValue ></code>	<code>selectFn</code>	The function to select values.

Methods

_OnSourceAdd(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TOrigValue>)

Declaration

```
protected override void _OnSourceAdd(in Dictionary<TKey, TValue> mutated, KeyValuePair<TKey, T
```

Parameters

TYPE	NAME
Dictionary<TKey, TValue>	mutated
KeyValuePair<TKey, TOrigValue>	kvp

Overrides

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>._OnSourceAdd(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TOrigValue>)

_OnSourceRemove(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TOrigValue>)

Declaration

```
protected override void _OnSourceRemove(in Dictionary<TKey, TValue> mutated, KeyValuePair<TKey
```

Parameters

TYPE	NAME
Dictionary<TKey, TValue>	mutated
KeyValuePair<TKey, TOrigValue>	kvp

Overrides

RtMutatedDictionaryBase<TKey, TOrigValue, TKey, TValue>._OnSourceRemove(in Dictionary<TKey, TValue>, KeyValuePair<TKey, TOrigValue>)

_OnSourceUpdate(in Dictionary< TKey, TValue>, KeyValuePair< TKey, TOrigValue>, KeyValuePair< TKey, TOrigValue>)

Declaration

```
protected override void _OnSourceUpdate(in Dictionary< TKey, TValue> mutated, KeyValuePair< TKey, TOrigValue> oldPair, KeyValuePair< TKey, TOrigValue> newPair)
```

Parameters

TYPE	NAME
Dictionary< TKey, TValue>	mutated
KeyValuePair< TKey, TOrigValue>	oldPair
KeyValuePair< TKey, TOrigValue>	newPair

Overrides

RtMutatedDictionaryBase< TKey, TOrigValue, TKey, TValue>._OnSourceUpdate(in Dictionary< TKey, TValue>, KeyValuePair< TKey, TOrigValue>, KeyValuePair< TKey, TOrigValue>)

Implements

IDisposable
IReadOnlyDictionary< TKey, TValue>
IReadOnlyCollection< T>
IEnumerable< T>
IEnumerable

Extension Methods

Linq.selectKey< TOrigKey, TSelectedKey, TValue>(RtReadOnlyDictionary< TOrigKey, TValue>, Func< TOrigKey, TSelectedKey>)
Linq.selectValue< TKey, TValue, TSelectValue>(RtReadOnlyDictionary< TKey, TValue>, Func< TValue, TSelectValue>)
Linq.where< TKey, TValue>(RtReadOnlyDictionary< TKey, TValue>, Func< KeyValuePair< TKey, TValue>, bool>)

Class Rt<T>

Inheritance

object → RtCollection<(int index, T item)> → RtReadOnlyList<T> → Rt<T>

Implements

IDisposable

IReadOnlyList<T>

IReadOnlyCollection<T>

IEnumerable<T>

IEnumerable

Inherited Members

RtCollection<(int index, T item)>.CollectionEvents

RtCollection<(int index, T item)>.Events

RtCollection<(int index, T item)>.Dispose()

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class Rt<T> : RtReadOnlyList<T>, IDisposable, IReadOnlyList<T>, IReadOnlyCollection<T>,
```

Type Parameters

NAME

T

Constructors

Rt()

Declaration

```
public Rt()
```

Rt(T)

Declaration

```
public Rt(T defaultValue)
```

Parameters

TYPE	NAME
T	defaultValue

Properties

Count

Gets the number of elements in the collection.

Declaration

```
public override int Count { get; }
```

Property Value

TYPE	DESCRIPTION
int	The number of elements in the collection.

Overrides

[RtReadOnlyList<T>.Count](#)

this[int]

Gets the element at the specified index in the read-only list.

Declaration

```
public override T this[int index] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
int	index	The zero-based index of the element to get.

Property Value

TYPE	DESCRIPTION
T	The element at the specified index in the read-only list.

Overrides

[RtReadOnlyList<T>.this\[int\]](#)

Value

Declaration

```
public T Value { get; }
```

Property Value

TYPE
T

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public override void Dispose()
```

Overrides

[RtCollection<\(int index, T item\)>.Dispose\(\)](#)

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public override IEnumerator<T> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
------	-------------

IEnumerator<T>	An enumerator that can be used to iterate through the collection.
----------------	---

Overrides

[RtReadOnlyList<T>.GetEnumerator\(\)](#)

Set(T)

Declaration

```
public void Set(T value)
```

Parameters

TYPE	NAME
------	------

T	value
---	-------

SetNoTrigger(T)

Declaration

```
public void SetNoTrigger(T value)
```

Parameters

TYPE NAME

T value

Operators

implicit operator T(Rt<T>)

Declaration

```
public static implicit operator T(Rt<T> rt)
```

Parameters

TYPE NAME

Rt<T> rt

Returns

TYPE

T

Implements

IDisposable

IReadOnlyList<T>

IReadOnlyCollection<T>

IEnumerable<T>

IEnumerable

Extension Methods

Linq.count<T>(RtReadOnlyList<T>)

Linq.groupBy< TKey, TValue >(RtReadOnlyList< TValue >, Func< TValue, TKey >)

Linq.selectNew<T, TSelect>(RtReadOnlyList<T>, Func<T, TSelect>)

Linq.select<T, TSelect>(RtReadOnlyList<T>, Func<T, TSelect>)

Class Subscription

Inheritance

object → Subscription → [SubscriptionBinding<TDelegate>](#) → [SubscriptionGroup](#)

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class Subscription
```

Methods

Unsubscribe()

Declaration

```
public abstract void Unsubscribe()
```

Extension Methods

[SubscriptionHandleExtension.UnsubscribefNotNull\(Subscription\)](#)

Class SubscriptionBinding<TDelegate>

Inheritance

object → [Subscription](#) → SubscriptionBinding<TDelegate>

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Rx](#)

Assembly: CatSweeper.dll

Syntax

```
public class SubscriptionBinding<TDelegate> : Subscription where TDelegate : class
```

Type Parameters

NAME

TDelegate

Constructors

SubscriptionBinding(TDelegate, RelayBase<TDelegate>)

Declaration

```
public SubscriptionBinding(TDelegate listener, RelayBase<TDelegate> relay)
```

Parameters

TYPE	NAME
TDelegate	listener
RelayBase<TDelegate>	relay

Fields

Listener

Declaration

```
public readonly TDelegate Listener
```

Field Value

TYPE

TDelegate

Methods

IsListener(TDelegate)

Declaration

```
public bool IsListener(TDelegate d)
```

Parameters

TYPE	NAME
TDelegate	d

Returns

TYPE
bool

Declaration

```
public override void Unsubscribe()
```

Overrides

[Subscription.Unsubscribe\(\)](#)

Extension Methods

[SubscriptionHandleExtension.UnsubscribeIfNotNull\(Subscription\)](#)

© Bus Fighter. All rights reserved.

Class SubscriptionGroup

Inheritance

`object` → `Subscription` → `SubscriptionGroup`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfEngine.Rx`

Assembly: `CatSweeper.dll`

Syntax

```
public class SubscriptionGroup : Subscription
```

Methods

Add(Subscription)

Declaration

```
public void Add(Subscription subscription)
```

Parameters

TYPE	NAME
<code>Subscription</code>	<code>subscription</code>

Declaration

```
public override void Unsubscribe()
```

Overrides

[Subscription.Unsubscribe\(\)](#)

Extension Methods

[SubscriptionHandleExtension.UnsubscribeIfNotNull\(Subscription\)](#)

© Bus Fighter. All rights reserved.

Class SubscriptionHandleExtension

Inheritance

`object` → `SubscriptionHandleExtension`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfEngine.Rx`

Assembly: `CatSweeper.dll`

Syntax

```
public static class SubscriptionHandleExtension
```

Methods

UnsubscribeIfNotNull(Subscription)

Declaration

```
public static void UnsubscribeIfNotNull(this Subscription binding)
```

Parameters

TYPE	NAME
<code>Subscription</code>	<code>binding</code>

Namespace cfEngine.Rx.Test

Classes

[RtDictionary_Test](#)

[RtList_Test](#)

© Bus Fighter. All rights reserved.

Class RtDictionary_Test

Inheritance

`object` → RtDictionary_Test

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfEngine.Rx.Test`

Assembly: CatSweeper.dll

Syntax

```
[TestFixture]
public class RtDictionary_Test
```

Methods

RtDictionary_AddRemove()

Declaration

```
[Test]
public void RtDictionary_AddRemove()
```

RtDictionary_Dispose()

Declaration

```
[Test]  
public void RtDictionary_Dispose()
```

RtDictionary_NotCacheSubscription()

Declaration

```
[Test]  
public void RtDictionary_NotCacheSubscription()
```

RtDictionary_Remove()

Declaration

```
[Test]  
public void RtDictionary_Remove()
```

RtDictionary_Upsert()

Declaration

```
[Test]  
public void RtDictionary_Upsert()
```

RtReadOnlyDictionary_RtKeys()

Declaration

```
[Test]  
public void RtReadOnlyDictionary_RtKeys()
```

RtReadOnlyDictionary_RtPairs()

Declaration

```
[Test]  
public void RtReadOnlyDictionary_RtPairs()
```

RtReadOnlyDictionary_RtValues()

Declaration

```
[Test]  
public void RtReadOnlyDictionary_RtValues()
```

© Bus Fighter. All rights reserved.

Class RtList_Test

Inheritance

`object` → RtList_Test

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfEngine.Rx.Test`

Assembly: CatSweeper.dll

Syntax

```
[TestFixture]
public class RtList_Test
```

Methods

RtList_Add()

Declaration

```
[Test]
public void RtList_Add()
```

RtList_Dispose()

Declaration

```
[Test]
```

```
public void RtList_Dispose()
```

© Bus Fighter. All rights reserved.

Namespace cfEngine.Serialize

Classes

[JsonSerializer](#)

[Serializer](#)

Interfaces

[ISerializer](#)

[ISerializer.IDeserializeParam](#)

[ISerializer.ISerializeParam](#)

© Bus Fighter. All rights reserved.

Interface ISerializer

Namespace: [cfEngine.Serialize](#)

Assembly: CatSweeper.dll

Syntax

```
public interface ISerializer
```

Methods

Deserialize(byte[], IDeserializeParam)

Declaration

```
object Deserialize(byte[] byteLoaded, ISerializer.IDeserializeParam param = null)
```

Parameters

TYPE	NAME
byte[]	byteLoaded
ISerializer.IDeserializeParam	param

Returns

TYPE
object

DeserializeAsAsync<T>(byte[], IDeserializeParam, CancellationToken)

Declaration

```
Task<T> DeserializeAsAsync<T>(byte[] byteLoaded, ISerializer.IDeserializeParam param = null, C
```

Parameters

TYPE	NAME
byte[]	byteLoaded
ISerializer.IDeserializeParam	param
CancellationToken	token

Returns

TYPE
Task<T>

Type Parameters

NAME
T

DeserializeAs<T>(byte[], IDeserializeParam)

Declaration

```
T DeserializeAs<T>(byte[] byteLoaded, ISerializer.IDeserializeParam param = null)
```

Parameters

TYPE	NAME
byte[]	byteLoaded
ISerializer.IDeserializeParam	param

Returns

TYPE
T

Type Parameters

NAME
T

DeserializeAsync(byte[], IDeserializeParam, CancellationToken)

Declaration

```
Task<object> DeserializeAsync(byte[] byteLoaded, ISerializer.IDeserializeParam param = null, C
```

Parameters

TYPE	NAME
byte[]	byteLoaded
ISerializer.IDeserializeParam	param
CancellationToken	token

Returns

TYPE

```
Task<object>
```

Serialize(object, ISerializeParam)

Declaration

```
string Serialize(object obj, ISerializer.ISerializeParam param = null)
```

Parameters

TYPE	NAME
object	obj
ISerializer.ISerializeParam	param

Returns

TYPE

```
string
```

SerializeAsync(object, ISerializeParam, CancellationToken)

Declaration

```
Task<string> SerializeAsync(object obj, ISerializer.ISerializeParam param = null, CancellationToken
```

Parameters

TYPE	NAME
object	obj
ISerializer.ISerializeParam	param
CancellationToken	token

Returns

TYPE
Task<string>

© Bus Fighter. All rights reserved.

Interface ISerializer.IDeserializeParam

Namespace: [cfEngine.Serialize](#)

Assembly: CatSweeper.dll

Syntax

```
public interface ISerializer.IDeserializeParam
```

© Bus Fighter. All rights reserved.

Interface ISerializer.ISerializeParam

Namespace: [cfEngine.Serialize](#)

Assembly: CatSweeper.dll

Syntax

```
public interface ISerializer.ISerializeParam
```

© Bus Fighter. All rights reserved.

Class JsonSerializer

Inheritance

object → [Serializer](#) → JsonSerializer

Implements

[ISerializer](#)

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Serialize](#)

Assembly: CatSweeper.dll

Syntax

```
public class JsonSerializer : Serializer, ISerializer
```

Fields

OPTIONS

Declaration

```
public JsonSerializerOptions OPTIONS
```

Field Value

[TYPE](#)

[JsonSerializerOptions](#)

Properties

Instance

Declaration

```
public static JsonSerializer Instance { get; }
```

Property Value

TYPE

JsonSerializer

Methods

Deserialize(byte[], IDeserializeParam)

Declaration

```
public override object Deserialize(byte[] byteLoaded, ISerializer.IDeserializeParam param = nu
```

Parameters

TYPE	NAME
byte[]	byteLoaded
ISerializer.IDeserializeParam	param

Returns

TYPE
object

Overrides

[Serializer.Deserialize\(byte\[\], ISerializer.IDeserializeParam\)](#)

DeserializeAsAsync<T>(byte[], IDeserializeParam, CancellationToken)

Declaration

```
public override Task<T> DeserializeAsAsync<T>(byte[] byteLoaded, ISerializer.IDeserializeParam
```

Parameters

TYPE	NAME
byte[]	byteLoaded
ISerializer.IDeserializeParam	deserializeParam
CancellationToken	token

Returns

TYPE
Task<T>

Type Parameters

NAME
T

Overrides

[Serializer.DeserializeAsAsync<T>\(byte\[\], ISerializer.IDeserializeParam, CancellationToken\)](#)

DeserializeAs<T>(byte[], IDeserializeParam)

Declaration

```
public override T DeserializeAs<T>(byte[] byteLoaded, ISerializer.IDeserializeParam param = nu
```

Parameters

TYPE	NAME
byte[]	byteLoaded
ISerializer.IDeserializeParam	param

Returns

TYPE
T

Type Parameters

NAME

T

Overrides

[Serializer.DeserializeAs<T>\(byte\[\], ISerializer.IDeserializeParam\)](#)

DeserializeAsync(byte[], IDeserializeParam, CancellationToken)

Declaration

```
public override Task<object> DeserializeAsync(byte[] byteLoaded, ISerializer.IDeserializeParam
```

Parameters

TYPE	NAME
byte[]	byteLoaded
ISerializer.IDeserializeParam	deserializeParam
CancellationToken	token

Returns

TYPE

[Task<object>](#)

Overrides

[Serializer.DeserializeAsync\(byte\[\], ISerializer.IDeserializeParam, CancellationToken\)](#)

Serialize(object, ISerializeParam)

Declaration

```
public override string Serialize(object obj, ISerializer.ISerializeParam param = null)
```

Parameters

TYPE	NAME
object	obj
ISerializer.ISerializeParam	param

Returns

TYPE

string

Overrides

[Serializer.Serialize\(object, ISerializer.ISerializeParam\)](#)

SerializeAsync(object, ISerializeParam, CancellationToken)

Declaration

```
public override Task<string> SerializeAsync(object obj, ISerializer.ISerializeParam param = nu
```

Parameters

TYPE	NAME
object	obj
ISerializer.ISerializeParam	param
CancellationToken	token

Returns

TYPE

Task<string>

Overrides

[Serializer.SerializeAsync\(object, ISerializer.ISerializeParam, CancellationToken\)](#)

Implements

[ISerializer](#)

© Bus Fighter. All rights reserved.

Class Serializer

Inheritance

object → Serializer → [JsonSerializer](#)

Implements

[ISerializer](#)

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Serialize](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class Serializer : ISerializer
```

Methods

Deserialize(byte[], IDeserializeParam)

Declaration

```
public abstract object Deserialize(byte[] byteLoaded, ISerializer.IDeserializeParam param = nu
```

Parameters

TYPE	NAME
byte[]	byteLoaded

TYPE**NAME**

ISerializer.IDeserializeParam	param
-------------------------------	-------

Returns**TYPE**

object

DeserializeAsAsync<T>(byte[], IDeserializeParam, CancellationToken)

Declaration

```
public abstract Task<T> DeserializeAsAsync<T>(byte[] byteLoaded, ISerializer.IDeserializeParam
```

Parameters**TYPE****NAME**

byte[]	byteLoaded
ISerializer.IDeserializeParam	param
CancellationToken	token

Returns**TYPE**

Task<T>

Type Parameters**NAME**

T

DeserializeAs<T>(byte[], IDeserializeParam)

Declaration

```
public abstract T DeserializeAs<T>(byte[] byteLoaded, ISerializer.IDeserializeParam param = nu
```

Parameters

TYPE**NAME**

byte[]	byteLoaded
ISerializer.IDeserializeParam	param

Returns**TYPE**

T

Type Parameters**NAME**

T

DeserializeAsync(byte[], IDeserializeParam, CancellationToken)

Declaration

```
public abstract Task<object> DeserializeAsync(byte[] byteLoaded, ISerializer.IDeserializeParam
```

Parameters**TYPE****NAME**

byte[]	byteLoaded
ISerializer.IDeserializeParam	param
CancellationToken	token

Returns**TYPE**

Task<object>

Serialize(object, ISerializeParam)

Declaration

```
public abstract string Serialize(object obj, ISerializer.ISerializeParam param = null)
```

Parameters

TYPE	NAME
object	obj
ISerializer.ISerializeParam	param

Returns

TYPE
string

SerializeAsync(object, ISerializeParam, CancellationToken)

Declaration

```
public abstract Task<string> SerializeAsync(object obj, ISerializer.ISerializeParam param = nu
```

Parameters

TYPE	NAME
object	obj
ISerializer.ISerializeParam	param
CancellationToken	token

Returns

TYPE
Task<string>

Implements

ISerializer

© Bus Fighter. All rights reserved.

Namespace cfEngine.Service

Classes

[InventoryInfo](#)

[InventoryInfoManager](#)

[ServiceLocator](#)

Interfaces

[IModelService](#)

[IService](#)

[IServiceLocator](#)

[IServiceModel](#)

© Bus Fighter. All rights reserved.

Interface IMModelService

Inherited Members

[IDisposable.Dispose\(\)](#)

Namespace: [cfEngine.Service](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IMModelService : IService, IDisposable
```

Properties

GetModel

Declaration

```
IServiceModel GetModel { get; }
```

Property Value

TYPE

[IServiceModel](#)

Interface IService

Inherited Members

[IDisposable.Dispose\(\)](#)

Namespace: `cfEngine.Service`

Assembly: CatSweeper.dll

Syntax

```
public interface IService : IDisposable
```

© Bus Fighter. All rights reserved.

Interface IServiceLocator

Inherited Members

[IEnumerable<IService>.GetEnumerator\(\)](#)

[IDisposable.Dispose\(\)](#)

Namespace: [cfEngine.Service](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IServiceLocator : IEnumerable<IService>, IEnumerable, IDisposable
```

Methods

GetService<T>()

Declaration

```
T GetService<T>() where T : IService
```

Returns

TYPE

T

Type Parameters

NAME

T

GetService<T>(string)

Declaration

```
T GetService<T>(string serviceName) where T : IService
```

Parameters

TYPE	NAME
string	serviceName

Returns

TYPE
T

Type Parameters

NAME
T

Register<T>(T, string)

Declaration

```
void Register<T>(T service, string serviceName) where T : IService
```

Parameters

TYPE	NAME
T	service
string	serviceName

Type Parameters

NAME
T

Unregister(string)

Declaration

```
void Unregister(string serviceName)
```

Parameters

TYPE	NAME
------	------

string	serviceName
--------	-------------

Unregister<T>(T)

Declaration

```
void Unregister<T>(T service) where T : IService
```

Parameters

TYPE	NAME
------	------

T	service
---	---------

Type Parameters

NAME

T

© Bus Fighter. All rights reserved.

Interface IServiceModel

Inherited Members

[IRuntimeSavable.Initialize\(IUserData\)](#)
[IRuntimeSavable.SetSaveData\(Dictionary<string, object>\)](#)
[IDisposable.Dispose\(\)](#)

Namespace: [cfEngine.Service](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IServiceModel : IRuntimeSavable, IDisposable
```

© Bus Fighter. All rights reserved.

Class InventoryInfo

Inheritance

[object](#) → [InventoryInfo](#)

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Service](#)

Assembly: CatSweeper.dll

Syntax

```
public class InventoryInfo
```

Properties

iconKey

Declaration

```
public string iconKey { get; set; }
```

Property Value

TYPE

[string](#)

itemId

Declaration

```
public string itemId { get; set; }
```

Property Value

TYPE

string

maxStackSize

Declaration

```
public int maxStackSize { get; set; }
```

Property Value

TYPE

int

© Bus Fighter. All rights reserved.

Class InventoryInfoManager

Inheritance

object → [InfoManager](#) → [ConfigInfoManager<string, InventoryInfo>](#) → [InventoryInfoManager](#)

Implements

[IInfoManager](#)

[IDisposable](#)

Inherited Members

[ConfigInfoManager<string, InventoryInfo>.valueMap](#)

[ConfigInfoManager<string, InventoryInfo>.GetAllValue\(\)](#)

[ConfigInfoManager<string, InventoryInfo>.keyFn](#)

[ConfigInfoManager<string, InventoryInfo>.infoType](#)

[ConfigInfoManager<string, InventoryInfo>.LoadInfo\(\)](#)

[ConfigInfoManager<string, InventoryInfo>.LoadInfoAsync\(CancellationToken\)](#)

[ConfigInfoManager<string, InventoryInfo>.AddValue\(InventoryInfo\)](#)

[ConfigInfoManager<string, InventoryInfo>.TryGetValue\(string, out InventoryInfo\)](#)

[ConfigInfoManager<string, InventoryInfo>.Dispose\(\)](#)

[InfoManager.infoType](#)

[InfoManager.GetAllValue\(\)](#)

[InfoManager.LoadInfo\(\)](#)

[InfoManager.LoadInfoAsync\(CancellationToken\)](#)

[InfoManager.OnLoadCompleted\(\)](#)

[InfoManager.Dispose\(\)](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfEngine.Service](#)

Assembly: CatSweeper.dll

Syntax

```
public class InventoryInfoManager : ConfigInfoManager<string, InventoryInfo>, IInfoManager, ID
```

Constructors

InventoryInfoManager(IValueLoader<InventoryInfo>)

Declaration

```
public InventoryInfoManager(IValueLoader<InventoryInfo> loader)
```

Parameters

TYPE	NAME
IValueLoader<InventoryInfo>	loader

Properties

keyFn

Declaration

```
protected override Func<InventoryInfo, string> keyFn { get; }
```

Property Value

TYPE
Func<InventoryInfo, string>

Overrides

[ConfigInfoManager<string, InventoryInfo>.keyFn](#)

Methods

GetOrDefault(string)

Declaration

```
public InventoryInfo GetOrDefault(string itemId)
```

Parameters

TYPE	NAME
------	------

string	itemId
--------	--------

Returns

TYPE

InventoryInfo

Implements

IInfoManager

IDisposable

© Bus Fighter. All rights reserved.

Class ServiceLocator

Inheritance

object → ServiceLocator → Game

Implements

IServiceLocator
IEnumerable<IService>
IEnumerable
IDisposable

Inherited Members

object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfEngine.Service](#)

Assembly: CatSweeper.dll

Syntax

```
public class ServiceLocator : IServiceLocator, IEnumerable<IService>, IEnumerable, IDisposable
```

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

GetEnumerator()

Returns an enumerator that iterates through the collection.

Declaration

```
public IEnumarator<IService> GetEnumerator()
```

Returns

TYPE	DESCRIPTION
IEnumarator<IService>	An enumerator that can be used to iterate through the collection.

GetService<T>()

Declaration

```
public T GetService<T>() where T : IService
```

Returns

TYPE
T

Type Parameters

NAME
T

GetService<T>(string)

Declaration

```
public T GetService<T>(string serviceName) where T : IService
```

Parameters

TYPE	NAME
string	serviceName

Returns

TYPE

T

Type Parameters

NAME

T

Register<T>(T, string)

Declaration

```
public void Register<T>(T service, string serviceName) where T : IService
```

Parameters

TYPE NAME

T service

string serviceName

Type Parameters

NAME

T

Unregister(string)

Declaration

```
public void Unregister(string serviceName)
```

Parameters

TYPE NAME

string serviceName

Unregister<T>(T)

Declaration

```
public void Unregister<T>(T service) where T : IService
```

Parameters

TYPE	NAME
------	------

T	service
---	---------

Type Parameters

NAME

T

Implements

IServiceLocator

IEnumerable<T>

IEnumerable

IDisposable

© Bus Fighter. All rights reserved.

Namespace cfEngine.Service.Auth

Classes

[AuthService](#)

[AuthService.Builder](#)

[LocalAuthService](#)

[LocalPlatform](#)

[LoginToken](#)

[PlatformAuth](#)

Interfaces

[IAuthService](#)

Enums

[LoginPlatform](#)

© Bus Fighter. All rights reserved.

Class AuthService

Inheritance

object → AuthService → LocalAuthService

Implements

IAuthService

IService

IDisposable

Inherited Members

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: [cfEngine.Service.Auth](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class AuthService : IAuthService, IService, IDisposable
```

Properties

PlatformAuths

Declaration

```
public IReadOnlyDictionary<LoginPlatform, PlatformAuth> PlatformAuths { get; }
```

Property Value

TYPE

IReadOnlyDictionary<LoginPlatform, PlatformAuth>

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public virtual void Dispose()
```

GetUserId()

Declaration

```
public abstract string GetUserId()
```

Returns

TYPE

string

InitAsync(CancellationToken)

Declaration

```
public abstract Task InitAsync(CancellationToken token)
```

Parameters

TYPE	NAME
CancellationToken	token

Returns

TYPE**Task**

IsSessionUserExist()

Declaration

```
public bool IsSessionUserExist()
```

Returns

TYPE**bool**

LinkAsync(LoginPlatform, LoginToken)

Declaration

```
public Task LinkAsync(LoginPlatform platform, LoginToken token)
```

Parameters

TYPE **NAME**

LoginPlatform platform

LoginToken token

Returns

TYPE**Task**

RegisterPlatform(PlatformAuth)

Declaration

```
public void RegisterPlatform(PlatformAuth platform)
```

Parameters

TYPE	NAME
PlatformAuth	platform

SignInAsync(LoginPlatform, LoginToken)

Declaration

```
public Task SignInAsync(LoginPlatform platform, LoginToken token)
```

Parameters

TYPE	NAME
LoginPlatform	platform
LoginToken	token

Returns

TYPE
Task

SignUpAsync(LoginPlatform, LoginToken)

Declaration

```
public Task SignUpAsync(LoginPlatform platform, LoginToken token)
```

Parameters

TYPE	NAME
LoginPlatform	platform
LoginToken	token

Returns

TYPE
Task

TryLoginCachedUserAsync(CancellationToken)

Declaration

```
public abstract Task<bool> TryLoginCachedUserAsync(CancellationToken token)
```

Parameters

TYPE	NAME
CancellationToken	token

Returns

TYPE
Task<bool>

Implements

[IAuthService](#)
[IService](#)
[IDisposable](#)

© Bus Fighter. All rights reserved.

Class AuthService.Builder

Inheritance

[object](#) → AuthService.Builder

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Service.Auth](#)

Assembly: CatSweeper.dll

Syntax

```
public class AuthService.Builder
```

Methods

Build()

Declaration

```
public IAuthService Build()
```

Returns

TYPE

[IAuthService](#)

RegisterPlatform(PlatformAuth)

Declaration

```
public AuthService.Builder RegisterPlatform(PlatformAuth platform)
```

Parameters

TYPE	NAME
PlatformAuth	platform

Returns

TYPE
AuthService.Builder

SetService(IAuthService)

Declaration

```
public AuthService.Builder SetService(IAuthService authService)
```

Parameters

TYPE	NAME
IAuthService	authService

Returns

TYPE
AuthService.Builder

© Bus Fighter. All rights reserved.

Interface IAuthService

Inherited Members

[IDisposable.Dispose\(\)](#)

Namespace: [cfEngine.Service.Auth](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IAuthService : IService, IDisposable
```

Methods

GetUserId()

Declaration

```
string GetUserId()
```

Returns

TYPE

string

InitAsync(CancellationToken)

Declaration

```
Task InitAsync(CancellationToken token)
```

Parameters

TYPE NAME

CancellationToken token

Returns

TYPE

Task

IsSessionUserExist()

Declaration

```
bool IsSessionUserExist()
```

Returns

TYPE

bool

LinkAsync(LoginPlatform, LoginToken)

Declaration

```
Task LinkAsync(LoginPlatform platform, LoginToken token)
```

Parameters

TYPE NAME

LoginPlatform platform

LoginToken token

Returns

TYPE

Task

RegisterPlatform(PlatformAuth)

Declaration

```
void RegisterPlatform(PlatformAuth platform)
```

Parameters

TYPE	NAME
PlatformAuth	platform

SignInAsync(LoginPlatform, LoginToken)

Declaration

```
Task SignInAsync(LoginPlatform platform, LoginToken token)
```

Parameters

TYPE	NAME
LoginPlatform	platform
LoginToken	token

Returns

TYPE
Task

SignUpAsync(LoginPlatform, LoginToken)

Declaration

```
Task SignUpAsync(LoginPlatform platform, LoginToken token)
```

Parameters

TYPE	NAME
LoginPlatform	platform
LoginToken	token

Returns

TryLoginCachedUserAsync(CancellationToken)

Declaration

```
Task<bool> TryLoginCachedUserAsync(CancellationToken token)
```

Parameters

TYPE	NAME
CancellationToken	token

Returns

TYPE
Task<bool>

© Bus Fighter. All rights reserved.

Class LocalAuthService

Inheritance

object → AuthService → LocalAuthService

Implements

IAuthService
IService
IDisposable

Inherited Members

AuthService.PlatformAuths
AuthService.IsSessionUserExist()
AuthService.RegisterPlatform(PlatformAuth)
AuthService.SignInAsync(LoginPlatform, LoginToken)
AuthService.SignUpAsync(LoginPlatform, LoginToken)
AuthService.LinkAsync(LoginPlatform, LoginToken)
AuthService.Dispose()
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfEngine.Service.Auth](#)

Assembly: CatSweeper.dll

Syntax

```
public class LocalAuthService : AuthService, IAuthService, IService, IDisposable
```

Methods

GetUserId()

Declaration

```
public override string GetUserId()
```

Returns

TYPE

string

Overrides

[AuthService.GetUserId\(\)](#)

InitAsync(CancellationToken)

Declaration

```
public override Task InitAsync(CancellationToken token)
```

Parameters

TYPE	NAME
CancellationToken	token

Returns

TYPE

Task

Overrides

[AuthService.InitAsync\(CancellationToken\)](#)

TryLoginCachedUserAsync(CancellationToken)

Declaration

```
public override Task<bool> TryLoginCachedUserAsync(CancellationToken token)
```

Parameters

TYPE	NAME
CancellationToken	token

Returns

TYPE

Task<bool>

Overrides

AuthService.TryLoginCachedUserAsync(CancellationToken)

Implements

IAuthService

IService

IDisposable

© Bus Fighter. All rights reserved.

Class LocalPlatform

Inheritance

object → PlatformAuth → LocalPlatform

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Service.Auth](#)

Assembly: CatSweeper.dll

Syntax

```
public class LocalPlatform : PlatformAuth
```

Properties

Platform

Declaration

```
public override LoginPlatform Platform { get; }
```

Property Value

TYPE

[LoginPlatform](#)

Overrides

[PlatformAuth.Platform](#)

Methods

LinkAsync(LoginToken)

Declaration

```
public override Task LinkAsync(LoginToken token)
```

Parameters

TYPE	NAME
LoginToken	token

Returns

TYPE
Task

Overrides

[PlatformAuth.LinkAsync\(LoginToken\)](#)

SignInAsync(LoginToken)

Declaration

```
public override Task SignInAsync(LoginToken token)
```

Parameters

TYPE	NAME
LoginToken	token

Returns

TYPE
Task

Overrides

[PlatformAuth.SignInAsync\(LoginToken\)](#)

SignUpAsync(LoginToken)

Declaration

```
public override Task SignUpAsync(LoginToken token)
```

Parameters

TYPE	NAME
LoginToken	token

Returns

TYPE
Task

Overrides

[PlatformAuth.SignUpAsync\(LoginToken\)](#)

© Bus Fighter. All rights reserved.

Enum LoginPlatform

Namespace: [cfEngine.Service.Auth](#)

Assembly: CatSweeper.dll

Syntax

```
public enum LoginPlatform : byte
```

Fields

NAME

Anonymous

Apple

AppleGame

Facebook

FromCached

Google

GooglePlay

Local

Oculus

Steam

Username

Extension Methods

[EnumExtension.hasFlag\(Enum, Enum\)](#)

Class LoginToken

Inheritance

[object](#) → LoginToken

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Service.Auth](#)

Assembly: CatSweeper.dll

Syntax

```
public class LoginToken
```

© Bus Fighter. All rights reserved.

Class PlatformAuth

Inheritance

object → PlatformAuth → [LocalPlatform](#)

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Service.Auth](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class PlatformAuth
```

Properties

Platform

Declaration

```
public abstract LoginPlatform Platform { get; }
```

Property Value

TYPE

[LoginPlatform](#)

Methods

LinkAsync(LoginToken)

Declaration

```
public abstract Task LinkAsync(LoginToken token)
```

Parameters

TYPE	NAME
LoginToken	token

Returns

TYPE
Task

SignInAsync(LoginToken)

Declaration

```
public abstract Task SignInAsync(LoginToken token)
```

Parameters

TYPE	NAME
LoginToken	token

Returns

TYPE
Task

SignUpAsync(LoginToken)

Declaration

```
public abstract Task SignUpAsync(LoginToken token)
```

Parameters

TYPE	NAME
LoginToken	token

Returns

TYPE
Task

© Bus Fighter. All rights reserved.

Namespace cfEngine.Service.Inventory

Classes

[InventoryModel](#)

[InventoryService](#)

Interfaces

[IInventoryService](#)

© Bus Fighter. All rights reserved.

Interface IInventoryService

Inherited Members

[IModelService.GetModel](#)

[IDisposable.Dispose\(\)](#)

Namespace: [cfEngine.Service.Inventory](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IInventoryService : IModelService, IService, IDisposable
```

© Bus Fighter. All rights reserved.

Class InventoryModel

Inheritance

`object` → `InventoryModel`

Implements

`IServiceModel`

`IRuntimeSavable`

`IDisposable`

Inherited Members

`object.Equals(object)`

`object.Equals(object, object)`

`object.GetHashCode()`

`object.GetType()`

`object.MemberwiseClone()`

`object.ReferenceEquals(object, object)`

`object.ToString()`

Namespace: `cfEngine.Service.Inventory`

Assembly: `CatSweeper.dll`

Syntax

```
public class InventoryModel : IServiceModel, IRuntimeSavable, IDisposable
```

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

Initialize(IUserData)

Declaration

```
public void Initialize(IUserData userData)
```

Parameters

TYPE	NAME
IUserData	userData

SetSaveData(Dictionary<string, object>)

Declaration

```
public void SetSaveData(Dictionary<string, object> dataMap)
```

Parameters

TYPE	NAME
Dictionary<string, object>	dataMap

Implements

[IServiceModel](#)
[IRuntimeSavable](#)
[IDisposable](#)

© Bus Fighter. All rights reserved.

Class InventoryService

Inheritance

`object` → `InventoryService`

Implements

`IInventoryService`

`IModelService`

`IService`

`IDisposable`

Inherited Members

`object.Equals(object)`

`object.Equals(object, object)`

`object.GetHashCode()`

`object.GetType()`

`object.MemberwiseClone()`

`object.ReferenceEquals(object, object)`

`object.ToString()`

Namespace: `cfEngine.Service.Inventory`

Assembly: `CatSweeper.dll`

Syntax

```
public class InventoryService : IInventoryService, IModelService, IService, IDisposable
```

Constructors

InventoryService(InventoryModel)

Declaration

```
public InventoryService(InventoryModel model)
```

Parameters

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

Implements

IInventoryService

IModelService

IService

IDisposable

© Bus Fighter. All rights reserved.

Namespace cfEngine.Service.Statistic

Classes

[Statistic](#)

[StatisticModel](#)

© Bus Fighter. All rights reserved.

Class Statistic

Inheritance

`object` → `Statistic`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfEngine.Service.Statistic](#)

Assembly: CatSweeper.dll

Syntax

```
public class Statistic
```

Properties

Value

Declaration

```
public double Value { get; }
```

Property Value

TYPE

`double`

Methods

RecordOnce()

Declaration

```
public void RecordOnce()
```

Events

OnUpdate

Declaration

```
public event Action<double> OnUpdate
```

Event Type

TYPE

Action<double>

© Bus Fighter. All rights reserved.

Class StatisticModel

Inheritance

`object` → StatisticModel

Implements

`IServiceModel`

`IRuntimeSavable`

`IDisposable`

Inherited Members

`object.Equals(object)`

`object.Equals(object, object)`

`object.GetHashCode()`

`object.GetType()`

`object.MemberwiseClone()`

`object.ReferenceEquals(object, object)`

`object.ToString()`

Namespace: `cfEngine.Service.Statistic`

Assembly: CatSweeper.dll

Syntax

```
public class StatisticModel : IServiceModel, IRuntimeSavable, IDisposable
```

Properties

StatisticMap

Declaration

```
public Dictionary<string, Statistic> StatisticMap { get; }
```

Property Value

TYPE

Dictionary<string, Statistic>

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

GetMatchedStatistic(string)

Declaration

```
public IEnumerable<KeyValuePair<string, Statistic>> GetMatchedStatistic(string regex)
```

Parameters

TYPE	NAME
------	------

string	regex
--------	-------

Returns

TYPE

IEnumerable<KeyValuePair<string, Statistic>>
--

GetOrCreateStatistic(string)

Declaration

```
public Statistic GetOrCreateStatistic(string statisticKey)
```

Parameters

TYPE**NAME**

`string statisticKey`**Returns****TYPE**

`Statistic`

Initialize(IUserData)

Declaration`public void Initialize(IUserData userData)`**Parameters****TYPE****NAME**

`IUserData userData`

SetSaveData(Dictionary<string, object>)

Declaration`public void SetSaveData(Dictionary<string, object> dataMap)`**Parameters****TYPE****NAME**

`Dictionary<string, object> dataMap`

TryGetStat(string, out Statistic)

Declaration`public bool TryGetStat(string key, out Statistic statistic)`**Parameters**

TYPE	NAME
------	------

string	key
--------	-----

Statistic	statistic
-----------	-----------

Returns

TYPE

bool

Events

OnNewStatisticRecorded

Declaration

```
public event Action<string> OnNewStatisticRecorded
```

Event Type

TYPE

Action<string>

Implements

[IServiceModel](#)

[IRuntimeSavable](#)

[IDisposable](#)

© Bus Fighter. All rights reserved.

Namespace cfEngine.Util

Classes

[DirectoryUtil](#)

[JsonElementExtension](#)

[PathSegmentBuilder](#)

[SanityCheck](#)

[SanityCheckException](#)

[StateExecutionException<TStatId>](#)

[StateMachine<TStatId, TState, TStateMachine>](#)

[StateParam](#)

[State<TStatId, TState, TStateMachine>](#)

[TypeExtension](#)

Structs

[PathSegment](#)

[StateChangeRecord<TStatId>](#)

Interfaces

[IStateMachine<TStatId>](#)

Class DirectoryUtil

Inheritance

[object](#) → [DirectoryUtil](#)

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public static class DirectoryUtil
```

Methods

CreateDirectoryIfNotExists(string, string)

Declaration

```
public static DirectoryInfo CreateDirectoryIfNotExists(string directoryPath, string assetFolde
```

Parameters

TYPE	NAME
string	directoryPath
string	assetFolderPath

Returns

© Bus Fighter. All rights reserved.

Interface IStateMachine<TStatId>

Namespace: [cfEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IStateMachine<TStateId>
```

Type Parameters

NAME

TStateId

Properties

currentStatId

Declaration

```
TStateId currentStateId { get; }
```

Property Value

TYPE

TStatId

lastStatId

Declaration

```
TStateId lastStateId { get; }
```

Property Value

TYPE

TStatelid

Methods

CanGoToState(TStatelid, StateParam)

Declaration

```
bool CanGoToState(TStateId id, StateParam param = null)
```

Parameters

TYPE	NAME
TStatelid	id
StateParam	param

Returns

TYPE
bool

ForceGoToState(TStatelid, StateParam)

Declaration

```
void ForceGoToState(TStateId nextStateId, StateParam param = null)
```

Parameters

TYPE	NAME
TStatelid	nextStateId
StateParam	param

SubscribeAfterStateChange(Action<StateChangeRecord<TStatId>>)

Declaration

```
Subscription SubscribeAfterStateChange(Action<StateChangeRecord<TStateId>> listener)
```

Parameters

TYPE	NAME
Action<StateChangeRecord<TStatId>>	listener

Returns

TYPE
Subscription

SubscribeBeforeStateChange(Action<StateChangeRecord<TStatId>>)

Declaration

```
Subscription SubscribeBeforeStateChange(Action<StateChangeRecord<TStateId>> listener)
```

Parameters

TYPE	NAME
Action<StateChangeRecord<TStatId>>	listener

Returns

TYPE
Subscription

TryGoToState(TStatId, StateParam)

Declaration

```
bool TryGoToState(TStateId nextStateId, StateParam param = null)
```

Parameters

TYPE	NAME
TStateId	nextStateId
StateParam	param

Returns

TYPE
bool

© Bus Fighter. All rights reserved.

Class JsonElementExtension

Inheritance

[object](#) → [JsonElementExtension](#)

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public static class JsonElementExtension
```

Methods

ToObject(JsonElement)

Declaration

```
public static object ToObject(this JsonElement jsonElement)
```

Parameters

TYPE	NAME
JsonElement	jsonElement

Returns

© Bus Fighter. All rights reserved.

Struct PathSegment

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public struct PathSegment
```

Constructors

PathSegment(ReadOnlyMemory<string>)

Declaration

```
public PathSegment(ReadOnlyMemory<string> segments)
```

Parameters

TYPE	NAME
ReadOnlyMemory<string>	segments

Methods

GetOsPath()

Declaration

```
public readonly string GetOsPath()
```

Returns

TYPE

string

GetPath()

Declaration

```
public readonly string GetPath()
```

Returns

TYPE

string

GetSegments()

Declaration

```
public ReadOnlyMemory<string> GetSegments()
```

Returns

TYPE

ReadOnlyMemory<string>

HasValue()

Declaration

```
public readonly bool HasValue()
```

Returns

TYPE

bool

ToString()

Returns the fully qualified type name of this instance.

Declaration

```
public override string ToString()
```

Returns

TYPE DESCRIPTION

string The fully qualified type name.

Overrides

[ValueType.ToString\(\)](#)

© Bus Fighter. All rights reserved.

Class PathSegmentBuilder

Inheritance

`object` → PathSegmentBuilder

Implements

`IDisposable`

Inherited Members

`object.Equals(object)`

`object.Equals(object, object)`

`object.GetHashCode()`

`object.GetType()`

`object.MemberwiseClone()`

`object.ReferenceEquals(object, object)`

`object.ToString()`

Namespace: `cfEngine.Util`

Assembly: CatSweeper.dll

Syntax

```
public class PathSegmentBuilder : IDisposable
```

Constructors

PathSegmentBuilder()

Declaration

```
public PathSegmentBuilder()
```

Methods

AppendPath(string)

Declaration

```
public PathSegmentBuilder AppendPath(string path)
```

Parameters

TYPE	NAME
------	------

string	path
--------	------

Returns

TYPE

PathSegmentBuilder

AppendPath(PathSegment)

Declaration

```
public PathSegmentBuilder AppendPath(PathSegment pathSegment)
```

Parameters

TYPE	NAME
------	------

PathSegment	pathSegment
-------------	-------------

Returns

TYPE

PathSegmentBuilder

Build()

Declaration

```
public PathSegment Build()
```

Returns

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

Implements

[IDisposable](#)

© Bus Fighter. All rights reserved.

Class SanityCheck

Inheritance

`object` → `SanityCheck`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public static class SanityCheck
```

Methods

WhenNull<T>(T, string)

Declaration

```
public static bool WhenNull<T>(T target, string message = "") where T : class
```

Parameters

TYPE	NAME
<code>T</code>	<code>target</code>
<code>string</code>	<code>message</code>

Returns

TYPE

bool

Type Parameters**NAME**

T

WhenTrue(bool, string)

Declaration

```
public static bool WhenTrue(bool condition, string message = "")
```

Parameters**TYPE NAME**

bool condition

string message

Returns**TYPE**

bool

© Bus Fighter. All rights reserved.

Class SanityCheckException

Inheritance

[object](#) → [Exception](#) → [SanityCheckException](#)

Implements

[ISerializable](#)

Inherited Members

[Exception.GetBaseException\(\)](#)

[Exception.GetType\(\)](#)

[Exception.ToString\(\)](#)

[Exception.Data](#)

[Exception.HelpLink](#)

[Exception.HResult](#)

[Exception.InnerException](#)

[Exception.Message](#)

[Exception.Source](#)

[Exception.StackTrace](#)

[Exception.TargetSite](#)

[Exception.SerializeObjectState](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class SanityCheckException : Exception, ISerializable
```

Constructors

[SanityCheckException\(\)](#)

Declaration

```
public SanityCheckException()
```

SanityCheckException(string)

Declaration

```
public SanityCheckException(string message)
```

Parameters

TYPE	NAME
------	------

string	message
--------	---------

SanityCheckException(string, Exception)

Declaration

```
public SanityCheckException(string message, Exception innerException)
```

Parameters

TYPE	NAME
------	------

string	message
--------	---------

Exception	innerException
-----------	----------------

Implements

ISerializable

Struct StateChangeRecord<TStatId>

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public struct StateChangeRecord<TStateId>
```

Type Parameters

NAME

TStateId

Fields

LastState

Declaration

```
public TStateId LastState
```

Field Value

TYPE

TStatId

Declaration

```
public TStateId NewState
```

Field Value

TYPE

TStateld

© Bus Fighter. All rights reserved.

Class StateExecutionException<TStatId>

Inheritance

[object](#) → [Exception](#) → [StateExecutionException<TStatId>](#)

Implements

[ISerializable](#)

Inherited Members

[Exception.GetBaseException\(\)](#)

[Exception.GetType\(\)](#)

[Exception.ToString\(\)](#)

[Exception.Data](#)

[Exception.HelpLink](#)

[Exception.HResult](#)

[Exception.InnerException](#)

[Exception.Message](#)

[Exception.Source](#)

[Exception.StackTrace](#)

[Exception.TargetSite](#)

[Exception.SerializeObjectState](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class StateExecutionException<TStateId> : Exception, ISerializable
```

Type Parameters

NAME

TStateId

Constructors

StateExecutionException(TStatId, Exception)

Declaration

```
public StateExecutionException(TStateId stateId, Exception innerException)
```

Parameters

TYPE	NAME
TStatId	stateId
Exception	innerException

Implements

ISerializable

© Bus Fighter. All rights reserved.

Class StateMachine<TStatId, TState, TStateMachine>

Inheritance

`object` → `StateMachine<TStatId, TState, TStateMachine>` → `GameStateMachine`

Implements

`IStateMachine<TStatId>`

`IDisposable`

Inherited Members

`object.Equals(object)`

`object.Equals(object, object)`

`object.GetHashCode()`

`object.GetType()`

`object.MemberwiseClone()`

`object.ReferenceEquals(object, object)`

`object.ToString()`

Namespace: `cfEngine.Util`

Assembly: `CatSweeper.dll`

Syntax

```
public class StateMachine<TStateId, TState, TStateMachine> : IStateMachine<TStateId>, IDisposable
```

Type Parameters

NAME

`TStateId`

`TState`

`TStateMachine`

Constructors

StateMachine()

Declaration

```
public StateMachine()
```

Properties

currentStatId

Declaration

```
public TStateId currentStateId { get; }
```

Property Value

TYPE

TStateId

lastStatId

Declaration

```
public TStateId lastStateId { get; }
```

Property Value

TYPE

TStateId

Methods

CanGoToState(TStateId, StateParam)

Declaration

```
public bool CanGoToState(TStateId id, StateParam param)
```

Parameters

TYPE	NAME
TStatId	id
StateParam	param

Returns

TYPE
bool

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public void Dispose()
```

ForceGoToState(TStatId, StateParam)

Declaration

```
public void ForceGoToState(TStateId nextStateId, StateParam param = null)
```

Parameters

TYPE	NAME
TStatId	nextStateId
StateParam	param

GetStateUnsafe(TStatId)

Declaration

```
public TState GetStateUnsafe(TStateId id)
```

Parameters

TYPE	NAME
------	------

TStateId	id
----------	----

Returns

TYPE

TState

GetStateUnsafe<T>(TStateId)

Declaration

```
public T GetStateUnsafe<T>(TStateId id) where T : TState
```

Parameters

TYPE	NAME
------	------

TStateId	id
----------	----

Returns

TYPE

T

Type Parameters

NAME

T

RegisterState(TState)

Declaration

```
public void RegisterState(TState state)
```

Parameters

TYPE NAME

TState state

SubscribeAfterStateChange(Action<StateChangeRecord<TStatId>>)

Declaration

```
public Subscription SubscribeAfterStateChange(Action<StateChangeRecord<TStateId>> listener)
```

Parameters

TYPE NAME

Action<StateChangeRecord<TStatId>> listener

Returns

TYPE

Subscription

SubscribeBeforeStateChange(Action<StateChangeRecord<TStatId>>)

Declaration

```
public Subscription SubscribeBeforeStateChange(Action<StateChangeRecord<TStateId>> listener)
```

Parameters

TYPE NAME

Action<StateChangeRecord<TStatId>> listener

Returns

TYPE

Subscription

TryGetState(TStatId, out TState)

Declaration

```
public bool TryGetState(TStateId id, out TState state)
```

Parameters

TYPE	NAME
TStatId	id
TState	state

Returns

TYPE
bool

TryGoToState(TStatId, StateParam)

Declaration

```
public bool TryGoToState(TStateId nextStateId, StateParam param = null)
```

Parameters

TYPE	NAME
TStatId	nextStateId
StateParam	param

Returns

TYPE
bool

Implements

IStateMachine<TStatId>
IDisposable

Class StateParam

Inheritance

object → StateParam → LoadStageState.InitParam → LoginState.Param

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class StateParam
```

© Bus Fighter. All rights reserved.

Class State<TStatId, TState, TStateMachine>

Inheritance

object → State<TStatId, TState, TStateMachine> → [GameState](#)

Implements

[IDisposable](#)

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public abstract class State<TStateId, TState, TStateMachine> : IDisposable where TState : Stat
```

Type Parameters

NAME

TStateId

TState

TStateMachine

Properties

Id

Declaration

```
public abstract TStateId Id { get; }
```

Property Value

TYPE

TStateId

StateMachine

Declaration

```
public TStateMachine StateMachine { get; }
```

Property Value

TYPE

TStateMachine

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Declaration

```
public virtual void Dispose()
```

IsReady()

Declaration

```
public virtual bool IsReady()
```

Returns

TYPE

bool

OnEndContext()

Declaration

```
protected virtual void OnEndContext()
```

StartContext(StateParam)

Declaration

```
public abstract void StartContext(StateParam param)
```

Parameters

TYPE	NAME
StateParam	param

Implements

IDisposable

© Bus Fighter. All rights reserved.

Class TypeExtension

Inheritance

`object` → TypeExtension

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public static class TypeExtension
```

Methods

FindDerivedTypes(Assembly, Type)

Declaration

```
public static IEnumerable<Type> FindDerivedTypes(Assembly assembly, Type baseType)
```

Parameters

TYPE	NAME
<code>Assembly</code>	<code>assembly</code>
<code>Type</code>	<code>baseType</code>

Returns

TYPE

IEnumerable<Type>

GetDefaultValue(Type)

Declaration

```
public static object GetDefaultValue(this Type type)
```

Parameters

TYPE	NAME
------	------

Type	type
------	------

Returns

TYPE

object

GetFlattenMethods(Type)

Declaration

```
public static MethodInfo[] GetFlattenMethods(this Type type)
```

Parameters

TYPE	NAME
------	------

Type	type
------	------

Returns

TYPE

MethodInfo[]

GetTypeName(Type)

Declaration

```
public static string GetTypeName(this Type type)
```

Parameters

TYPE	NAME
------	------

Type	type
------	------

Returns

TYPE

string

© Bus Fighter. All rights reserved.

Namespace cfGodotEngine.Asset

Classes

[AsyncResourceLoader](#)

[AsyncResourceLoader.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[AsyncResourceLoader.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

[AsyncResourceLoader.SignalName](#)

Cached StringNames for the signals contained in this class, for fast lookup.

[GDAtlas](#)

[GDAtlas.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[GDAtlas.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

[GDAtlas.SignalName](#)

Cached StringNames for the signals contained in this class, for fast lookup.

[GDAtlasPack](#)

[GDAtlasPack.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[GDAtlasPack.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

[GDAtlasPack.SignalName](#)

Cached StringNames for the signals contained in this class, for fast lookup.

[GDAtlasTextureRef](#)

[GDAtlasTextureRef.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[GDAtlasTextureRef.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

[GDAtlasTextureRef.SignalName](#)

Cached StringNames for the signals contained in this class, for fast lookup.

[ResourceAssetManager](#)

© Bus Fighter. All rights reserved.

Class AsyncResourceLoader

Inheritance

object → GodotObject → Node → [MonoInstance<AsyncResourceLoader>](#) → AsyncResourceLoader

Implements

[IDisposable](#)

Inherited Members

[MonoInstance<AsyncResourceLoader>.Instance](#)

Node.NotificationEnterTree

Node.NotificationExitTree

Node.NotificationMovedInParent

Node.NotificationReady

Node.NotificationPaused

Node.NotificationUnpaused

Node.NotificationPhysicsProcess

Node.NotificationProcess

Node.NotificationParented

Node.NotificationUnparented

Node.NotificationSceneInstantiated

Node.NotificationDragBegin

Node.NotificationDragEnd

Node.NotificationPathRenamed

Node.NotificationChildOrderChanged

Node.NotificationInternalProcess

Node.NotificationInternalPhysicsProcess

Node.NotificationPostEnterTree

Node.NotificationDisabled

Node.NotificationEnabled

Node.NotificationResetPhysicsInterpolation

Node.NotificationEditorPreSave

Node.NotificationEditorPostSave

Node.NotificationWMMouseEnter

Node.NotificationWMMouseExit

Node.NotificationWMWindowFocusIn

Node.NotificationWMWindowFocusOut

Node.NotificationWMCloseRequest

Node.NotificationWMGoBackRequest

Node.NotificationWMSizeChanged

Node.NotificationWMDpiChange

Node.NotificationVpMouseEnter
Node.NotificationVpMouseExit
Node.NotificationWMPositionChanged
Node.NotificationOsMemoryWarning
Node.NotificationTranslationChanged
Node.NotificationWMAbout
Node.NotificationCrash
Node.NotificationOslmeUpdate
Node.NotificationApplicationResumed
Node.NotificationApplicationPaused
Node.NotificationApplicationFocusIn
Node.NotificationApplicationFocusOut
Node.NotificationTextServerChanged
Node.GetNode<T>(NodePath)
Node.GetNodeOrNull<T>(NodePath)
Node.GetChild<T>(int, bool)
Node.GetChildOrNull<T>(int, bool)
Node.GetOwner<T>()
Node.GetOwnerOrNull<T>()
Node.SetParent<T>()
Node.SetParentOrNull<T>()
Node._EnterTree()
Node._ExitTree()
Node._GetConfigurationWarnings()
Node._Input(InputEvent)
Node._PhysicsProcess(double)
Node._Ready()
Node._ShortcutInput(InputEvent)
Node._UnhandledInput(InputEvent)
Node._UnhandledKeyInput(InputEvent)
Node.PrintOrphanNodes()
Node.AddSibling(Node, bool)
Node.AddChild(Node, bool, Node.InternalMode)
Node.RemoveChild(Node)
Node.Reparent(Node, bool)
Node.GetChildCount(bool)
Node.GetChildren(bool)
Node.GetChild(int, bool)
Node.HasNode(NodePath)
Node.GetNode(NodePath)
Node.GetNodeOrNull(NodePath)
Node.SetParent()
Node.FindChild(string, bool, bool)
Node.FindChildren(string, string, bool, bool)
Node.FindParent(string)
Node.HasNodeAndResource(NodePath)
Node.GetNodeAndResource(NodePath)
Node.IsInsideTree()
Node.IsPartOfEditedScene()
Node.IsAncestorOf(Node)

Node.IsGreater Than(Node)
Node.GetPath()
Node.GetPath To(Node, bool)
Node.AddToGroup(StringName, bool)
Node.RemoveFromGroup(StringName)
Node.IsInGroup(StringName)
Node.MoveChild(Node, int)
Node.GetGroups()
Node.GetIndex(bool)
Node.PrintTree()
Node.PrintTreePretty()
Node.GetTreeString()
Node.GetTreeStringPretty()
Node.PropagateNotification(int)
Node.PropagateCall(StringName, Array, bool)
Node.SetPhysicsProcess(bool)
Node.GetPhysicsProcessDelta Time()
Node.IsPhysicsProcessing()
Node.GetProcessDelta Time()
Node.SetProcess(bool)
Node.IsProcessing()
Node.SetProcessInput(bool)
Node.IsProcessingInput()
Node.SetProcessShortcutInput(bool)
Node.IsProcessingShortcutInput()
Node.SetProcessUnhandledInput(bool)
Node.IsProcessingUnhandledInput()
Node.SetProcessUnhandledKeyInput(bool)
Node.IsProcessingUnhandledKeyInput()
Node.CanProcess()
Node.SetDisplayFolded(bool)
Node.IsDisplayedFolded()
Node.SetProcessInternal(bool)
Node.IsProcessingInternal()
Node.SetPhysicsProcessInternal(bool)
Node.IsPhysicsProcessingInternal()
Node.IsPhysicsInterpolated()
Node.IsPhysicsInterpolatedAndEnabled()
Node.ResetPhysicsInterpolation()
Node.SetTranslationDomainInherited()
Node.GetWindow()
Node.GetLastExclusiveWindow()
Node.GetTree()
Node.CreateTween()
Node.Duplicate(int)
Node.ReplaceBy(Node, bool)
Node.setSceneInstanceLoadPlaceholder(bool)
Node.GetSceneInstanceLoadPlaceholder()
Node.setEditableInstance(Node, bool)
Node.setEditableInstance(Node)

```
Node.GetViewport()
Node.QueueFree()
Node.RequestReady()
Node.IsNodeReady()
Node.SetMultiplayerAuthority(int, bool)
Node.GetMultiplayerAuthority()
Node.IsMultiplayerAuthority()
Node.RpcConfig(StringName, Variant)
Node.GetRpcConfig()
Node.Atr(string, StringName)
Node.AtrN(string, StringName, int, StringName)
Node.Rpc(StringName, params Variant[])
Node.Rpc(StringName, ReadOnlySpan<Variant>)
Node.RpcId(long, StringName, params Variant[])
Node.RpcId(long, StringName, ReadOnlySpan<Variant>)
Node.UpdateConfigurationWarnings()
Node.CallDeferredThreadGroup(StringName, params Variant[])
Node.CallDeferredThreadGroup(StringName, ReadOnlySpan<Variant>)
Node.SetDeferredThreadGroup(StringName, Variant)
Node.NotifyDeferredThreadGroup(int)
Node.CallThreadSafe(StringName, params Variant[])
Node.CallThreadSafe(StringName, ReadOnlySpan<Variant>)
Node.SetThreadSafe(StringName, Variant)
Node.NotifyThreadSafe(int)
Node.EmitSignalReady()
Node.EmitSignalRenamed()
Node.EmitSignalTreeEntered()
Node.EmitSignalTreeExiting()
Node.EmitSignalTreeExited()
Node.EmitSignalChildEnteredTree(Node)
Node.EmitSignalChildExitingTree(Node)
Node.EmitSignalChildOrderChanged()
Node.EmitSignalReplacingBy(Node)
Node.EmitSignalEditorDescriptionChanged(Node)
Node.EmitSignalEditorStateChanged()
Node.InvokeGodotClassMethod(in godot_string_name, NativeVariantPtrArgs, out godot_variant)
Node.HasGodotClassMethod(in godot_string_name)
Node.HasGodotClassSignal(in godot_string_name)
Node.Name
Node.UniqueNameInOwner
Node.SceneFilePath
Node.Owner
Node.Multiplayer
Node.ProcessMode
Node.ProcessPriority
Node.ProcessPhysicsPriority
Node.ProcessThreadGroup
Node.ProcessThreadGroupOrder
Node.ProcessThreadMessages
Node.PhysicsInterpolationMode
```

Node.AutoTranslateMode
Node.EditorDescription
Node.Ready
Node.Renamed
Node.TreeEntered
Node.TreeExiting
Node.TreeExited
Node.ChildEnteredTree
Node.ChildExitingTree
Node.ChildOrderChanged
Node.ReplacingBy
Node.EditorDescriptionChanged
Node.EditorStateChanged
GodotObject.NotificationPostinitialize
GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)

GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)
GodotObject.Cally(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObjectGetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Namespace: **cfGodotEngine.Asset**

Assembly: CatSweeper.dll

Syntax

```
[ScriptPath("res://Modules/cfGodotEngine/asset/AsyncResourceLoader.cs")]
public class AsyncResourceLoader : MonoInstance<AsyncResourceLoader>, IDisposable
```

Methods

Load(string, in IProgress<float>, string, bool, CacheMode)

Declaration

```
public Task<Resource> Load(string path, in IProgress<float> progress, string typeHint = "", bo
```

Parameters

TYPE	NAME
string	path
IProgress<float>	progress
string	typeHint
bool	useSubThread
ResourceLoader.CacheMode	cacheMode

Returns

TYPE

```
Task<Resource>
```

LoadAsync(string, in IProgress<float>, string, bool, CacheMode)

Declaration

```
public static Task<Resource> LoadAsync(string path, in IProgress<float> progress, string typeH
```

Parameters

TYPE	NAME
string	path
IProgress<float>	progress
string	typeHint
bool	useSubThread
ResourceLoader.CacheMode	cacheMode

Returns

TYPE

Task<Resource>

SetLogger(ILogger)

Declaration

```
public void SetLogger(ILogger logger)
```

Parameters

TYPE	NAME
ILogger	logger

_Process(double)

Called during the processing step of the main loop. Processing happens at every frame and as fast as possible, so the `delta` time since the previous frame is not constant. `delta` is in seconds.

It is only called if processing is enabled, which is done automatically if this method is overridden, and can be toggled with `SetProcess(bool)`.

Processing happens in order of `Godot.Node.ProcessPriority`, lower priority values are called first. Nodes with the same priority are processed in tree order, or top to bottom as seen in the editor (also known as pre-order traversal).

Corresponds to the `Godot.Node.NotificationProcess` notification in `_Notification(int)`.

Note: This method is only called if the node is present in the scene tree (i.e. if it's not an orphan).

Note: `delta` will be larger than expected if running at a framerate lower than `Godot.Engine.PhysicsTicksPerSecond / Godot.Engine.MaxPhysicsStepsPerFrame` FPS. This is done to avoid "spiral of death" scenarios where performance would plummet due to an ever-increasing number of physics steps per frame. This behavior affects both

`_Process(double)` and `_PhysicsProcess(double)`. As a result, avoid using `delta` for time measurements in real-world seconds. Use the `Godot.Time` singleton's methods for this purpose instead, such as `Godot.Time.GetTicksUsec()`.

Declaration

```
public override void _Process(double delta)
```

Parameters

TYPE	NAME
double	delta

Overrides

`Node._Process(double)`

Implements

`IDisposable`

Extension Methods

`NodeUtil.DontDestroyOnLoad(Node)`

© Bus Fighter. All rights reserved.

Class AsyncResourceLoader.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

`object` → `GodotObject.MethodName` → `Node.MethodName` → `MonoInstance<AsyncResourceLoader>.MethodName` → `AsyncResourceLoader.MethodName`

Inherited Members

`Node.MethodName._EnterTree`
`Node.MethodName._ExitTree`
`Node.MethodName._GetConfigurationWarnings`
`Node.MethodName._Input`
`Node.MethodName._PhysicsProcess`
`Node.MethodName._Ready`
`Node.MethodName._ShortcutInput`
`Node.MethodName._UnhandledInput`
`Node.MethodName._UnhandledKeyInput`
`Node.MethodName.PrintOrphanNodes`
`Node.MethodName.AddSibling`
`Node.MethodName.SetName`
`Node.MethodName.GetName`
`Node.MethodName.AddChild`
`Node.MethodName.RemoveChild`
`Node.MethodName.Reparent`
`Node.MethodName.GetChildCount`
`Node.MethodName.GetChildren`
`Node.MethodName.GetChild`
`Node.MethodName.HasNode`
`Node.MethodName.GetNode`
`Node.MethodName.GetNodeOrNull`
`Node.MethodName.GetParent`
`Node.MethodName.FindChild`
`Node.MethodName.FindChildren`
`Node.MethodName.FindParent`
`Node.MethodName.HasNodeAndResource`
`Node.MethodName.GetNodeAndResource`
`Node.MethodName.IsInsideTree`
`Node.MethodName.IsPartOfEditedScene`
`Node.MethodName.IsAncestorOf`
`Node.MethodName.IsGreaterThan`
`Node.MethodName.GetPath`

Node.MethodName.GetPathTo
Node.MethodName.AddToGroup
Node.MethodName.RemoveFromGroup
Node.MethodName.IsInGroup
Node.MethodName.MoveChild
Node.MethodName.GetGroups
Node.MethodName.SetOwner
Node.MethodName.GetOwner
Node.MethodName.GetIndex
Node.MethodName.PrintTree
Node.MethodName.PrintTreePretty
Node.MethodName.GetTreeString
Node.MethodName.GetTreeStringPretty
Node.MethodName.SetSceneFilePath
Node.MethodName.GetSceneFilePath
Node.MethodName.PropagateNotification
Node.MethodName.PropagateCall
Node.MethodName.SetPhysicsProcess
Node.MethodName.GetPhysicsProcessDeltaTime
Node.MethodName.IsPhysicsProcessing
Node.MethodName.GetProcessDeltaTime
Node.MethodName.SetProcess
Node.MethodName.SetProcessPriority
Node.MethodName.GetProcessPriority
Node.MethodName.SetPhysicsProcessPriority
Node.MethodName.GetPhysicsProcessPriority
Node.MethodName.IsProcessing
Node.MethodName.SetProcessInput
Node.MethodName.IsProcessingInput
Node.MethodName.SetProcessShortcutInput
Node.MethodName.IsProcessingShortcutInput
Node.MethodName.SetProcessUnhandledInput
Node.MethodName.IsProcessingUnhandledInput
Node.MethodName.SetProcessUnhandledKeyInput
Node.MethodName.IsProcessingUnhandledKeyInput
Node.MethodName.SetProcessMode
Node.MethodName.GetProcessMode
Node.MethodName.CanProcess
Node.MethodName.SetProcessThreadGroup
Node.MethodName.GetProcessThreadGroup
Node.MethodName.SetProcessThreadMessages
Node.MethodName.GetProcessThreadMessages
Node.MethodName.SetProcessThreadGroupOrder
Node.MethodName.GetProcessThreadGroupOrder
Node.MethodName.SetDisplayFolded
Node.MethodName.IsDisplayedFolded
Node.MethodName.SetProcessInternal
Node.MethodName.IsProcessingInternal
Node.MethodName.SetPhysicsProcessInternal
Node.MethodName.IsPhysicsProcessingInternal

Node.MethodName.SetPhysicsInterpolationMode
Node.MethodName.GetPhysicsInterpolationMode
Node.MethodName.IsPhysicsInterpolated
Node.MethodName.IsPhysicsInterpolatedAndEnabled
Node.MethodName.ResetPhysicsInterpolation
Node.MethodName.SetAutoTranslateMode
Node.MethodName.GetAutoTranslateMode
Node.MethodName.SetTranslationDomainInherited
Node.MethodName.GetWindow
Node.MethodName.GetLastExclusiveWindow
Node.MethodName.GetTree
Node.MethodName.CreateTween
Node.MethodName.Duplicate
Node.MethodName.ReplaceBy
Node.MethodName.SetSceneInstanceLoadPlaceholder
Node.MethodName.GetSceneInstanceLoadPlaceholder
Node.MethodName.SetEditableInstance
Node.MethodName.IsEditableInstance
Node.MethodName.GetViewport
Node.MethodName.QueueFree
Node.MethodName.RequestReady
Node.MethodName.IsNodeReady
Node.MethodName.SetMultiplayerAuthority
Node.MethodName.GetMultiplayerAuthority
Node.MethodName.IsMultiplayerAuthority
Node.MethodName.GetMultiplayer
Node.MethodName.RpcConfig
Node.MethodName.GetRpcConfig
Node.MethodName.SetEditorDescription
Node.MethodName.GetEditorDescription
Node.MethodName._SetImportPath
Node.MethodName._GetImportPath
Node.MethodName.SetUniqueNameInOwner
Node.MethodName.IsUniqueNameInOwner
Node.MethodName.Atr
Node.MethodName.AtrN
Node.MethodName.Rpc
Node.MethodName.RpcId
Node.MethodName.UpdateConfigurationWarnings
Node.MethodName.CallDeferredThreadGroup
Node.MethodName.SetDeferredThreadGroup
Node.MethodName.NotifyDeferredThreadGroup
Node.MethodName.CallThreadSafe
Node.MethodName.SetThreadSafe
Node.MethodName.NotifyThreadSafe
GodotObject.MethodName._Get
GodotObject.MethodName._GetPropertyList
GodotObject.MethodName._IterGet
GodotObject.MethodName._IterInit
GodotObject.MethodName._IterNext

GodotObject.MethodName._Notification
GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceId
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyListChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr
GodotObject.MethodName.TrN

GodotObject.MethodName.GetTranslationDomain
GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfGodotEngine.Asset](#)

Assembly: CatSweeper.dll

Syntax

```
public class AsyncResourceLoader.MethodName : MonoInstance<AsyncResourceLoader>.MethodName
```

Fields

_Process

Cached name for the '_Process' method.

Declaration

```
public static readonly StringName _Process
```

Field Value

TYPE

StringName

Class AsyncResourceLoader.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

`object` → `GodotObject.PropertyName` → `Node.PropertyName` → `MonoInstance<AsyncResourceLoader>.PropertyName` → `AsyncResourceLoader.PropertyName`

Inherited Members

`Node.PropertyName._ImportPath`
`Node.PropertyName.Name`
`Node.PropertyName.UniqueNameInOwner`
`Node.PropertyName.SceneFilePath`
`Node.PropertyName.Owner`
`Node.PropertyName.Multiplayer`
`Node.PropertyName.ProcessMode`
`Node.PropertyName.ProcessPriority`
`Node.PropertyName.ProcessPhysicsPriority`
`Node.PropertyName.ProcessThreadGroup`
`Node.PropertyName.ProcessThreadGroupOrder`
`Node.PropertyName.ProcessThreadMessages`
`Node.PropertyName.PhysicsInterpolationMode`
`Node.PropertyName.AutoTranslateMode`
`Node.PropertyName.EditorDescription`
[`object.Equals\(object\)`](#)
[`object.Equals\(object, object\)`](#)
[`object.GetHashCode\(\)`](#)
[`object.GetType\(\)`](#)
[`object.MemberwiseClone\(\)`](#)
[`object.ReferenceEquals\(object, object\)`](#)
[`object.ToString\(\)`](#)

Namespace: [cfGodotEngine.Asset](#)

Assembly: CatSweeper.dll

Syntax

```
public class AsyncResourceLoader.PropertyName : MonoInstance<AsyncResourceLoader>.PropertyName
```

Fields

progressArray

Cached name for the 'progressArray' field.

Declaration

```
public static readonly StringName progressArray
```

Field Value

TYPE

StringName

© Bus Fighter. All rights reserved.

Class AsyncResourceLoader.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

`object` → `GodotObject.SignalName` → `Node.SignalName` → `MonoInstance<AsyncResourceLoader>.SignalName` → `AsyncResourceLoader.SignalName`

Inherited Members

`Node.SignalName.Ready`
`Node.SignalName.Renamed`
`Node.SignalName.TreeEntered`
`Node.SignalName.TreeExiting`
`Node.SignalName.TreeExited`
`Node.SignalName.ChildEnteredTree`
`Node.SignalName.ChildExitingTree`
`Node.SignalName.ChildOrderChanged`
`Node.SignalName.ReplacingBy`
`Node.SignalName.EditorDescriptionChanged`
`Node.SignalName.EditorStateChanged`
`GodotObject.SignalName.ScriptChanged`
`GodotObject.SignalName.PropertyListChanged`
`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfGodotEngine.Asset`

Assembly: CatSweeper.dll

Syntax

```
public class AsyncResourceLoader.SignalName : MonoInstance<AsyncResourceLoader>.SignalName
```

Class GDAtlas

Inheritance

[object](#) → GodotObject → RefCounted → Resource → GDAtlas

Implements

[IDisposable](#)

Inherited Members

Resource._GetRid()

Resource._ResetState()

[Resource._SetPathCache\(string\)](#)

Resource._SetupLocalToScene()

[Resource.TakeOverPath\(string\)](#)

[Resource.SetPathCache\(string\)](#)

Resource.GetRid()

Resource.GetLocalScene()

Resource.SetupLocalToScene()

Resource.ResetState()

[Resource.SetIdForPath\(string, string\)](#)

[Resource.GetIdForPath\(string\)](#)

Resource.IsBuiltIn()

Resource.GenerateSceneUniqueId()

Resource.EmitChanged()

[Resource.Duplicate\(bool\)](#)

Resource.EmitSignalChanged()

Resource.EmitSignalSetupLocalToSceneRequested()

Resource.InvokeGodotClassMethod(in godot_string_name, NativeVariantPtrArgs, out godot_variant)

Resource.HasGodotClassMethod(in godot_string_name)

Resource.HasGodotClassSignal(in godot_string_name)

Resource.ResourceLocalToScene

Resource.ResourcePath

Resource.ResourceName

Resource.ResourceSceneUniqueId

Resource.Changed

Resource.SetupLocalToSceneRequested

RefCounted.InitRef()

RefCounted.Reference()

RefCounted.Unreference()

RefCounted.GetReferenceCount()

GodotObject.NotificationPostInitialize

GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)
GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)

GodotObject.Callv(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObject.GetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Namespace: **cfGodotEngine.Asset**

Assembly: CatSweeper.dll

Syntax

```
[Tool]
[GlobalClass]
[ScriptPath("res://Modules/cfGodotEngine/atlaspack/GDAtlas.cs")]
public class GDAtlas : Resource, IDisposable
```

Fields

atlasId

Declaration

```
[Export(PropertyHint.None, "")]  
public string atlasId
```

Field Value

TYPE

string

atlasTexture

Declaration

```
[Export(PropertyHint.None, "")]  
public Texture2D atlasTexture
```

Field Value

TYPE

Texture2D

dimension

Declaration

```
[Export(PropertyHint.None, "")]  
public Vector2 dimension
```

Field Value

TYPE

Vector2

imageMap

```
[Export(PropertyHint.None, "")]  
public Dictionary<string, AtlasTexture> imageMap
```

Field Value

TYPE

Dictionary<[string](#), [AtlasTexture](#)>

Implements

[IDisposable](#)

© Bus Fighter. All rights reserved.

Class GDAtlas.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.MethodName](#) → [RefCounted.MethodName](#) → [Resource.MethodName](#) → [GDAtlas.MethodName](#)

Inherited Members

[Resource.MethodName._GetRid](#)
[Resource.MethodName._ResetState](#)
[Resource.MethodName._SetPathCache](#)
[Resource.MethodName._SetupLocalToScene](#)
[Resource.MethodName.SetPath](#)
[Resource.MethodName.TakeOverPath](#)
[Resource.MethodName.GetPath](#)
[Resource.MethodName.SetPathCache](#)
[Resource.MethodNameSetName](#)
[Resource.MethodName.GetName](#)
[Resource.MethodName.GetRid](#)
[Resource.MethodName.SetLocalToScene](#)
[Resource.MethodName.IsLocalToScene](#)
[Resource.MethodName.GetLocalScene](#)
[Resource.MethodName.SetupLocalToScene](#)
[Resource.MethodName.ResetState](#)
[Resource.MethodName.SetIdForPath](#)
[Resource.MethodName.GetIdForPath](#)
[Resource.MethodName.IsBuiltIn](#)
[Resource.MethodName.GenerateSceneUniqueId](#)
[Resource.MethodName.SetSceneUniqueId](#)
[Resource.MethodName.GetSceneUniqueId](#)
[Resource.MethodName.EmitChanged](#)
[Resource.MethodName.Duplicate](#)
[RefCounted.MethodName.InitRef](#)
[RefCounted.MethodName.Reference](#)
[RefCounted.MethodName.Unreference](#)
[RefCounted.MethodName.GetReferenceCount](#)
[GodotObject.MethodName._Get](#)
[GodotObject.MethodName._GetPropertyList](#)
[GodotObject.MethodName._IterGet](#)
[GodotObject.MethodName._IterInit](#)
[GodotObject.MethodName._IterNext](#)
[GodotObject.MethodName._Notification](#)

GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceId
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr
GodotObject.MethodName.TrN
GodotObject.MethodName.GetTranslationDomain

GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfGodotEngine.Asset`

Assembly: CatSweeper.dll

Syntax

```
public class GDAtlas.MethodName : Resource.MethodName
```

© Bus Fighter. All rights reserved.

Class GDAtlas.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.PropertyName](#) → [RefCounted.PropertyName](#) → [Resource.PropertyName](#) → [GDAtlas.PropertyName](#)

Inherited Members

[Resource.PropertyName.ResourceLocalToScene](#)

[Resource.PropertyName.ResourcePath](#)

[Resource.PropertyName.ResourceName](#)

[Resource.PropertyName.ResourceSceneUniqueld](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Asset](#)

Assembly: [CatSweeper.dll](#)

Syntax

```
public class GDAtlas.PropertyName : Resource.PropertyName
```

Fields

atlasId

Cached name for the 'atlasId' field.

Declaration

```
public static readonly StringName atlasId
```

Field Value

TYPE

StringName

atlasTexture

Cached name for the 'atlasTexture' field.

Declaration

```
public static readonly StringName atlasTexture
```

Field Value

TYPE

StringName

dimension

Cached name for the 'dimension' field.

Declaration

```
public static readonly StringName dimension
```

Field Value

TYPE

StringName

imageMap

Cached name for the 'imageMap' field.

Declaration

```
public static readonly StringName imageMap
```

Field Value

TYPE

StringName

© Bus Fighter. All rights reserved.

Class GDAtlas.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.SignalName](#) → [RefCounted.SignalName](#) → [Resource.SignalName](#) → [GDAtlas.SignalName](#)

Inherited Members

[Resource.SignalName.Changed](#)

[Resource.SignalName.SetupLocalToSceneRequested](#)

[GodotObject.SignalName.ScriptChanged](#)

[GodotObject.SignalName.PropertyListChanged](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Asset](#)

Assembly: CatSweeper.dll

Syntax

```
public class GDAtlas.SignalName : Resource.SignalName
```

© Bus Fighter. All rights reserved.

Class GDAtlasPack

Inheritance

[object](#) → [GodotObject](#) → [RefCounted](#) → [Resource](#) → [GDAtlasPack](#)

Implements

[IDisposable](#)

Inherited Members

[Resource._GetRid\(\)](#)

[Resource._ResetState\(\)](#)

[Resource._SetPathCache\(string\)](#)

[Resource._SetupLocalToScene\(\)](#)

[Resource.TakeOverPath\(string\)](#)

[Resource.SetPathCache\(string\)](#)

[Resource.GetRid\(\)](#)

[Resource.GetLocalScene\(\)](#)

[Resource.SetupLocalToScene\(\)](#)

[Resource.ResetState\(\)](#)

[Resource.SetIdForPath\(string, string\)](#)

[Resource.GetIdForPath\(string\)](#)

[Resource.IsBuiltIn\(\)](#)

[Resource.GenerateSceneUniqueId\(\)](#)

[Resource.EmitChanged\(\)](#)

[Resource.Duplicate\(bool\)](#)

[Resource.EmitSignalChanged\(\)](#)

[Resource.EmitSignalSetupLocalToSceneRequested\(\)](#)

[Resource.InvokeGodotClassMethod\(in godot_string_name, NativeVariantPtrArgs, out godot_variant\)](#)

[Resource.HasGodotClassMethod\(in godot_string_name\)](#)

[Resource.HasGodotClassSignal\(in godot_string_name\)](#)

[Resource.ResourceLocalToScene](#)

[Resource.ResourcePath](#)

[Resource.ResourceName](#)

[Resource.ResourceSceneUniqueId](#)

[Resource.Changed](#)

[Resource.SetupLocalToSceneRequested](#)

[RefCounted.InitRef\(\)](#)

[RefCounted.Reference\(\)](#)

[RefCounted.Unreference\(\)](#)

[RefCounted.GetReferenceCount\(\)](#)

[GodotObject.NotificationPostInitialize](#)

GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)
GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)

GodotObject.Callv(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObject.GetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Namespace: **cfGodotEngine.Asset**

Assembly: CatSweeper.dll

Syntax

```
[Tool]
[GlobalClass]
[ScriptPath("res://Modules/cfGodotEngine/atlaspack/GDAtlasPack.cs")]
public class GDAtlasPack : Resource, IDisposable
```

Fields

Declaration

```
[Export(PropertyHint.None, "")]  
public Array<GAtlas> atlasList
```

Field Value

TYPE

Array<[GAtlas](#)>

Methods

AddPack(string, AtlasContext, Texture2D)

Declaration

```
public void AddPack(string atlasId, AtlasContext context, Texture2D atlasTexture)
```

Parameters

TYPE	NAME
string	atlasId
AtlasContext	context
Texture2D	atlasTexture

Implements

[IDisposable](#)

Class GDAtlasPack.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.MethodName](#) → [RefCounted.MethodName](#) → [Resource.MethodName](#) → [GDAtlasPack.MethodName](#)

Inherited Members

[Resource.MethodName._GetRid](#)
[Resource.MethodName._ResetState](#)
[Resource.MethodName._SetPathCache](#)
[Resource.MethodName._SetupLocalToScene](#)
[Resource.MethodName.SetPath](#)
[Resource.MethodName.TakeOverPath](#)
[Resource.MethodName.GetPath](#)
[Resource.MethodName.SetPathCache](#)
[Resource.MethodNameSetName](#)
[Resource.MethodName.GetName](#)
[Resource.MethodName.GetRid](#)
[Resource.MethodName.SetLocalToScene](#)
[Resource.MethodName.IsLocalToScene](#)
[Resource.MethodName.GetLocalScene](#)
[Resource.MethodName.SetupLocalToScene](#)
[Resource.MethodName.ResetState](#)
[Resource.MethodName.SetIdForPath](#)
[Resource.MethodName.GetIdForPath](#)
[Resource.MethodName.IsBuiltIn](#)
[Resource.MethodName.GenerateSceneUniqueId](#)
[Resource.MethodName.SetSceneUniqueId](#)
[Resource.MethodName.GetSceneUniqueId](#)
[Resource.MethodName.EmitChanged](#)
[Resource.MethodName.Duplicate](#)
[RefCounted.MethodName.InitRef](#)
[RefCounted.MethodName.Reference](#)
[RefCounted.MethodName.Unreference](#)
[RefCounted.MethodName.GetReferenceCount](#)
[GodotObject.MethodName._Get](#)
[GodotObject.MethodName._GetPropertyList](#)
[GodotObject.MethodName._IterGet](#)
[GodotObject.MethodName._IterInit](#)
[GodotObject.MethodName._IterNext](#)
[GodotObject.MethodName._Notification](#)

GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceId
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr
GodotObject.MethodName.TrN
GodotObject.MethodName.GetTranslationDomain

GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfGodotEngine.Asset`

Assembly: CatSweeper.dll

Syntax

```
public class GDAtlasPack.MethodName : Resource.MethodName
```

© Bus Fighter. All rights reserved.

Class GDAtlasPack.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.PropertyName](#) → [RefCounted.PropertyName](#) → [Resource.PropertyName](#) → [GDAtlasPack.PropertyName](#)

Inherited Members

[Resource.PropertyName.ResourceLocalToScene](#)

[Resource.PropertyName.ResourcePath](#)

[Resource.PropertyName.ResourceName](#)

[Resource.PropertyName.ResourceSceneUniqueld](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Asset](#)

Assembly: CatSweeper.dll

Syntax

```
public class GDAtlasPack.PropertyName : Resource.PropertyName
```

Fields

atlasList

Cached name for the 'atlasList' field.

Declaration

```
public static readonly StringName atlasList
```

Field Value

TYPE

StringName

© Bus Fighter. All rights reserved.

Class GDAtlasPack.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.SignalName](#) → [RefCounted.SignalName](#) → [Resource.SignalName](#) → [GDAtlasPack.SignalName](#)

Inherited Members

[Resource.SignalName.Changed](#)

[Resource.SignalName.SetupLocalToSceneRequested](#)

[GodotObject.SignalName.ScriptChanged](#)

[GodotObject.SignalName.PropertyListChanged](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Asset](#)

Assembly: CatSweeper.dll

Syntax

```
public class GDAtlasPack.SignalName : Resource.SignalName
```

© Bus Fighter. All rights reserved.

Class GDAtlasTextureRef

Inheritance

[object](#) → [GodotObject](#) → [RefCounted](#) → [Resource](#) → [GDAtlasTextureRef](#)

Implements

[IDisposable](#)

Inherited Members

[Resource._GetRid\(\)](#)

[Resource._ResetState\(\)](#)

[Resource._SetPathCache\(string\)](#)

[Resource._SetupLocalToScene\(\)](#)

[Resource.TakeOverPath\(string\)](#)

[Resource.SetPathCache\(string\)](#)

[Resource.GetRid\(\)](#)

[Resource.GetLocalScene\(\)](#)

[Resource.SetupLocalToScene\(\)](#)

[Resource.ResetState\(\)](#)

[Resource.SetIdForPath\(string, string\)](#)

[Resource.GetIdForPath\(string\)](#)

[Resource.IsBuiltIn\(\)](#)

[Resource.GenerateSceneUniqueId\(\)](#)

[Resource.EmitChanged\(\)](#)

[Resource.Duplicate\(bool\)](#)

[Resource.EmitSignalChanged\(\)](#)

[Resource.EmitSignalSetupLocalToSceneRequested\(\)](#)

[Resource.InvokeGodotClassMethod\(in godot_string_name, NativeVariantPtrArgs, out godot_variant\)](#)

[Resource.HasGodotClassMethod\(in godot_string_name\)](#)

[Resource.HasGodotClassSignal\(in godot_string_name\)](#)

[Resource.ResourceLocalToScene](#)

[Resource.ResourcePath](#)

[Resource.ResourceName](#)

[Resource.ResourceSceneUniqueId](#)

[Resource.Changed](#)

[Resource.SetupLocalToSceneRequested](#)

[RefCounted.InitRef\(\)](#)

[RefCounted.Reference\(\)](#)

[RefCounted.Unreference\(\)](#)

[RefCounted.GetReferenceCount\(\)](#)

[GodotObject.NotificationPostInitialize](#)

GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)
GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)

GodotObject.Callv(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObject.GetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Namespace: **cfGodotEngine.Asset**

Assembly: CatSweeper.dll

Syntax

```
[Tool]
[GlobalClass]
[ScriptPath("res://Modules/cfGodotEngine/atlaspack/GDAtlasTextureRef.cs")]
public class GDAtlasTextureRef : Resource, IDisposable
```

Properties

atlasPack

Declaration

```
[Export(PropertyHint.None, "")]  
public GDAtlasPack atlasPack { get; set; }
```

Property Value

TYPE

GDAtlasPack

imageName

Declaration

```
[Export(PropertyHint.None, "")]  
public string imageName { get; set; }
```

Property Value

TYPE

string

Events

OnAtlasTextureUpdated

Declaration

```
public event Action<AtlasTexture> OnAtlasTextureUpdated
```

Event Type

TYPE

Action<AtlasTexture>

Implements

[IDisposable](#)

© Bus Fighter. All rights reserved.

Class GDAtlasTextureRef.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.MethodName](#) → [RefCounted.MethodName](#) → [Resource.MethodName](#) → [GDAtlasTextureRef.MethodName](#)

Inherited Members

[Resource.MethodName._GetRid](#)
[Resource.MethodName._ResetState](#)
[Resource.MethodName._SetPathCache](#)
[Resource.MethodName._SetupLocalToScene](#)
[Resource.MethodName.SetPath](#)
[Resource.MethodName.TakeOverPath](#)
[Resource.MethodName.GetPath](#)
[Resource.MethodName.SetPathCache](#)
[Resource.MethodNameSetName](#)
[Resource.MethodName.GetName](#)
[Resource.MethodName.GetRid](#)
[Resource.MethodName.SetLocalToScene](#)
[Resource.MethodName.IsLocalToScene](#)
[Resource.MethodName.GetLocalScene](#)
[Resource.MethodName.SetupLocalToScene](#)
[Resource.MethodName.ResetState](#)
[Resource.MethodName.SetIdForPath](#)
[Resource.MethodName.GetIdForPath](#)
[Resource.MethodName.IsBuiltIn](#)
[Resource.MethodName.GenerateSceneUniqueId](#)
[Resource.MethodName.SetSceneUniqueId](#)
[Resource.MethodName.GetSceneUniqueId](#)
[Resource.MethodName.EmitChanged](#)
[Resource.MethodName.Duplicate](#)
[RefCounted.MethodName.InitRef](#)
[RefCounted.MethodName.Reference](#)
[RefCounted.MethodName.Unreference](#)
[RefCounted.MethodName.GetReferenceCount](#)
[GodotObject.MethodName._Get](#)
[GodotObject.MethodName._GetPropertyList](#)
[GodotObject.MethodName._IterGet](#)
[GodotObject.MethodName._IterInit](#)
[GodotObject.MethodName._IterNext](#)

GodotObject.MethodName._Notification
GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceId
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyListChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr
GodotObject.MethodName.TrN

GodotObject.MethodName.GetTranslationDomain
GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfGodotEngine.Asset](#)

Assembly: CatSweeper.dll

Syntax

```
public class GDAtlasTextureRef.MethodName : Resource.MethodName
```

Fields

OnTextureUpdate

Cached name for the 'OnTextureUpdate' method.

Declaration

```
public static readonly StringName OnTextureUpdate
```

Field Value

TYPE

StringName

Class GDAtlasTextureRef.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.PropertyName](#) → [RefCounted.PropertyName](#) → [Resource.PropertyName](#) → [GDAtlasTextureRef.PropertyName](#)

Inherited Members

[Resource.PropertyName.ResourceLocalToScene](#)

[Resource.PropertyName.ResourcePath](#)

[Resource.PropertyName.ResourceName](#)

[Resource.PropertyName.ResourceSceneUniqueld](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Asset](#)

Assembly: CatSweeper.dll

Syntax

```
public class GDAtlasTextureRef.PropertyName : Resource.PropertyName
```

Fields

_atlasPack

Cached name for the '_atlasPack' field.

Declaration

```
public static readonly StringName _atlasPack
```

Field Value

TYPE

StringName

_imageName

Cached name for the '_imageName' field.

Declaration

```
public static readonly StringName _imageName
```

Field Value

TYPE

StringName

atlasPack

Cached name for the 'atlasPack' property.

Declaration

```
public static readonly StringName atlasPack
```

Field Value

TYPE

StringName

imageName

Cached name for the 'imageName' property.

Declaration

```
public static readonly StringName imageName
```

TYPE

StringName

© Bus Fighter. All rights reserved.

Class GDAtlasTextureRef.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.SignalName](#) → [RefCounted.SignalName](#) → [Resource.SignalName](#) → [GDAtlasTextureRef.SignalName](#)

Inherited Members

[Resource.SignalName.Changed](#)

[Resource.SignalName.SetupLocalToSceneRequested](#)

[GodotObject.SignalName.ScriptChanged](#)

[GodotObject.SignalName.PropertyListChanged](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Asset](#)

Assembly: CatSweeper.dll

Syntax

```
public class GDAtlasTextureRef.SignalName : Resource.SignalName
```

© Bus Fighter. All rights reserved.

Class ResourceAssetManager

Inheritance

object → AssetManager<Resource> → ResourceAssetManager

Implements

IService

IDisposable

Inherited Members

AssetManager<Resource>.Load<T>(string)

AssetManager<Resource>.LoadAsync<T>(string, CancellationToken)

AssetManager<Resource>.TryGetAsset<T>(string, out T)

AssetManager<Resource>.Dispose()

object.Equals(object)

object.Equals(object, object)

object.GetHashCode()

object.GetType()

object.MemberwiseClone()

object.ReferenceEquals(object, object)

object.ToString()

Namespace: [cfGodotEngine.Asset](#)

Assembly: CatSweeper.dll

Syntax

```
public class ResourceAssetManager : AssetManager<Resource>, IService, IDisposable
```

Methods

_LoadAsync<T>(string, CancellationToken)

Declaration

```
protected override Task<AssetHandle<T>> _LoadAsync<T>(string path, CancellationToken token = d
```

Parameters

TYPE	NAME
string	path
CancellationToken	token

Returns

TYPE

Task<AssetHandle<T>>

Type Parameters

NAME

T

Overrides

AssetManager<Resource>._LoadAsync<T>(string, CancellationToken)

_Load<T>(string)

Declaration

```
protected override AssetHandle<T> _Load<T>(string path) where T : class, Resource
```

Parameters

TYPE	NAME
string	path

Returns

TYPE

AssetHandle<T>

Type Parameters

NAME

T

Overrides

AssetManager<Resource>._Load<T>(string)

Implements

[IService](#)

[IDisposable](#)

© Bus Fighter. All rights reserved.

Namespace cfGodotEngine.Controls

Classes

[AtlasTextureButton](#)

[AtlasTextureButton.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[AtlasTextureButton.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

[AtlasTextureButton.SignalName](#)

Cached StringNames for the signals contained in this class, for fast lookup.

[AtlasTextureRect](#)

[AtlasTextureRect.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[AtlasTextureRect.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

[AtlasTextureRect.SignalName](#)

Cached StringNames for the signals contained in this class, for fast lookup.

© Bus Fighter. All rights reserved.

Class AtlasTextureButton

Inheritance

[object](#) → [GodotObject](#) → [Node](#) → [CanvasItem](#) → [Control](#) → [BaseButton](#) → [TextureButton](#) → [AtlasTextureButton](#)

Implements

[IDisposable](#)

Inherited Members

[TextureButton.InvokeGodotClassMethod\(in godot_string_name, NativeVariantPtrArgs, out godot_variant\)](#)

[TextureButton.HasGodotClassMethod\(in godot_string_name\)](#)

[TextureButton.HasGodotClassSignal\(in godot_string_name\)](#)

[TextureButton.TextureNormal](#)

[TextureButton.TexturePressed](#)

[TextureButton.TextureHover](#)

[TextureButton.TextureDisabled](#)

[TextureButton.TextureFocused](#)

[TextureButton.TextureClickMask](#)

[TextureButton.IgnoreTextureSize](#)

[TextureButton.StretchMode](#)

[TextureButton.FlipH](#)

[TextureButton.FlipV](#)

[BaseButton._Pressed\(\)](#)

[BaseButton._Toggled\(bool\)](#)

[BaseButton.SetPressedNoSignal\(bool\)](#)

[BaseButton.IsHovered\(\)](#)

[BaseButton.GetDrawMode\(\)](#)

[BaseButton.EmitSignalPressed\(\)](#)

[BaseButton.EmitSignalButtonUp\(\)](#)

[BaseButton.EmitSignalButtonDown\(\)](#)

[BaseButton.EmitSignalToggled\(bool\)](#)

[BaseButton.Disabled](#)

[BaseButton.ToggleMode](#)

[BaseButton.ButtonPressed](#)

[BaseButton.ActionMode](#)

[BaseButton.ButtonMask](#)

[BaseButton.KeepPressedOutside](#)

[BaseButton.ButtonGroup](#)

[BaseButton.Shortcut](#)

[BaseButton.ShortcutFeedback](#)

[BaseButton.ShortcutInTooltip](#)

BaseButton.Pressed
BaseButton.ButtonUp
BaseButtonButtonDown
BaseButton.Toggled
Control.NotificationResized
Control.NotificationMouseEnter
Control.NotificationMouseExit
Control.NotificationMouseEnterSelf
Control.NotificationMouseExitSelf
Control.NotificationFocusEnter
Control.NotificationFocusExit
Control.NotificationThemeChanged
Control.NotificationScrollBegin
Control.NotificationScrollEnd
Control.NotificationLayoutDirectionChanged
Control._CanDropData(Vector2, Variant)
Control._DropData(Vector2, Variant)
Control._GetDragData(Vector2)
Control._GetMinimumSize()
Control._GetTooltip(Vector2)
Control._GuilInput(InputEvent)
Control._HasPoint(Vector2)
Control._MakeCustomTooltip(string)
Control._StructuredTextParser(Array, string)
Control.AcceptEvent()
Control.GetMinimumSize()
Control.GetCombinedMinimumSize()
Control.SetAnchorsPreset(Control.LayoutPreset, bool)
Control.SetOffsetsPreset(Control.LayoutPreset, Control.LayoutPresetMode, int)
Control.SetAnchorsAndOffsetsPreset(Control.LayoutPreset, Control.LayoutPresetMode, int)
Control.SetAnchor(Side, float, bool, bool)
Control.SetAnchorAndOffset(Side, float, float, bool)
Control.SetBegin(Vector2)
Control.SetEnd(Vector2)
Control.SetPosition(Vector2, bool)
Control.SetSize(Vector2, bool)
Control.ResetSize()
Control.SetGlobalPosition(Vector2, bool)
Control.GetBegin()
Control.GetEnd()
Control.GetParentAreaSize()
Control.GetScreenPosition()
Control.GetRect()
Control.GetGlobalRect()
Control.HasFocus()
Control.GrabFocus()
Control.ReleaseFocus()
Control.FindPrevValidFocus()
Control.FindNextValidFocus()
Control.FindValidFocusNeighbor(Side)

```
Control.BeginBulkThemeOverride()
Control.EndBulkThemeOverride()
Control.AddThemelconOverride(StringName, Texture2D)
Control.AddThemeStyleboxOverride(StringName, StyleBox)
Control.AddThemeFontOverride(StringName, Font)
Control.AddThemeFontSizeOverride(StringName, int)
Control.AddThemeColorOverride(StringName, Color)
Control.AddThemeConstantOverride(StringName, int)
Control.RemoveThemelconOverride(StringName)
Control.RemoveThemeStyleboxOverride(StringName)
Control.RemoveThemeFontOverride(StringName)
Control.RemoveThemeFontSizeOverride(StringName)
Control.RemoveThemeColorOverride(StringName)
Control.RemoveThemeConstantOverride(StringName)
Control.GetThemelcon(StringName, StringName)
Control.GetThemeStylebox(StringName, StringName)
Control.GetThemeFont(StringName, StringName)
Control.GetThemeFontSize(StringName, StringName)
Control.GetThemeColor(StringName, StringName)
Control.GetThemeConstant(StringName, StringName)
Control.HasThemelconOverride(StringName)
Control.HasThemeStyleboxOverride(StringName)
Control.HasThemeFontOverride(StringName)
Control.HasThemeFontSizeOverride(StringName)
Control.HasThemeColorOverride(StringName)
Control.HasThemeConstantOverride(StringName)
Control.HasThemelcon(StringName, StringName)
Control.HasThemeStylebox(StringName, StringName)
Control.HasThemeFont(StringName, StringName)
Control.HasThemeFontSize(StringName, StringName)
Control.HasThemeColor(StringName, StringName)
Control.HasThemeConstant(StringName, StringName)
Control.GetThemeDefaultBaseScale()
Control.GetThemeDefaultFont()
Control.GetThemeDefaultFontSize()
Control.GetParentControl()
Control.GetTooltip(Vector2?)
Control.GetCursorShape(Vector2?)
Control.ForceDrag(Variant, Control)
Control.GrabClickFocus()
Control.SetDragForwarding(Callable, Callable, Callable)
Control.SetDragPreview(Control)
Control.IsDragSuccessful()
Control.WarpMouse(Vector2)
Control.UpdateMinimumSize()
Control.IsLayoutRtl()
Control.EmitSignalResized()
Control.EmitSignalGuilnput(InputEvent)
Control.EmitSignalMouseEntered()
Control.EmitSignalMouseExited()
```

Control.EmitSignalFocusEntered()
Control.EmitSignalFocusExited()
Control.EmitSignalSizeFlagsChanged()
Control.EmitSignalMinimumSizeChanged()
Control.EmitSignalThemeChanged()
Control.ClipContents
Control.CustomMinimumSize
Control.LayoutDirection
Control.AnchorLeft
Control.AnchorTop
Control.AnchorRight
Control.AnchorBottom
Control.OffsetLeft
Control.OffsetTop
Control.OffsetRight
Control.OffsetBottom
Control.GrowHorizontal
Control.GrowVertical
Control.Size
Control.Position
Control.GlobalPosition
Control.Rotation
Control.RotationDegrees
Control.Scale
Control.PivotOffset
Control.SizeFlagsHorizontal
Control.SizeFlagsVertical
Control.SizeFlagsStretchRatio
Control.LocalizeNumeralSystem
Control.AutoTranslate
Control.TooltipText
Control.TooltipAutoTranslateMode
Control.FocusNeighborLeft
Control.FocusNeighborTop
Control.FocusNeighborRight
Control.FocusNeighborBottom
Control.FocusNext
Control.FocusPrevious
Control.FocusMode
Control.MouseFilter
Control.MouseForcePassScrollEvents
Control.MouseDefaultCursorShape
Control.ShortcutContext
Control.Theme
Control.ThemeTypeVariation
Control.Resized
Control.GuiInput
Control.MouseEntered
Control.MouseExited
Control.FocusEntered

Control.FocusExited
Control.SizeFlagsChanged
Control.MinimumSizeChanged
Control.ThemeChanged
CanvasItem.NotificationTransformChanged
CanvasItem.NotificationLocalTransformChanged
CanvasItem.NotificationDraw
CanvasItem.NotificationVisibilityChanged
CanvasItem.NotificationEnterCanvas
CanvasItem.NotificationExitCanvas
CanvasItem.NotificationWorld2DChanged
CanvasItem._Draw()
CanvasItem.GetCanvasItem()
CanvasItem.IsVisibleInTree()
CanvasItem.Show()
CanvasItem.Hide()
CanvasItem.QueueRedraw()
CanvasItem.MoveToFront()
CanvasItem.DrawLine(Vector2, Vector2, Color, float, bool)
CanvasItem.DrawDashedLine(Vector2, Vector2, Color, float, float, bool, bool)
CanvasItem.DrawPolyline(Vector2[], Color, float, bool)
CanvasItem.DrawPolyline(ReadOnlySpan<Vector2>, Color, float, bool)
CanvasItem.DrawPolylineColors(Vector2[], Color[], float, bool)
CanvasItem.DrawPolylineColors(ReadOnlySpan<Vector2>, ReadOnlySpan<Color>, float, bool)
CanvasItem.DrawArc(Vector2, float, float, float, int, Color, float, bool)
CanvasItem.DrawMultiline(Vector2[], Color, float, bool)
CanvasItem.DrawMultiline(ReadOnlySpan<Vector2>, Color, float, bool)
CanvasItem.DrawMultilineColors(Vector2[], Color[], float, bool)
CanvasItem.DrawMultilineColors(ReadOnlySpan<Vector2>, ReadOnlySpan<Color>, float, bool)
CanvasItem.DrawRect(Rect2, Color, bool, float, bool)
CanvasItem.DrawCircle(Vector2, float, Color, bool, float, bool)
CanvasItem.DrawTexture(Texture2D, Vector2, Color?)
CanvasItem.DrawTextureRect(Texture2D, Rect2, bool, Color?, bool)
CanvasItem.DrawTextureRectRegion(Texture2D, Rect2, Rect2, Color?, bool, bool)
CanvasItem.DrawMsdfTextureRectRegion(Texture2D, Rect2, Rect2, Color?, double, double, double)
CanvasItem.DrawLcdTextureRectRegion(Texture2D, Rect2, Rect2, Color?)
CanvasItem.DrawStyleBox(StyleBox, Rect2)
CanvasItem.DrawPrimitive(Vector2[], Color[], Vector2[], Texture2D)
CanvasItem.DrawPrimitive(ReadOnlySpan<Vector2>, ReadOnlySpan<Color>, ReadOnlySpan<Vector2>, Texture2D)
CanvasItem.DrawPolygon(Vector2[], Color[], Vector2[], Texture2D)
CanvasItem.DrawPolygon(ReadOnlySpan<Vector2>, ReadOnlySpan<Color>, ReadOnlySpan<Vector2>, Texture2D)
CanvasItem.DrawColoredPolygon(Vector2[], Color, Vector2[], Texture2D)
CanvasItem.DrawColoredPolygon(ReadOnlySpan<Vector2>, Color, ReadOnlySpan<Vector2>, Texture2D)
CanvasItem.DrawString(Font, Vector2, string, HorizontalAlignment, float, int, Color?, TextServer.JustificationFlag, TextServer.Direction, TextServer.Orientation)
CanvasItem.DrawString(Font, Vector2, string, HorizontalAlignment, float, int, int, Color?, TextServer.LineBreakFlag, TextServer.JustificationFlag, TextServer.Direction, TextServer.Orientation)
CanvasItem.DrawStringOutline(Font, Vector2, string, HorizontalAlignment, float, int, int, Color?, TextServer.JustificationFlag, TextServer.Direction, TextServer.Orientation)

CanvasItem.DrawMultilineStringOutline(Font, Vector2, string, HorizontalAlignment, float, int, int, int, Color?, TextServer.LineBreakFlag, TextServer.JustificationFlag, TextServer.Direction, TextServer.Orientation)
CanvasItem.DrawChar(Font, Vector2, string, int, Color?)
CanvasItem.DrawCharOutline(Font, Vector2, string, int, int, Color?)
CanvasItem.DrawMesh(Mesh, Texture2D, Transform2D?, Color?)
CanvasItem.DrawMultimesh(MultiMesh, Texture2D)
CanvasItem.DrawSetTransform(Vector2, float, Vector2?)
CanvasItem.DrawSetTransformMatrix(Transform2D)
CanvasItem.DrawAnimationSlice(double, double, double, double)
CanvasItem.DrawEndAnimation()
CanvasItem.GetTransform()
CanvasItem.GetGlobalTransform()
CanvasItem.GetGlobalTransformWithCanvas()
CanvasItem.GetViewportTransform()
CanvasItem.GetViewportRect()
CanvasItem.GetCanvasTransform()
CanvasItem.GetScreenTransform()
CanvasItem.GetLocalMousePosition()
CanvasItem.GetGlobalMousePosition()
CanvasItem.GetCanvas()
CanvasItem.GetCanvasLayerNode()
CanvasItem.GetWorld2D()
CanvasItem.SetInstanceShaderParameter(StringName, Variant)
CanvasItem.GetInstanceShaderParameter(StringName)
CanvasItem.SetNotifyLocalTransform(bool)
CanvasItem.IsLocalTransformNotificationEnabled()
CanvasItem.SetNotifyTransform(bool)
CanvasItem.TransformNotificationEnabled()
CanvasItem.ForceUpdateTransform()
CanvasItem.MakeCanvasPositionLocal(Vector2)
CanvasItem.MakeInputLocal(InputEvent)
CanvasItem.SetVisibilityLayerBit(uint, bool)
CanvasItem.GetVisibilityLayerBit(uint)
CanvasItem.EmitSignalDraw()
CanvasItem.EmitSignalVisibilityChanged()
CanvasItem.EmitSignalHidden()
CanvasItem.EmitSignalItemRectChanged()
CanvasItem.Visible
CanvasItem.Modulate
CanvasItem.SelfModulate
CanvasItem.ShowBehindParent
CanvasItem.TopLevel
CanvasItem.ClipChildren
CanvasItem.LightMask
CanvasItem.VisibilityLayer
CanvasItem.ZIndex
CanvasItem.ZAsRelative
CanvasItem.YSortEnabled
CanvasItem.TextureFilter
CanvasItem.TextureRepeat

CanvasItem.Material
CanvasItem.UseParentMaterial
CanvasItem.Draw
CanvasItem.VisibilityChanged
CanvasItem.Hidden
CanvasItem.ItemRectChanged
Node.NotificationEnterTree
Node.NotificationExitTree
Node.NotificationMovedInParent
Node.NotificationReady
Node.NotificationPaused
Node.NotificationUnpaused
Node.NotificationPhysicsProcess
Node.NotificationProcess
Node.NotificationParented
Node.NotificationUnparented
Node.NotificationSceneInstantiated
Node.NotificationDragBegin
Node.NotificationDragEnd
Node.NotificationPathRenamed
Node.NotificationChildOrderChanged
Node.NotificationInternalProcess
Node.NotificationInternalPhysicsProcess
Node.NotificationPostEnterTree
Node.NotificationDisabled
Node.NotificationEnabled
Node.NotificationResetPhysicsInterpolation
Node.NotificationEditorPreSave
Node.NotificationEditorPostSave
Node.NotificationWMMouseEnter
Node.NotificationWMMouseExit
Node.NotificationWMWindowFocusIn
Node.NotificationWMWindowFocusOut
Node.NotificationWMCloseRequest
Node.NotificationWMGoBackRequest
Node.NotificationWMSizeChanged
Node.NotificationWMDpiChange
Node.NotificationVpMouseEnter
Node.NotificationVpMouseExit
Node.NotificationWMPositionChanged
Node.NotificationOsMemoryWarning
Node.NotificationTranslationChanged
Node.NotificationWMAbout
Node.NotificationCrash
Node.NotificationOslmeUpdate
Node.NotificationApplicationResumed
Node.NotificationApplicationPaused
Node.NotificationApplicationFocusIn
Node.NotificationApplicationFocusOut
Node.NotificationTextServerChanged

Node.GetNode<T>(NodePath)
Node.GetNodeOrNull<T>(NodePath)
Node.GetChild<T>(int, bool)
Node.GetChildOrNull<T>(int, bool)
Node.GetOwner<T>()
Node.GetOwnerOrNull<T>()
Node.GetParent<T>()
Node.GetParentOrNull<T>()
Node._EnterTree()
Node._ExitTree()
Node._GetConfigurationWarnings()
Node._Input(InputEvent)
Node._PhysicsProcess(double)
Node._Process(double)
Node._Ready()
Node._ShortcutInput(InputEvent)
Node._UnhandledInput(InputEvent)
Node._UnhandledKeyInput(InputEvent)
Node.PrintOrphanNodes()
Node.AddSibling(Node, bool)
Node.AddChild(Node, bool, Node.InternalMode)
Node.RemoveChild(Node)
Node.Reparent(Node, bool)
Node.GetChildCount(bool)
Node.GetChildren(bool)
Node.GetChild(int, bool)
Node.HasNode(NodePath)
Node.GetNode(NodePath)
Node.GetNodeOrNull(NodePath)
Node.GetParent()
Node.FindChild(string, bool, bool)
Node.FindChildren(string, string, bool, bool)
Node.FindParent(string)
Node.HasNodeAndResource(NodePath)
Node.GetNodeAndResource(NodePath)
Node.IsInsideTree()
Node.IsPartOfEditedScene()
Node.IsAncestorOf(Node)
Node.IsGreater Than(Node)
Node.GetPath()
Node.GetPathTo(Node, bool)
Node.AddToGroup(StringName, bool)
Node.RemoveFromGroup(StringName)
Node.IsInGroup(StringName)
Node.MoveChild(Node, int)
Node.GetGroups()
Node.GetIndex(bool)
Node.PrintTree()
Node.PrintTreePretty()
Node.GetTreeString()

Node.GetTreeStringPretty()
Node.PropagateNotification(int)
Node.PropagateCall(StringName, Array, bool)
Node.SetPhysicsProcess(bool)
Node.GetPhysicsProcessDeltaTime()
Node.IsPhysicsProcessing()
Node.GetProcessDeltaTime()
Node.SetProcess(bool)
Node.IsProcessing()
Node.SetProcessInput(bool)
Node.IsProcessingInput()
Node.SetProcessShortcutInput(bool)
Node.IsProcessingShortcutInput()
Node.SetProcessUnhandledInput(bool)
Node.IsProcessingUnhandledInput()
Node.SetProcessUnhandledKeyInput(bool)
Node.IsProcessingUnhandledKeyInput()
Node.CanProcess()
Node.SetDisplayFolded(bool)
Node.IsDisplayedFolded()
Node.SetProcessInternal(bool)
Node.IsProcessingInternal()
Node.SetPhysicsProcessInternal(bool)
Node.IsPhysicsProcessingInternal()
Node.IsPhysicsInterpolated()
Node.IsPhysicsInterpolatedAndEnabled()
Node.ResetPhysicsInterpolation()
Node.SetTranslationDomainInherited()
Node.GetWindow()
Node.GetLastExclusiveWindow()
Node.GetTree()
Node.CreateTween()
Node.Duplicate(int)
Node.ReplaceBy(Node, bool)
Node.setSceneInstanceLoadPlaceholder(bool)
Node.GetSceneInstanceLoadPlaceholder()
Node.setEditableInstance(Node, bool)
Node.setEditableInstance(Node)
Node.GetViewport()
Node.QueueFree()
Node.RequestReady()
Node.IsNodeReady()
Node.SetMultiplayerAuthority(int, bool)
Node.GetMultiplayerAuthority()
Node.IsMultiplayerAuthority()
Node.RpcConfig(StringName, Variant)
Node.GetRpcConfig()
Node.Atr(string, StringName)
Node.AtrN(string, StringName, int, StringName)
Node.Rpc(StringName, params Variant[]])

Node.Rpc(StringName, ReadOnlySpan<Variant>)
Node.RpcId(long, StringName, params Variant[])
Node.RpcId(long, StringName, ReadOnlySpan<Variant>)
Node.UpdateConfigurationWarnings()
Node.CallDeferredThreadGroup(StringName, params Variant[])
Node.CallDeferredThreadGroup(StringName, ReadOnlySpan<Variant>)
Node.SetDeferredThreadGroup(StringName, Variant)
Node.NotifyDeferredThreadGroup(int)
Node.CallThreadSafe(StringName, params Variant[])
Node.CallThreadSafe(StringName, ReadOnlySpan<Variant>)
Node.SetThreadSafe(StringName, Variant)
Node.NotifyThreadSafe(int)
Node.EmitSignalReady()
Node.EmitSignalRenamed()
Node.EmitSignalTreeEntered()
Node.EmitSignalTreeExiting()
Node.EmitSignalTreeExited()
Node.EmitSignalChildEnteredTree(Node)
Node.EmitSignalChildExitingTree(Node)
Node.EmitSignalChildOrderChanged()
Node.EmitSignalReplacingBy(Node)
Node.EmitSignalEditorDescriptionChanged(Node)
Node.EmitSignalEditorStateChanged()
Node.Name
Node.UniqueNameInOwner
Node.SceneFilePath
Node.Owner
Node.Multiplayer
Node.ProcessMode
Node.ProcessPriority
Node.ProcessPhysicsPriority
Node.ProcessThreadGroup
Node.ProcessThreadGroupOrder
Node.ProcessThreadMessages
Node.PhysicsInterpolationMode
Node.AutoTranslateMode
Node.EditorDescription
Node.Ready
Node.Renamed
Node.TreeEntered
Node.TreeExiting
Node.TreeExited
Node.ChildEnteredTree
Node.ChildExitingTree
Node.ChildOrderChanged
Node.ReplacingBy
Node.EditorDescriptionChanged
Node.EditorStateChanged
GodotObject.NotificationPostinitialize
GodotObject.NotificationPredelete

GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsInstanceIdValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)
GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)
GodotObject.Callv(StringName, Array)

GodotObject.HasMethod(StringName)
GodotObject.GetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
Namespace: [cfGodotEngine.Controls](#)
Assembly: CatSweeper.dll

Syntax

```
[Tool]
[GlobalClass]
[ScriptPath("res://Modules/cfGodotEngine/atlaspack/controls/AtlasTextureButton.cs")]
public class AtlasTextureButton : TextureButton, IDisposable
```

Implements

[IDisposable](#)

Extension Methods

[NodeUtil.DontDestroyOnLoad\(Node\)](#)

© Bus Fighter. All rights reserved.

Class `AtlasTextureButton.MethodName`

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

`object` → `GodotObject.MethodName` → `Node.MethodName` → `CanvasItem.MethodName` → `Control.MethodName` → `BaseButton.MethodName` → `TextureButton.MethodName` → `AtlasTextureButton.MethodName`

Inherited Members

`TextureButton.MethodName.SetTextureNormal`
`TextureButton.MethodName.SetTexturePressed`
`TextureButton.MethodName.SetTextureHover`
`TextureButton.MethodName.SetTextureDisabled`
`TextureButton.MethodName.SetTextureFocused`
`TextureButton.MethodName.SetClickMask`
`TextureButton.MethodName.SetIgnoreTextureSize`
`TextureButton.MethodName.SetStretchMode`
`TextureButton.MethodName.SetFlipH`
`TextureButton.MethodName.IsFlippedH`
`TextureButton.MethodName.SetFlipV`
`TextureButton.MethodName.IsFlippedV`
`TextureButton.MethodName.GetTextureNormal`
`TextureButton.MethodName.GetTexturePressed`
`TextureButton.MethodName.GetTextureHover`
`TextureButton.MethodName.GetTextureDisabled`
`TextureButton.MethodName.GetTextureFocused`
`TextureButton.MethodName.GetClickMask`
`TextureButton.MethodName.GetIgnoreTextureSize`
`TextureButton.MethodName.GetStretchMode`
`BaseButton.MethodName._Pressed`
`BaseButton.MethodName._Toggled`
`BaseButton.MethodName.SetPressed`
`BaseButton.MethodName.IsPressed`
`BaseButton.MethodName.SetPressedNoSignal`
`BaseButton.MethodName.IsHovered`
`BaseButton.MethodName.SetToggleMode`
`BaseButton.MethodName.IsToggleMode`
`BaseButton.MethodName.SetShortcutInTooltip`
`BaseButton.MethodName.IsShortcutInTooltipEnabled`
`BaseButton.MethodName.SetDisabled`
`BaseButton.MethodName.IsEnabled`
`BaseButton.MethodName.setActionMode`

BaseButton.MethodName.GetActionMode
BaseButton.MethodName.SetButtonMask
BaseButton.MethodName.GetButtonMask
BaseButton.MethodName.GetDrawMode
BaseButton.MethodName.SetKeepPressedOutside
BaseButton.MethodName.IsKeepPressedOutside
BaseButton.MethodName.SetShortcutFeedback
BaseButton.MethodName.IsShortcutFeedback
BaseButton.MethodName.SetShortcut
BaseButton.MethodName.GetShortcut
BaseButton.MethodName.SetButtonGroup
BaseButton.MethodName.GetButtonGroup
Control.MethodName._CanDropData
Control.MethodName._DropData
Control.MethodName._GetDragData
Control.MethodName._GetMinimumSize
Control.MethodName._GetTooltip
Control.MethodName._Guilinput
Control.MethodName._HasPoint
Control.MethodName._MakeCustomTooltip
Control.MethodName._StructuredTextParser
Control.MethodName.AcceptEvent
Control.MethodName.GetMinimumSize
Control.MethodName.GetCombinedMinimumSize
Control.MethodName._SetLayoutMode
Control.MethodName._GetLayoutMode
Control.MethodName._SetAnchorsLayoutPreset
Control.MethodName._GetAnchorsLayoutPreset
Control.MethodName.SetAnchorsPreset
Control.MethodName.SetOffsetsPreset
Control.MethodName.SetAnchorsAndOffsetsPreset
Control.MethodName._SetAnchor
Control.MethodName.SetAnchor
Control.MethodName.GetAnchor
Control.MethodName.SetOffset
Control.MethodName.GetOffset
Control.MethodName.SetAnchorAndOffset
Control.MethodName.SetBegin
Control.MethodName.SetEnd
Control.MethodName.SetPosition
Control.MethodName._SetPosition
Control.MethodName.SetSize
Control.MethodName.ResetSize
Control.MethodName._SetSize
Control.MethodName.SetCustomMinimumSize
Control.MethodName.SetGlobalPosition
Control.MethodName._SetGlobalPosition
Control.MethodName.SetRotation
Control.MethodName.SetRotationDegrees
Control.MethodName.setScale

Control.MethodName.SetPivotOffset
Control.MethodName.GetBegin
Control.MethodName.GetEnd
Control.MethodNameGetPosition
Control.MethodName.GetSize
Control.MethodName.GetRotation
Control.MethodName.GetRotationDegrees
Control.MethodName.GetScale
Control.MethodName.GetPivotOffset
Control.MethodName.GetCustomMinimumSize
Control.MethodName.GetParentAreaSize
Control.MethodName.GetGlobalPosition
Control.MethodName.GetScreenPosition
Control.MethodName.GetRect
Control.MethodName.GetGlobalRect
Control.MethodName.SetFocusMode
Control.MethodName.GetFocusMode
Control.MethodName.HasFocus
Control.MethodName.GrabFocus
Control.MethodName.ReleaseFocus
Control.MethodName.FindPrevValidFocus
Control.MethodName.FindNextValidFocus
Control.MethodName.FindValidFocusNeighbor
Control.MethodName.SetHSizeFlags
Control.MethodName.GetHSizeFlags
Control.MethodName.SetStretchRatio
Control.MethodName.GetStretchRatio
Control.MethodName.SetVSizeFlags
Control.MethodName.GetVSizeFlags
Control.MethodName.SetTheme
Control.MethodName.GetTheme
Control.MethodName.SetThemeTypeVariation
Control.MethodName.GetThemeTypeVariation
Control.MethodName.BeginBulkThemeOverride
Control.MethodName.EndBulkThemeOverride
Control.MethodName.AddThemelconOverride
Control.MethodName.AddThemeStyleboxOverride
Control.MethodName.AddThemeFontOverride
Control.MethodName.AddThemeFontSizeOverride
Control.MethodName.AddThemeColorOverride
Control.MethodName.AddThemeConstantOverride
Control.MethodName.RemoveThemelconOverride
Control.MethodName.RemoveThemeStyleboxOverride
Control.MethodName.RemoveThemeFontOverride
Control.MethodName.RemoveThemeFontSizeOverride
Control.MethodName.RemoveThemeColorOverride
Control.MethodName.RemoveThemeConstantOverride
Control.MethodName.GetThemelcon
Control.MethodName.GetThemeStylebox
Control.MethodName.GetThemeFont

Control.MethodName.GetThemeFontSize
Control.MethodName.GetThemeColor
Control.MethodName.GetThemeConstant
Control.MethodName.HasThemelconOverride
Control.MethodName.HasThemeStyleboxOverride
Control.MethodName.HasThemeFontOverride
Control.MethodName.HasThemeFontSizeOverride
Control.MethodName.HasThemeColorOverride
Control.MethodName.HasThemeConstantOverride
Control.MethodName.HasThemelcon
Control.MethodName.HasThemeStylebox
Control.MethodName.HasThemeFont
Control.MethodName.HasThemeFontSize
Control.MethodName.HasThemeColor
Control.MethodName.HasThemeConstant
Control.MethodName.GetThemeDefaultBaseScale
Control.MethodName.GetThemeDefaultFont
Control.MethodName.GetThemeDefaultFontSize
Control.MethodName.GetParentControl
Control.MethodName.SetHGrowDirection
Control.MethodName.GetHGrowDirection
Control.MethodName.SetVGrowDirection
Control.MethodName.GetVGrowDirection
Control.MethodName.SetToolTipAutoTranslateMode
Control.MethodName.GetTooltipAutoTranslateMode
Control.MethodName.SetToolTipText
Control.MethodName.GetTooltipText
Control.MethodName.GetTooltip
Control.MethodName.SetDefaultCursorShape
Control.MethodName.GetDefaultCursorShape
Control.MethodName.GetCursorShape
Control.MethodName.SetFocusNeighbor
Control.MethodName.GetFocusNeighbor
Control.MethodName.SetFocusNext
Control.MethodName.GetFocusNext
Control.MethodName.SetFocusPrevious
Control.MethodName.GetFocusPrevious
Control.MethodName.ForceDrag
Control.MethodName.SetMouseFilter
Control.MethodName.GetMouseFilter
Control.MethodName.SetForcePassScrollEvents
Control.MethodName.IsForcePassScrollEvents
Control.MethodName.SetClipContents
Control.MethodName.IsClippingContents
Control.MethodName.GrabClickFocus
Control.MethodName.SetDragForwarding
Control.MethodName.SetDragPreview
Control.MethodName.IsDragSuccessful
Control.MethodName.WarpMouse
Control.MethodName.SetShortcutContext

Control.MethodName.GetShortcutContext
Control.MethodName.UpdateMinimumSize
Control.MethodName.SetLayoutDirection
Control.MethodName.GetLayoutDirection
Control.MethodName.IsLayoutRtl
Control.MethodName.SetAutoTranslate
Control.MethodName.IsAutoTranslating
Control.MethodName.SetLocalizeNumeralSystem
Control.MethodName.IsLocalizingNumeralSystem
CanvasItem.MethodName._Draw
CanvasItem.MethodName.GetCanvasItem
CanvasItem.MethodName.SetVisible
CanvasItem.MethodName.IsVisible
CanvasItem.MethodName.IsVisibleInTree
CanvasItem.MethodName.Show
CanvasItem.MethodName.Hide
CanvasItem.MethodName.QueueRedraw
CanvasItem.MethodName.MoveToFront
CanvasItem.MethodName.SetAsTopLevel
CanvasItem.MethodName.IsSetAsTopLevel
CanvasItem.MethodName.SetLightMask
CanvasItem.MethodName.GetLightMask
CanvasItem.MethodName.SetModulate
CanvasItem.MethodName.GetModulate
CanvasItem.MethodName.SetSelfModulate
CanvasItem.MethodName.GetSelfModulate
CanvasItem.MethodName.SetZIndex
CanvasItem.MethodName.GetZIndex
CanvasItem.MethodName.SetZAsRelative
CanvasItem.MethodName.IsZRelative
CanvasItem.MethodName.SetYSortEnabled
CanvasItem.MethodName.IsYSortEnabled
CanvasItem.MethodName.SetDrawBehindParent
CanvasItem.MethodName.IsDrawBehindParentEnabled
CanvasItem.MethodName.DrawLine
CanvasItem.MethodName.DrawDashedLine
CanvasItem.MethodName.DrawPolyline
CanvasItem.MethodName.DrawPolylineColors
CanvasItem.MethodName.DrawArc
CanvasItem.MethodName.DrawMultiline
CanvasItem.MethodName.DrawMultilineColors
CanvasItem.MethodName.DrawRect
CanvasItem.MethodName.DrawCircle
CanvasItem.MethodName.DrawTexture
CanvasItem.MethodName.DrawTextureRect
CanvasItem.MethodName.DrawTextureRectRegion
CanvasItem.MethodName.DrawMsdfTextureRectRegion
CanvasItem.MethodName.DrawLcdTextureRectRegion
CanvasItem.MethodName.DrawStyleBox
CanvasItem.MethodName.DrawPrimitive

CanvasItem.MethodName.DrawPolygon
CanvasItem.MethodName.DrawColoredPolygon
CanvasItem.MethodName.DrawString
CanvasItem.MethodName.DrawMultilineString
CanvasItem.MethodName.DrawStringOutline
CanvasItem.MethodName.DrawMultilineStringOutline
CanvasItem.MethodName.DrawChar
CanvasItem.MethodName.DrawCharOutline
CanvasItem.MethodName.DrawMesh
CanvasItem.MethodName.DrawMultimesh
CanvasItem.MethodName.DrawSetTransform
CanvasItem.MethodName.DrawSetTransformMatrix
CanvasItem.MethodName.DrawAnimationSlice
CanvasItem.MethodName.DrawEndAnimation
CanvasItem.MethodName.GetTransform
CanvasItem.MethodName.GetGlobalTransform
CanvasItem.MethodName.GetGlobalTransformWithCanvas
CanvasItem.MethodName.GetViewportTransform
CanvasItem.MethodName.GetViewportRect
CanvasItem.MethodName.GetCanvasTransform
CanvasItem.MethodName.GetScreenTransform
CanvasItem.MethodName.GetLocalMousePosition
CanvasItem.MethodName.GetGlobalMousePosition
CanvasItem.MethodName.GetCanvas
CanvasItem.MethodName.GetCanvasLayerNode
CanvasItem.MethodName.GetWorld2D
CanvasItem.MethodName.SetMaterial
CanvasItem.MethodName.GetMaterial
CanvasItem.MethodName.SetInstanceShaderParameter
CanvasItem.MethodName.GetInstanceShaderParameter
CanvasItem.MethodName.SetUseParentMaterial
CanvasItem.MethodName.GetUseParentMaterial
CanvasItem.MethodName.SetNotifyLocalTransform
CanvasItem.MethodName.IsLocalTransformNotificationEnabled
CanvasItem.MethodName.SetNotifyTransform
CanvasItem.MethodName.IsTransformNotificationEnabled
CanvasItem.MethodName.ForceUpdateTransform
CanvasItem.MethodName.MakeCanvasPositionLocal
CanvasItem.MethodName.MakeInputLocal
CanvasItem.MethodName.SetVisibilityLayer
CanvasItem.MethodName.GetVisibilityLayer
CanvasItem.MethodName.SetVisibilityLayerBit
CanvasItem.MethodName.GetVisibilityLayerBit
CanvasItem.MethodName.SetTextureFilter
CanvasItem.MethodName.GetTextureFilter
CanvasItem.MethodName.SetTextureRepeat
CanvasItem.MethodName.GetTextureRepeat
CanvasItem.MethodName.SetClipChildrenMode
CanvasItem.MethodName.GetClipChildrenMode
Node.MethodName._EnterTree

Node.MethodName._ExitTree
Node.MethodName._GetConfigurationWarnings
Node.MethodName._Input
Node.MethodName._PhysicsProcess
Node.MethodName._Process
Node.MethodName._Ready
Node.MethodName._ShortcutInput
Node.MethodName._UnhandledInput
Node.MethodName._UnhandledKeyInput
Node.MethodName.PrintOrphanNodes
Node.MethodName.AddSibling
Node.MethodName.SetName
Node.MethodName.GetName
Node.MethodName.AddChild
Node.MethodName.RemoveChild
Node.MethodName.Reparent
Node.MethodName.GetChildCount
Node.MethodName.GetChildren
Node.MethodName.GetChild
Node.MethodName.HasNode
Node.MethodName.GetNode
Node.MethodName.GetNodeOrNull
Node.MethodName.GetParent
Node.MethodName.FindChild
Node.MethodName.FindChildren
Node.MethodName.FindParent
Node.MethodName.HasNodeAndResource
Node.MethodName.GetNodeAndResource
Node.MethodName.IsInsideTree
Node.MethodName.IsPartOfEditedScene
Node.MethodName.IsAncestorOf
Node.MethodName.IsGreaterThan
Node.MethodName.GetPath
Node.MethodName.GetPathTo
Node.MethodName.AddToGroup
Node.MethodName.RemoveFromGroup
Node.MethodName.IsInGroup
Node.MethodName.MoveChild
Node.MethodName.GetGroups
Node.MethodName.SetOwner
Node.MethodName.GetOwner
Node.MethodName.GetIndex
Node.MethodName.PrintTree
Node.MethodName.PrintTreePretty
Node.MethodName.GetTreeString
Node.MethodName.GetTreeStringPretty
Node.MethodName.SetSceneFilePath
Node.MethodName.GetSceneFilePath
Node.MethodName.PropagateNotification
Node.MethodName.PropagateCall

Node.MethodName.SetPhysicsProcess
Node.MethodName.GetPhysicsProcessDeltaTime
Node.MethodName.IsPhysicsProcessing
Node.MethodName.GetProcessDeltaTime
Node.MethodName.SetProcess
Node.MethodName.SetProcessPriority
Node.MethodName.GetProcessPriority
Node.MethodName.SetPhysicsProcessPriority
Node.MethodName.GetPhysicsProcessPriority
Node.MethodName.IsProcessing
Node.MethodName.SetProcessInput
Node.MethodName.IsProcessingInput
Node.MethodName.SetProcessShortcutInput
Node.MethodName.IsProcessingShortcutInput
Node.MethodName.SetProcessUnhandledInput
Node.MethodName.IsProcessingUnhandledInput
Node.MethodName.SetProcessUnhandledKeyInput
Node.MethodName.IsProcessingUnhandledKeyInput
Node.MethodName.SetProcessMode
Node.MethodName.GetProcessMode
Node.MethodName.CanProcess
Node.MethodName.SetProcessThreadGroup
Node.MethodName.GetProcessThreadGroup
Node.MethodName.SetProcessThreadMessages
Node.MethodName.GetProcessThreadMessages
Node.MethodName.SetProcessThreadGroupOrder
Node.MethodName.GetProcessThreadGroupOrder
Node.MethodName.SetDisplayFolded
Node.MethodName.IsDisplayedFolded
Node.MethodName.SetProcessInternal
Node.MethodName.IsProcessingInternal
Node.MethodName.SetPhysicsProcessInternal
Node.MethodName.IsPhysicsProcessingInternal
Node.MethodName.SetPhysicsInterpolationMode
Node.MethodName.GetPhysicsInterpolationMode
Node.MethodName.IsPhysicsInterpolated
Node.MethodName.IsPhysicsInterpolatedAndEnabled
Node.MethodName.ResetPhysicsInterpolation
Node.MethodName.SetAutoTranslateMode
Node.MethodName.GetAutoTranslateMode
Node.MethodName.SetTranslationDomainInherited
Node.MethodName.GetWindow
Node.MethodName.GetLastExclusiveWindow
Node.MethodName.GetTree
Node.MethodName.CreateTween
Node.MethodName.Duplicate
Node.MethodName.ReplaceBy
Node.MethodName.SetSceneInstanceLoadPlaceholder
Node.MethodName.GetSceneInstanceLoadPlaceholder
Node.MethodName.setEditableInstance

Node.MethodName.IsEditableInstance
Node.MethodName.GetViewport
Node.MethodName.QueueFree
Node.MethodName.RequestReady
Node.MethodName.IsNodeReady
Node.MethodName.SetMultiplayerAuthority
Node.MethodName.GetMultiplayerAuthority
Node.MethodName.IsMultiplayerAuthority
Node.MethodName.GetMultiplayer
Node.MethodName.RpcConfig
Node.MethodName.GetRpcConfig
Node.MethodName.SetEditorDescription
Node.MethodName.GetEditorDescription
Node.MethodName._SetImportPath
Node.MethodName._GetImportPath
Node.MethodName.SetUniqueNameInOwner
Node.MethodName.IsUniqueNameInOwner
Node.MethodName.Atr
Node.MethodName.AtrN
Node.MethodName.Rpc
Node.MethodName.RpcId
Node.MethodName.UpdateConfigurationWarnings
Node.MethodName.CallDeferredThreadGroup
Node.MethodName.SetDeferredThreadGroup
Node.MethodName.NotifyDeferredThreadGroup
Node.MethodName.CallThreadSafe
Node.MethodName.SetThreadSafe
Node.MethodName.NotifyThreadSafe
GodotObject.MethodName._Get
GodotObject.MethodName._GetPropertyList
GodotObject.MethodName._IterGet
GodotObject.MethodName._IterInit
GodotObject.MethodName._IterNext
GodotObject.MethodName._Notification
GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification

GodotObject.MethodName.GetInstanceld
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr
GodotObject.MethodName.TrN
GodotObject.MethodName.GetTranslationDomain
GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfGodotEngine.Controls](#)

Assembly: CatSweeper.dll

Syntax

```
public class AtlasTextureButton.MethodName : TextureButton.MethodName
```

© Bus Fighter. All rights reserved.

Class AtlasTextureButton.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.PropertyName](#) → [Node.PropertyName](#) → [CanvasItem.PropertyName](#) → [Control.PropertyName](#) → [BaseButton.PropertyName](#) → [TextureButton.PropertyName](#) → [AtlasTextureButton.PropertyName](#)

Inherited Members

TextureButton.PropertyName.TextureNormal
TextureButton.PropertyName.TexturePressed
TextureButton.PropertyName.TextureHover
TextureButton.PropertyName.TextureDisabled
TextureButton.PropertyName.TextureFocused
TextureButton.PropertyName.TextureClickMask
TextureButton.PropertyName.IgnoreTextureSize
TextureButton.PropertyName.StretchMode
TextureButton.PropertyName.FlipH
TextureButton.PropertyName.FlipV
BaseButton.PropertyName.Disabled
BaseButton.PropertyName.ToggleMode
BaseButton.PropertyName.ButtonPressed
BaseButton.PropertyName.ActionMode
BaseButton.PropertyName.ButtonMask
BaseButton.PropertyName.KeepPressedOutside
BaseButton.PropertyName.ButtonGroup
BaseButton.PropertyName.Shortcut
BaseButton.PropertyName.ShortcutFeedback
BaseButton.PropertyName.ShortcutInTooltip
Control.PropertyName.ClipContents
Control.PropertyName.CustomMinimumSize
Control.PropertyName.LayoutDirection
Control.PropertyName.LayoutMode
Control.PropertyName.AnchorsPreset
Control.PropertyName.AnchorLeft
Control.PropertyName.AnchorTop
Control.PropertyName.AnchorRight
Control.PropertyName.AnchorBottom
Control.PropertyName.OffsetLeft
Control.PropertyName.OffsetTop
Control.PropertyName.OffsetRight
Control.PropertyName.OffsetBottom

Control.PropertyName.GrowHorizontal
Control.PropertyName.GrowVertical
Control.PropertyName.Size
Control.PropertyName.Position
Control.PropertyName.GlobalPosition
Control.PropertyName.Rotation
Control.PropertyName.RotationDegrees
Control.PropertyName.Scale
Control.PropertyName.PivotOffset
Control.PropertyName.SizeTypeHorizontal
Control.PropertyName.SizeTypeVertical
Control.PropertyName.SizeTypeStretchRatio
Control.PropertyName.LocalizeNumeralSystem
Control.PropertyName.AutoTranslate
Control.PropertyName.TooltipText
Control.PropertyName.TooltipAutoTranslateMode
Control.PropertyName.FocusNeighborLeft
Control.PropertyName.FocusNeighborTop
Control.PropertyName.FocusNeighborRight
Control.PropertyName.FocusNeighborBottom
Control.PropertyName.FocusNext
Control.PropertyName.FocusPrevious
Control.PropertyName.FocusMode
Control.PropertyName.MouseFilter
Control.PropertyName.MouseForcePassScrollEvents
Control.PropertyName.MouseDefaultCursorShape
Control.PropertyName.ShortcutContext
Control.PropertyName.Theme
Control.PropertyName.ThemeTypeVariation
CanvasItem.PropertyName.Visible
CanvasItem.PropertyName.Modulate
CanvasItem.PropertyName.SelfModulate
CanvasItem.PropertyName.ShowBehindParent
CanvasItem.PropertyName.TopLevel
CanvasItem.PropertyName.ClipChildren
CanvasItem.PropertyName.LightMask
CanvasItem.PropertyName.VisibilityLayer
CanvasItem.PropertyName.ZIndex
CanvasItem.PropertyName.ZAsRelative
CanvasItem.PropertyName.YSortEnabled
CanvasItem.PropertyName.TextureFilter
CanvasItem.PropertyName.TextureRepeat
CanvasItem.PropertyName.Material
CanvasItem.PropertyName.UseParentMaterial
Node.PropertyName._ImportPath
Node.PropertyName.Name
Node.PropertyName.UniqueNameInOwner
Node.PropertyName.SceneFilePath
Node.PropertyName.Owner
Node.PropertyName.Multiplayer

Node.PropertyName.ProcessMode
Node.PropertyName.ProcessPriority
Node.PropertyName.ProcessPhysicsPriority
Node.PropertyName.ProcessThreadGroup
Node.PropertyName.ProcessThreadGroupOrder
Node.PropertyName.ProcessThreadMessages
Node.PropertyName.PhysicsInterpolationMode
Node.PropertyName.AutoTranslateMode
Node.PropertyName.EditorDescription
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()
Namespace: **cfGodotEngine.Controls**
Assembly: CatSweeper.dll

Syntax

```
public class AtlasTextureButton.PropertyName : TextureButton.PropertyName
```

© Bus Fighter. All rights reserved.

Class `AtlasTextureButton.SignalName`

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

`object` → `GodotObject.SignalName` → `Node.SignalName` → `CanvasItem.SignalName` → `Control.SignalName` → `BaseButton.SignalName` → `TextureButton.SignalName` → `AtlasTextureButton.SignalName`

Inherited Members

`BaseButton.SignalName.Pressed`
`BaseButton.SignalName.ButtonUp`
`BaseButton.SignalNameButtonDown`
`BaseButton.SignalName.Toggled`
`Control.SignalName.Resized`
`Control.SignalName.GuiInput`
`Control.SignalName.MouseEntered`
`Control.SignalName.MouseExited`
`Control.SignalName.FocusEntered`
`Control.SignalName.FocusExited`
`Control.SignalName.SizeFlagsChanged`
`Control.SignalName.MinimumSizeChanged`
`Control.SignalName.ThemeChanged`
`CanvasItem.SignalName.Draw`
`CanvasItem.SignalName.VisibilityChanged`
`CanvasItem.SignalName.Hidden`
`CanvasItem.SignalName.ItemRectChanged`
`Node.SignalName.Ready`
`Node.SignalName.Renamed`
`Node.SignalName.TreeEntered`
`Node.SignalName.TreeExiting`
`Node.SignalName.TreeExited`
`Node.SignalName.ChildEnteredTree`
`Node.SignalName.ChildExitingTree`
`Node.SignalName.ChildOrderChanged`
`Node.SignalNameReplacingBy`
`Node.SignalName.EditorDescriptionChanged`
`Node.SignalName.EditorStateChanged`
`GodotObject.SignalName.ScriptChanged`
`GodotObject.SignalName.PropertyListChanged`
[`object.Equals\(object\)`](#)
[`object.Equals\(object, object\)`](#)
[`object.GetHashCode\(\)`](#)

object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfGodotEngine.Controls](#)

Assembly: CatSweeper.dll

Syntax

```
public class AtlasTextureButton.SignalName : TextureButton.SignalName
```

© Bus Fighter. All rights reserved.

Class AtlasTextureRect

Inheritance

[object](#) → [GodotObject](#) → [Node](#) → [CanvasItem](#) → [Control](#) → [TextureRect](#) → [AtlasTextureRect](#)

Implements

[IDisposable](#)

Inherited Members

[TextureRect.InvokeGodotClassMethod\(in godot_string_name, NativeVariantPtrArgs, out godot_variant\)](#)

[TextureRect.HasGodotClassMethod\(in godot_string_name\)](#)

[TextureRect.HasGodotClassSignal\(in godot_string_name\)](#)

[TextureRect.Texture](#)

[TextureRect.ExpandMode](#)

[TextureRect.StretchMode](#)

[TextureRect.FlipH](#)

[TextureRect.FlipV](#)

[Control.NotificationResized](#)

[Control.NotificationMouseEnter](#)

[Control.NotificationMouseExit](#)

[Control.NotificationMouseEnterSelf](#)

[Control.NotificationMouseExitSelf](#)

[Control.NotificationFocusEnter](#)

[Control.NotificationFocusExit](#)

[Control.NotificationThemeChanged](#)

[Control.NotificationScrollBegin](#)

[Control.NotificationScrollEnd](#)

[Control.NotificationLayoutDirectionChanged](#)

[Control._CanDropData\(Vector2, Variant\)](#)

[Control._DropData\(Vector2, Variant\)](#)

[Control._GetDragData\(Vector2\)](#)

[Control._GetMinimumSize\(\)](#)

[Control._GetTooltip\(Vector2\)](#)

[Control._GUILInput\(InputEvent\)](#)

[Control._HasPoint\(Vector2\)](#)

[**Control._MakeCustomTooltip\(string\)**](#)

[**Control._StructuredTextParser\(Array, string\)**](#)

[Control.AcceptEvent\(\)](#)

[Control.GetMinimumSize\(\)](#)

[Control.GetCombinedMinimumSize\(\)](#)

[Control.SetAnchorsPreset\(Control.LayoutPreset, bool\)](#)

Control.SetOffsetsPreset(Control.LayoutPreset, Control.LayoutPresetMode, int)
Control.SetAnchorsAndOffsetsPreset(Control.LayoutPreset, Control.LayoutPresetMode, int)
Control.SetAnchor(Side, float, bool, bool)
Control.SetAnchorAndOffset(Side, float, float, bool)
Control.SetBegin(Vector2)
Control.SetEnd(Vector2)
Control.SetPosition(Vector2, bool)
Control.SetSize(Vector2, bool)
Control.ResetSize()
Control.SetGlobalPosition(Vector2, bool)
Control.GetBegin()
Control.GetEnd()
Control.GetParentAreaSize()
Control.GetScreenPosition()
Control.GetRect()
Control.GetGlobalRect()
Control.HasFocus()
Control.GrabFocus()
Control.ReleaseFocus()
Control.FindPrevValidFocus()
Control.FindNextValidFocus()
Control.FindValidFocusNeighbor(Side)
Control.BeginBulkThemeOverride()
Control.EndBulkThemeOverride()
Control.AddThemelconOverride(StringName, Texture2D)
Control.AddThemeStyleboxOverride(StringName, StyleBox)
Control.AddThemeFontOverride(StringName, Font)
Control.AddThemeFontSizeOverride(StringName, int)
Control.AddThemeColorOverride(StringName, Color)
Control.AddThemeConstantOverride(StringName, int)
Control.RemoveThemelconOverride(StringName)
Control.RemoveThemeStyleboxOverride(StringName)
Control.RemoveThemeFontOverride(StringName)
Control.RemoveThemeFontSizeOverride(StringName)
Control.RemoveThemeColorOverride(StringName)
Control.RemoveThemeConstantOverride(StringName)
Control.GetThemelcon(StringName, StringName)
Control.GetThemeStylebox(StringName, StringName)
Control.GetThemeFont(StringName, StringName)
Control.GetThemeFontSize(StringName, StringName)
Control.GetThemeColor(StringName, StringName)
Control.GetThemeConstant(StringName, StringName)
Control.HasThemelconOverride(StringName)
Control.HasThemeStyleboxOverride(StringName)
Control.HasThemeFontOverride(StringName)
Control.HasThemeFontSizeOverride(StringName)
Control.HasThemeColorOverride(StringName)
Control.HasThemeConstantOverride(StringName)
Control.HasThemelcon(StringName, StringName)
Control.HasThemeStylebox(StringName, StringName)

Control.HasThemeFont(StringName, StringName)
Control.HasThemeFontSize(StringName, StringName)
Control.HasThemeColor(StringName, StringName)
Control.HasThemeConstant(StringName, StringName)
Control.GetThemeDefaultBaseScale()
Control.GetThemeDefaultFont()
Control.GetThemeDefaultFontSize()
Control.GetParentControl()
Control.GetTooltip(Vector2?)
Control.GetCursorShape(Vector2?)
Control.ForceDrag(Variant, Control)
Control.GrabClickFocus()
Control.SetDragForwarding(Callable, Callable, Callable)
Control.SetDragPreview(Control)
Control.IsDragSuccessful()
Control.WarpMouse(Vector2)
Control.UpdateMinimumSize()
Control.IsLayoutRtl()
Control.EmitSignalResized()
Control.EmitSignalGuilInput(InputEvent)
Control.EmitSignalMouseEntered()
Control.EmitSignalMouseExited()
Control.EmitSignalFocusEntered()
Control.EmitSignalFocusExited()
Control.EmitSignalSizeFlagsChanged()
Control.EmitSignalMinimumSizeChanged()
Control.EmitSignalThemeChanged()
Control.ClipContents
Control.CustomMinimumSize
Control.LayoutDirection
Control.AnchorLeft
Control.AnchorTop
Control.AnchorRight
Control.AnchorBottom
Control.OffsetLeft
Control.OffsetTop
Control.OffsetRight
Control.OffsetBottom
Control.GrowHorizontal
Control.GrowVertical
Control.Size
Control.Position
Control.GlobalPosition
Control.Rotation
Control.RotationDegrees
Control.Scale
Control.PivotOffset
Control.SizeFlagsHorizontal
Control.SizeFlagsVertical
Control.SizeFlagsStretchRatio

Control.LocalizeNumeralSystem
Control.AutoTranslate
Control.TooltipText
Control.TooltipAutoTranslateMode
Control.FocusNeighborLeft
Control.FocusNeighborTop
Control.FocusNeighborRight
Control.FocusNeighborBottom
Control.FocusNext
Control.FocusPrevious
Control.FocusMode
Control.MouseFilter
Control.MouseForcePassScrollEvents
Control.MouseDefaultCursorShape
Control.ShortcutContext
Control.Theme
Control.ThemeTypeVariation
Control.Resized
Control.GuiInput
Control.MouseEntered
Control.MouseExited
Control.FocusEntered
Control.FocusExited
Control.SizeFlagsChanged
Control.MinimumSizeChanged
Control.ThemeChanged
CanvasItem.NotificationTransformChanged
CanvasItem.NotificationLocalTransformChanged
CanvasItem.NotificationDraw
CanvasItem.NotificationVisibilityChanged
CanvasItem.NotificationEnterCanvas
CanvasItem.NotificationExitCanvas
CanvasItem.NotificationWorld2DChanged
CanvasItem._Draw()
CanvasItem.GetCanvasItem()
CanvasItem.IsVisibleInTree()
CanvasItem.Show()
CanvasItem.Hide()
CanvasItem.QueueRedraw()
CanvasItem.MoveToFront()
CanvasItem.DrawLine(Vector2, Vector2, Color, float, bool)
CanvasItem.DrawDashedLine(Vector2, Vector2, Color, float, float, bool, bool)
CanvasItem.DrawLine(ReadOnlySpan<Vector2>, Color, float, bool)
CanvasItem.DrawLine(ReadOnlySpan<Vector2>, Color, float, bool)
CanvasItem.DrawLineColors(ReadOnlySpan<Vector2>, Color[], float, bool)
CanvasItem.DrawLineColors(ReadOnlySpan<Vector2>, ReadOnlySpan<Color>, float, bool)
CanvasItem.DrawArc(Vector2, float, float, float, int, Color, float, bool)
CanvasItem.DrawMultiline(Vector2[], Color, float, bool)
CanvasItem.DrawMultiline(ReadOnlySpan<Vector2>, Color, float, bool)
CanvasItem.DrawMultilineColors(Vector2[], Color[], float, bool)

CanvasItem.DrawMultilineColors(ReadOnlySpan<Vector2>, ReadOnlySpan<Color>, float, bool)
CanvasItem.DrawRect(Rect2, Color, bool, float, bool)
CanvasItem.DrawCircle(Vector2, float, Color, bool, float, bool)
CanvasItem.DrawTexture(Texture2D, Vector2, Color?)
CanvasItem.DrawTextureRect(Texture2D, Rect2, bool, Color?, bool)
CanvasItem.DrawTextureRectRegion(Texture2D, Rect2, Rect2, Color?, bool, bool)
CanvasItem.DrawMsdfTextureRectRegion(Texture2D, Rect2, Rect2, Color?, double, double, double)
CanvasItem.DrawLcdTextureRectRegion(Texture2D, Rect2, Rect2, Color?)
CanvasItem.DrawStyleBox(StyleBox, Rect2)
CanvasItem.DrawPrimitive(Vector2[], Color[], Vector2[], Texture2D)
CanvasItem.DrawPrimitive(ReadOnlySpan<Vector2>, ReadOnlySpan<Color>, ReadOnlySpan<Vector2>, Texture2D)
CanvasItem.DrawPolygon(Vector2[], Color[], Vector2[], Texture2D)
CanvasItem.DrawPolygon(ReadOnlySpan<Vector2>, ReadOnlySpan<Color>, ReadOnlySpan<Vector2>, Texture2D)
CanvasItem.DrawColoredPolygon(Vector2[], Color, Vector2[], Texture2D)
CanvasItem.DrawColoredPolygon(ReadOnlySpan<Vector2>, Color, ReadOnlySpan<Vector2>, Texture2D)
CanvasItem.DrawString(Font, Vector2, string, HorizontalAlignment, float, int, Color?, TextServer.JustificationFlag, TextServer.Direction, TextServer.Orientation)
CanvasItem.DrawString(Font, Vector2, string, HorizontalAlignment, float, int, int, Color?, TextServer.LineBreakFlag, TextServer.JustificationFlag, TextServer.Direction, TextServer.Orientation)
CanvasItem.DrawStringOutline(Font, Vector2, string, HorizontalAlignment, float, int, int, Color?, TextServer.JustificationFlag, TextServer.Direction, TextServer.Orientation)
CanvasItem.DrawStringOutline(Font, Vector2, string, HorizontalAlignment, float, int, int, int, Color?, TextServer.LineBreakFlag, TextServer.JustificationFlag, TextServer.Direction, TextServer.Orientation)
CanvasItem.DrawChar(Font, Vector2, string, int, Color?)
CanvasItem.DrawCharOutline(Font, Vector2, string, int, int, Color?)
CanvasItem.DrawMesh(Mesh, Texture2D, Transform2D?, Color?)
CanvasItem.DrawMultimesh(MultiMesh, Texture2D)
CanvasItem.DrawSetTransform(Vector2, float, Vector2?)
CanvasItem.DrawSetTransformMatrix(Transform2D)
CanvasItem.DrawAnimationSlice(double, double, double, double)
CanvasItem.DrawEndAnimation()
CanvasItem.GetTransform()
CanvasItem.GetGlobalTransform()
CanvasItem.GetGlobalTransformWithCanvas()
CanvasItem.GetViewportTransform()
CanvasItem.GetViewportRect()
CanvasItem.GetCanvasTransform()
CanvasItem.GetScreenTransform()
CanvasItem.GetLocalMousePosition()
CanvasItem.GetGlobalMousePosition()
CanvasItem.GetCanvas()
CanvasItem.GetCanvasLayerNode()
CanvasItem.GetWorld2D()
CanvasItem.SetInstanceShaderParameter(StringName, Variant)
CanvasItem.GetInstanceShaderParameter(StringName)
CanvasItem.SetNotifyLocalTransform(bool)
CanvasItem.IsLocalTransformNotificationEnabled()
CanvasItem.SetNotifyTransform(bool)
CanvasItem.TransformNotificationEnabled()
CanvasItem.ForceUpdateTransform()

CanvasItem.MakeCanvasPositionLocal(Vector2)
CanvasItem.MakeInputLocal(InputEvent)
CanvasItem.SetVisibilityLayerBit(uint, bool)
CanvasItem.GetVisibilityLayerBit(uint)
CanvasItem.EmitSignalDraw()
CanvasItem.EmitSignalVisibilityChanged()
CanvasItem.EmitSignalHidden()
CanvasItem.EmitSignalItemRectChanged()
CanvasItem.Visible
CanvasItem.Modulate
CanvasItem.SelfModulate
CanvasItem.ShowBehindParent
CanvasItem.TopLevel
CanvasItem.ClipChildren
CanvasItem.LightMask
CanvasItem.VisibilityLayer
CanvasItem.ZIndex
CanvasItem.ZAsRelative
CanvasItem.YSortEnabled
CanvasItem.TextureFilter
CanvasItem.TextureRepeat
CanvasItem.Material
CanvasItem.UseParentMaterial
CanvasItem.Draw
CanvasItem.VisibilityChanged
CanvasItem.Hidden
CanvasItem.ItemRectChanged
Node.NotificationEnterTree
Node.NotificationExitTree
Node.NotificationMovedInParent
Node.NotificationReady
Node.NotificationPaused
Node.NotificationUnpaused
Node.NotificationPhysicsProcess
Node.NotificationProcess
Node.NotificationParented
Node.NotificationUnparented
Node.NotificationSceneInstantiated
Node.NotificationDragBegin
Node.NotificationDragEnd
Node.NotificationPathRenamed
Node.NotificationChildOrderChanged
Node.NotificationInternalProcess
Node.NotificationInternalPhysicsProcess
Node.NotificationPostEnterTree
Node.NotificationDisabled
Node.NotificationEnabled
Node.NotificationResetPhysicsInterpolation
Node.NotificationEditorPreSave
Node.NotificationEditorPostSave

Node.NotificationWMMouseEnter
Node.NotificationWMMouseExit
Node.NotificationWMWindowFocusIn
Node.NotificationWMWindowFocusOut
Node.NotificationWMCloseRequest
Node.NotificationWMGoBackRequest
Node.NotificationWMSizeChanged
Node.NotificationWMDpiChange
Node.NotificationVpMouseEnter
Node.NotificationVpMouseExit
Node.NotificationWMPositionChanged
Node.NotificationOsMemoryWarning
Node.NotificationTranslationChanged
Node.NotificationWMAbout
Node.NotificationCrash
Node.NotificationOslmeUpdate
Node.NotificationApplicationResumed
Node.NotificationApplicationPaused
Node.NotificationApplicationFocusIn
Node.NotificationApplicationFocusOut
Node.NotificationTextServerChanged
Node.GetNode<T>(NodePath)
Node.GetNodeOrNull<T>(NodePath)
Node.GetChild<T>(int, bool)
Node.GetChildOrNull<T>(int, bool)
Node.GetOwner<T>()
Node.GetOwnerOrNull<T>()
Node.SetParent<T>()
Node.SetParentOrNull<T>()
Node._EnterTree()
Node._ExitTree()
Node._GetConfigurationWarnings()
Node._Input(InputEvent)
Node._PhysicsProcess(double)
Node._Process(double)
Node._Ready()
Node._ShortcutInput(InputEvent)
Node._UnhandledInput(InputEvent)
Node._UnhandledKeyInput(InputEvent)
Node.PrintOrphanNodes()
Node.AddSibling(Node, bool)
Node.AddChild(Node, bool, Node.InternalMode)
Node.RemoveChild(Node)
Node.Reparent(Node, bool)
Node.GetChildCount(bool)
Node.GetChildren(bool)
Node.GetChild(int, bool)
Node.HasNode(NodePath)
Node.GetNode(NodePath)
Node.GetNodeOrNull(NodePath)

Node.GetParent()
Node.FindChild(string, bool, bool)
Node.FindChildren(string, string, bool, bool)
Node.FindParent(string)
Node.HasNodeAndResource(NodePath)
Node.GetNodeAndResource(NodePath)
Node.IsInsideTree()
Node.IsPartOfEditedScene()
Node.IsAncestorOf(Node)
Node.IsGreater Than(Node)
Node.GetPath()
Node.GetPathTo(Node, bool)
Node.AddToGroup(StringName, bool)
Node.RemoveFromGroup(StringName)
Node.IsInGroup(StringName)
Node.MoveChild(Node, int)
Node.GetGroups()
Node.GetIndex(bool)
Node.PrintTree()
Node.PrintTreePretty()
Node.GetTreeString()
Node.GetTreeStringPretty()
Node.PropagateNotification(int)
Node.PropagateCall(StringName, Array, bool)
Node.SetPhysicsProcess(bool)
Node.GetPhysicsProcessDeltaTime()
Node.IsPhysicsProcessing()
Node.GetProcessDeltaTime()
Node.SetProcess(bool)
Node.IsProcessing()
Node.SetProcessInput(bool)
Node.IsProcessingInput()
Node.SetProcessShortcutInput(bool)
Node.IsProcessingShortcutInput()
Node.SetProcessUnhandledInput(bool)
Node.IsProcessingUnhandledInput()
Node.SetProcessUnhandledKeyInput(bool)
Node.IsProcessingUnhandledKeyInput()
Node.CanProcess()
Node.SetDisplayFolded(bool)
Node.IsDisplayedFolded()
Node.SetProcessInternal(bool)
Node.IsProcessingInternal()
Node.SetPhysicsProcessInternal(bool)
Node.IsPhysicsProcessingInternal()
Node.IsPhysicsInterpolated()
Node.IsPhysicsInterpolatedAndEnabled()
Node.ResetPhysicsInterpolation()
Node.SetTranslationDomainInherited()
Node.GetWindow()

Node.GetLastExclusiveWindow()
Node.GetTree()
Node.CreateTween()
Node.Duplicate(int)
Node.ReplaceBy(Node, bool)
Node.setSceneInstanceLoadPlaceholder(bool)
Node.setSceneInstanceLoadPlaceholder()
Node.setEditableInstance(Node, bool)
Node.setEditableInstance(Node)
Node.GetViewport()
Node.QueueFree()
Node.RequestReady()
Node.IsNodeReady()
Node.SetMultiplayerAuthority(int, bool)
Node.GetMultiplayerAuthority()
Node.IsMultiplayerAuthority()
Node.RpcConfig(StringName, Variant)
Node.GetRpcConfig()
Node.Atr(string, StringName)
Node.AtrN(string, StringName, int, StringName)
Node.Rpc(StringName, params Variant[])
Node.Rpc(string, ReadOnlySpan<Variant>)
Node.RpcId(long, StringName, params Variant[])
Node.RpcId(long, StringName, ReadOnlySpan<Variant>)
Node.UpdateConfigurationWarnings()
Node.CallDeferredThreadGroup(StringName, params Variant[])
Node.CallDeferredThreadGroup(StringName, ReadOnlySpan<Variant>)
Node.SetDeferredThreadGroup(StringName, Variant)
Node.NotifyDeferredThreadGroup(int)
Node.CallThreadSafe(StringName, params Variant[])
Node.CallThreadSafe(StringName, ReadOnlySpan<Variant>)
Node.SetThreadSafe(StringName, Variant)
Node.NotifyThreadSafe(int)
Node.EmitSignalReady()
Node.EmitSignalRenamed()
Node.EmitSignalTreeEntered()
Node.EmitSignalTreeExiting()
Node.EmitSignalTreeExited()
Node.EmitSignalChildEnteredTree(Node)
Node.EmitSignalChildExitingTree(Node)
Node.EmitSignalChildOrderChanged()
Node.EmitSignalReplacingBy(Node)
Node.EmitSignalEditorDescriptionChanged(Node)
Node.EmitSignalEditorStateChanged()
Node.Name
Node.UniqueNameInOwner
Node.SceneFilePath
Node.Owner
Node.Multiplayer
Node.ProcessMode

Node.ProcessPriority
Node.ProcessPhysicsPriority
Node.ProcessThreadGroup
Node.ProcessThreadGroupOrder
Node.ProcessThreadMessages
Node.PhysicsInterpolationMode
Node.AutoTranslateMode
Node.EditorDescription
Node.Ready
Node.Renamed
Node.TreeEntered
Node.TreeExiting
Node.TreeExited
Node.ChildEnteredTree
Node.ChildExitingTree
Node.ChildOrderChanged
Node.ReplacingBy
Node.EditorDescriptionChanged
Node.EditorStateChanged
GodotObject.NotificationPostinitialize
GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)

GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceID()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)
GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)
GodotObject.Callv(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObject.GetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()

object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Namespace: [cfGodotEngine.Controls](#)

Assembly: CatSweeper.dll

Syntax

```
[Tool]
[GlobalClass]
[ScriptPath("res://Modules/cfGodotEngine/atlaspack/controls/AtlasTextureRect.cs")]
public class AtlasTextureRect : TextureRect, IDisposable
```

Implements

[IDisposable](#)

Extension Methods

[NodeUtil.DontDestroyOnLoad\(Node\)](#)

© Bus Fighter. All rights reserved.

Class AtlasTextureRect.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject](#).[MethodName](#) → [Node](#).[MethodName](#) → [CanvasItem](#).[MethodName](#) → [Control](#).[MethodName](#) → [TextureRect](#).[MethodName](#) → [AtlasTextureRect](#).[MethodName](#)

Inherited Members

[TextureRect](#).[MethodName](#).[SetTexture](#)
[TextureRect](#).[MethodName](#).[GetTexture](#)
[TextureRect](#).[MethodName](#).[SetExpandMode](#)
[TextureRect](#).[MethodName](#).[GetExpandMode](#)
[TextureRect](#).[MethodName](#).[SetFlipH](#)
[TextureRect](#).[MethodName](#).[IsFlippedH](#)
[TextureRect](#).[MethodName](#).[SetFlipV](#)
[TextureRect](#).[MethodName](#).[IsFlippedV](#)
[TextureRect](#).[MethodName](#).[SetStretchMode](#)
[TextureRect](#).[MethodName](#).[GetStretchMode](#)
[Control](#).[MethodName](#).[_CanDropData](#)
[Control](#).[MethodName](#).[_DropData](#)
[Control](#).[MethodName](#).[_GetDragData](#)
[Control](#).[MethodName](#).[_GetMinimumSize](#)
[Control](#).[MethodName](#).[_GetTooltip](#)
[Control](#).[MethodName](#).[_Guilinput](#)
[Control](#).[MethodName](#).[_HasPoint](#)
[Control](#).[MethodName](#).[_MakeCustomTooltip](#)
[Control](#).[MethodName](#).[_StructuredTextParser](#)
[Control](#).[MethodName](#).[AcceptEvent](#)
[Control](#).[MethodName](#).[GetMinimumSize](#)
[Control](#).[MethodName](#).[GetCombinedMinimumSize](#)
[Control](#).[MethodName](#).[_SetLayoutMode](#)
[Control](#).[MethodName](#).[_GetLayoutMode](#)
[Control](#).[MethodName](#).[_SetAnchorsLayoutPreset](#)
[Control](#).[MethodName](#).[_GetAnchorsLayoutPreset](#)
[Control](#).[MethodName](#).[SetAnchorsPreset](#)
[Control](#).[MethodName](#).[SetOffsetsPreset](#)
[Control](#).[MethodName](#).[SetAnchorsAndOffsetsPreset](#)
[Control](#).[MethodName](#).[_SetAnchor](#)
[Control](#).[MethodName](#).[SetAnchor](#)
[Control](#).[MethodName](#).[GetAnchor](#)
[Control](#).[MethodName](#).[SetOffset](#)

Control.MethodName.GetOffset
Control.MethodName.SetAnchorAndOffset
Control.MethodName.SetBegin
Control.MethodName.SetEnd
Control.MethodName.SetPosition
Control.MethodName._SetPosition
Control.MethodName.SetSize
Control.MethodName.ResetSize
Control.MethodName._SetSize
Control.MethodName.SetCustomMinimumSize
Control.MethodName.SetGlobalPosition
Control.MethodName._SetGlobalPosition
Control.MethodName.SetRotation
Control.MethodName.SetRotationDegrees
Control.MethodName.setScale
Control.MethodName.SetPivotOffset
Control.MethodName.GetBegin
Control.MethodName.GetEnd
Control.MethodName.GetPosition
Control.MethodName.GetSize
Control.MethodName.GetRotation
Control.MethodName.GetRotationDegrees
Control.MethodName.GetScale
Control.MethodName.GetPivotOffset
Control.MethodName.GetCustomMinimumSize
Control.MethodName.GetParentAreaSize
Control.MethodName.GetGlobalPosition
Control.MethodName.GetScreenPosition
Control.MethodName.GetRect
Control.MethodName.GetGlobalRect
Control.MethodName.SetFocusMode
Control.MethodName.GetFocusMode
Control.MethodName.HasFocus
Control.MethodName.GrabFocus
Control.MethodName.ReleaseFocus
Control.MethodName.FindPrevValidFocus
Control.MethodName.FindNextValidFocus
Control.MethodName.FindValidFocusNeighbor
Control.MethodName.SetHSizeFlags
Control.MethodName.GetHSizeFlags
Control.MethodName.SetStretchRatio
Control.MethodName.GetStretchRatio
Control.MethodName.SetVSizeFlags
Control.MethodName.GetVSizeFlags
Control.MethodName.SetTheme
Control.MethodName.GetTheme
Control.MethodName.SetThemeTypeVariation
Control.MethodName.GetThemeTypeVariation
Control.MethodName.BeginBulkThemeOverride
Control.MethodName.EndBulkThemeOverride

Control.MethodName.AddThemelconOverride
Control.MethodName.AddThemeStyleboxOverride
Control.MethodName.AddThemeFontOverride
Control.MethodName.AddThemeFontSizeOverride
Control.MethodName.AddThemeColorOverride
Control.MethodName.AddThemeConstantOverride
Control.MethodName.RemoveThemelconOverride
Control.MethodName.RemoveThemeStyleboxOverride
Control.MethodName.RemoveThemeFontOverride
Control.MethodName.RemoveThemeFontSizeOverride
Control.MethodName.RemoveThemeColorOverride
Control.MethodName.RemoveThemeConstantOverride
Control.MethodName.GetThemelcon
Control.MethodName.GetThemeStylebox
Control.MethodName.GetThemeFont
Control.MethodName.GetThemeFontSize
Control.MethodName.GetThemeColor
Control.MethodName.GetThemeConstant
Control.MethodName.HasThemelconOverride
Control.MethodName.HasThemeStyleboxOverride
Control.MethodName.HasThemeFontOverride
Control.MethodName.HasThemeFontSizeOverride
Control.MethodName.HasThemeColorOverride
Control.MethodName.HasThemeConstantOverride
Control.MethodName.HasThemelcon
Control.MethodName.HasThemeStylebox
Control.MethodName.HasThemeFont
Control.MethodName.HasThemeFontSize
Control.MethodName.HasThemeColor
Control.MethodName.HasThemeConstant
Control.MethodName.GetThemeDefaultBaseScale
Control.MethodName.GetThemeDefaultFont
Control.MethodName.GetThemeDefaultFontSize
Control.MethodName.GetParentControl
Control.MethodName.SetHGrowDirection
Control.MethodName.GetHGrowDirection
Control.MethodName.SetVGrowDirection
Control.MethodName.GetVGrowDirection
Control.MethodName.SetToolTipAutoTranslateMode
Control.MethodName.GetTooltipAutoTranslateMode
Control.MethodName.SetToolTipText
Control.MethodName.GetTooltipText
Control.MethodName.GetTooltip
Control.MethodName.SetDefaultCursorShape
Control.MethodName.GetDefaultCursorShape
Control.MethodName.GetCursorShape
Control.MethodName.SetFocusNeighbor
Control.MethodName.GetFocusNeighbor
Control.MethodName.SetFocusNext
Control.MethodName.GetFocusNext

Control.MethodName.SetFocusPrevious
Control.MethodName.GetFocusPrevious
Control.MethodName.ForceDrag
Control.MethodName.SetMouseFilter
Control.MethodName.GetMouseFilter
Control.MethodName.SetForcePassScrollEvents
Control.MethodName.IsForcePassScrollEvents
Control.MethodName.SetClipContents
Control.MethodName.IsClippingContents
Control.MethodName.GrabClickFocus
Control.MethodName.SetDragForwarding
Control.MethodName.SetDragPreview
Control.MethodName.IsDragSuccessful
Control.MethodName.WarpMouse
Control.MethodName.SetShortcutContext
Control.MethodName.GetShortcutContext
Control.MethodName.UpdateMinimumSize
Control.MethodName.SetLayoutDirection
Control.MethodName.GetLayoutDirection
Control.MethodName.IsLayoutRtl
Control.MethodName.SetAutoTranslate
Control.MethodName.IsAutoTranslating
Control.MethodName.SetLocalizeNumeralSystem
Control.MethodName.IsLocalizingNumeralSystem
CanvasItem.MethodName._Draw
CanvasItem.MethodName.GetCanvasItem
CanvasItem.MethodName.SetVisible
CanvasItem.MethodName.IsVisible
CanvasItem.MethodName.IsVisibleInTree
CanvasItem.MethodName.Show
CanvasItem.MethodName.Hide
CanvasItem.MethodName.QueueRedraw
CanvasItem.MethodName.MoveToFront
CanvasItem.MethodName.SetAsTopLevel
CanvasItem.MethodName.IsSetAsTopLevel
CanvasItem.MethodName.SetLightMask
CanvasItem.MethodName.GetLightMask
CanvasItem.MethodName.SetModulate
CanvasItem.MethodName.GetModulate
CanvasItem.MethodName.SetSelfModulate
CanvasItem.MethodName.GetSelfModulate
CanvasItem.MethodName.SetZIndex
CanvasItem.MethodName.GetZIndex
CanvasItem.MethodName.SetZAsRelative
CanvasItem.MethodName.IsZRelative
CanvasItem.MethodName.SetYSortEnabled
CanvasItem.MethodName.IsYSortEnabled
CanvasItem.MethodName.SetDrawBehindParent
CanvasItem.MethodName.IsDrawBehindParentEnabled
CanvasItem.MethodName.DrawLine

CanvasItem.MethodName.DrawDashedLine
CanvasItem.MethodName.DrawPolyline
CanvasItem.MethodName.DrawPolylineColors
CanvasItem.MethodName.DrawArc
CanvasItem.MethodName.DrawMultiline
CanvasItem.MethodName.DrawMultilineColors
CanvasItem.MethodName.DrawRect
CanvasItem.MethodName.DrawCircle
CanvasItem.MethodName.DrawTexture
CanvasItem.MethodName.DrawTextureRect
CanvasItem.MethodName.DrawTextureRectRegion
CanvasItem.MethodName.DrawMsdfTextureRectRegion
CanvasItem.MethodName.DrawLcdTextureRectRegion
CanvasItem.MethodName.DrawStyleBox
CanvasItem.MethodName.DrawPrimitive
CanvasItem.MethodName.DrawPolygon
CanvasItem.MethodName.DrawColoredPolygon
CanvasItem.MethodName.DrawString
CanvasItem.MethodName.DrawMultilineString
CanvasItem.MethodName.DrawStringOutline
CanvasItem.MethodName.DrawMultilineStringOutline
CanvasItem.MethodName.DrawChar
CanvasItem.MethodName.DrawCharOutline
CanvasItem.MethodName.DrawMesh
CanvasItem.MethodName.DrawMultimesh
CanvasItem.MethodName.DrawSetTransform
CanvasItem.MethodName.DrawSetTransformMatrix
CanvasItem.MethodName.DrawAnimationSlice
CanvasItem.MethodName.DrawEndAnimation
CanvasItem.MethodName.GetTransform
CanvasItem.MethodName.GetGlobalTransform
CanvasItem.MethodName.GetGlobalTransformWithCanvas
CanvasItem.MethodName.GetViewportTransform
CanvasItem.MethodName.GetViewportRect
CanvasItem.MethodName.GetCanvasTransform
CanvasItem.MethodName.GetScreenTransform
CanvasItem.MethodName.GetLocalMousePosition
CanvasItem.MethodName.GetGlobalMousePosition
CanvasItem.MethodName.GetCanvas
CanvasItem.MethodName.GetCanvasLayerNode
CanvasItem.MethodName.GetWorld2D
CanvasItem.MethodName.SetMaterial
CanvasItem.MethodName.GetMaterial
CanvasItem.MethodName.SetInstanceShaderParameter
CanvasItem.MethodName.GetInstanceShaderParameter
CanvasItem.MethodName.SetUseParentMaterial
CanvasItem.MethodName.GetUseParentMaterial
CanvasItem.MethodName.SetNotifyLocalTransform
CanvasItem.MethodName.IsLocalTransformNotificationEnabled
CanvasItem.MethodName.SetNotifyTransform

CanvasItem.MethodName.IsTransformNotificationEnabled
CanvasItem.MethodName.ForceUpdateTransform
CanvasItem.MethodName.MakeCanvasPositionLocal
CanvasItem.MethodName.MakeInputLocal
CanvasItem.MethodName.SetVisibilityLayer
CanvasItem.MethodName.GetVisibilityLayer
CanvasItem.MethodName.SetVisibilityLayerBit
CanvasItem.MethodName.GetVisibilityLayerBit
CanvasItem.MethodName.SetTextureFilter
CanvasItem.MethodName.GetTextureFilter
CanvasItem.MethodName.SetTextureRepeat
CanvasItem.MethodName.GetTextureRepeat
CanvasItem.MethodName.SetClipChildrenMode
CanvasItem.MethodName.GetClipChildrenMode
Node.MethodName._EnterTree
Node.MethodName._ExitTree
Node.MethodName._GetConfigurationWarnings
Node.MethodName._Input
Node.MethodName._PhysicsProcess
Node.MethodName._Process
Node.MethodName._Ready
Node.MethodName._ShortcutInput
Node.MethodName._UnhandledInput
Node.MethodName._UnhandledKeyInput
Node.MethodName.PrintOrphanNodes
Node.MethodName.AddSibling
Node.MethodName.SetName
Node.MethodName.GetName
Node.MethodName.AddChild
Node.MethodName.RemoveChild
Node.MethodName.Reparent
Node.MethodName.GetChildCount
Node.MethodName.GetChildren
Node.MethodName.GetChild
Node.MethodName.HasNode
Node.MethodName.GetNode
Node.MethodName.GetNodeOrNull
Node.MethodName.GetParent
Node.MethodName.FindChild
Node.MethodName.FindChildren
Node.MethodName.FindParent
Node.MethodName.HasNodeAndResource
Node.MethodName.GetNodeAndResource
Node.MethodName.IsInsideTree
Node.MethodName.IsPartOfEditedScene
Node.MethodName.IsAncestorOf
Node.MethodName.IsGreaterThanOrEqual
Node.MethodName.GetPath
Node.MethodName.GetPathTo
Node.MethodName.AddToGroup

Node.MethodName.RemoveFromGroup
Node.MethodName.IsInGroup
Node.MethodName.MoveChild
Node.MethodName.GetGroups
Node.MethodName.SetOwner
Node.MethodName.GetOwner
Node.MethodName.GetIndex
Node.MethodName.PrintTree
Node.MethodName.PrintTreePretty
Node.MethodName.GetTreeString
Node.MethodName.GetTreeStringPretty
Node.MethodName.SetSceneFilePath
Node.MethodName.GetSceneFilePath
Node.MethodName.PropagateNotification
Node.MethodName.PropagateCall
Node.MethodName.SetPhysicsProcess
Node.MethodName.GetPhysicsProcessDeltaTime
Node.MethodName.IsPhysicsProcessing
Node.MethodName.GetProcessDeltaTime
Node.MethodName.SetProcess
Node.MethodName.SetProcessPriority
Node.MethodName.GetProcessPriority
Node.MethodName.SetPhysicsProcessPriority
Node.MethodName.GetPhysicsProcessPriority
Node.MethodName.IsProcessing
Node.MethodName.SetProcessInput
Node.MethodName.IsProcessingInput
Node.MethodName.SetProcessShortcutInput
Node.MethodName.IsProcessingShortcutInput
Node.MethodName.SetProcessUnhandledInput
Node.MethodName.IsProcessingUnhandledInput
Node.MethodName.SetProcessUnhandledKeyInput
Node.MethodName.IsProcessingUnhandledKeyInput
Node.MethodName.SetProcessMode
Node.MethodName.GetProcessMode
Node.MethodName.CanProcess
Node.MethodName.SetProcessThreadGroup
Node.MethodName.GetProcessThreadGroup
Node.MethodName.SetProcessThreadMessages
Node.MethodName.GetProcessThreadMessages
Node.MethodName.SetProcessThreadGroupOrder
Node.MethodName.GetProcessThreadGroupOrder
Node.MethodName.SetDisplayFolded
Node.MethodName.IsDisplayedFolded
Node.MethodName.SetProcessInternal
Node.MethodName.IsProcessingInternal
Node.MethodName.SetPhysicsProcessInternal
Node.MethodName.IsPhysicsProcessingInternal
Node.MethodName.SetPhysicsInterpolationMode
Node.MethodName.GetPhysicsInterpolationMode

Node.MethodName.IsPhysicsInterpolated
Node.MethodName.IsPhysicsInterpolatedAndEnabled
Node.MethodName.ResetPhysicsInterpolation
Node.MethodName.SetAutoTranslateMode
Node.MethodName.GetAutoTranslateMode
Node.MethodName.SetTranslationDomainInherited
Node.MethodName.GetWindow
Node.MethodName.GetLastExclusiveWindow
Node.MethodName.GetTree
Node.MethodName.CreateTween
Node.MethodName.Duplicate
Node.MethodName.ReplaceBy
Node.MethodName.SetSceneInstanceLoadPlaceholder
Node.MethodName.GetSceneInstanceLoadPlaceholder
Node.MethodName.SetEditableInstance
Node.MethodName.IsEditableInstance
Node.MethodName.GetViewport
Node.MethodName.QueueFree
Node.MethodName.RequestReady
Node.MethodName.IsNodeReady
Node.MethodName.SetMultiplayerAuthority
Node.MethodName.GetMultiplayerAuthority
Node.MethodName.IsMultiplayerAuthority
Node.MethodName.GetMultiplayer
Node.MethodName.RpcConfig
Node.MethodName.GetRpcConfig
Node.MethodName.SetEditorDescription
Node.MethodName.GetEditorDescription
Node.MethodName._SetImportPath
Node.MethodName._GetImportPath
Node.MethodName.SetUniqueNameInOwner
Node.MethodName.IsUniqueNameInOwner
Node.MethodName.Atr
Node.MethodName.AtrN
Node.MethodName.Rpc
Node.MethodName.Rpcld
Node.MethodName.UpdateConfigurationWarnings
Node.MethodName.CallDeferredThreadGroup
Node.MethodName.SetDeferredThreadGroup
Node.MethodName.NotifyDeferredThreadGroup
Node.MethodName.CallThreadSafe
Node.MethodName.SetThreadSafe
Node.MethodName.NotifyThreadSafe
GodotObject.MethodName._Get
GodotObject.MethodName._GetPropertyList
GodotObject.MethodName._IterGet
GodotObject.MethodName._IterInit
GodotObject.MethodName._IterNext
GodotObject.MethodName._Notification
GodotObject.MethodName._PropertyCanRevert

GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceID
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.CallV
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyListChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr
GodotObject.MethodName.TrN
GodotObject.MethodName.GetTranslationDomain
GodotObject.MethodName.SetTranslationDomain

GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfGodotEngine.Controls](#)

Assembly: CatSweeper.dll

Syntax

```
public class AtlasTextureRect.MethodName : TextureRect.MethodName
```

Fields

UpdateTexture

Cached name for the 'UpdateTexture' method.

Declaration

```
public static readonly StringName UpdateTexture
```

Field Value

TYPE

StringName

© Bus Fighter. All rights reserved.

Class AtlasTextureRect.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.PropertyName](#) → [Node.PropertyName](#) → [CanvasItem.PropertyName](#) → [Control.PropertyName](#) → [TextureRect.PropertyName](#) → [AtlasTextureRect.PropertyName](#)

Inherited Members

TextureRect.PropertyName.Texture
TextureRect.PropertyName.ExpandMode
TextureRect.PropertyName.StretchMode
TextureRect.PropertyName.FlipH
TextureRect.PropertyName.FlipV
Control.PropertyName.ClipContents
Control.PropertyName.CustomMinimumSize
Control.PropertyName.LayoutDirection
Control.PropertyName.LayoutMode
Control.PropertyName.AnchorsPreset
Control.PropertyName.AnchorLeft
Control.PropertyName.AnchorTop
Control.PropertyName.AnchorRight
Control.PropertyName.AnchorBottom
Control.PropertyName.OffsetLeft
Control.PropertyName.OffsetTop
Control.PropertyName.OffsetRight
Control.PropertyName.OffsetBottom
Control.PropertyName.GrowHorizontal
Control.PropertyName.GrowVertical
Control.PropertyName.Size
Control.PropertyName.Position
Control.PropertyName.GlobalPosition
Control.PropertyName.Rotation
Control.PropertyName.RotationDegrees
Control.PropertyName.Scale
Control.PropertyName.PivotOffset
Control.PropertyName.SizeTypeHorizontal
Control.PropertyName.SizeTypeVertical
Control.PropertyName.SizeTypeStretchRatio
Control.PropertyName.LocalizeNumeralSystem
Control.PropertyName.AutoTranslate
Control.PropertyName.TooltipText

Control.PropertyName.TooltipAutoTranslateMode
Control.PropertyName.FocusNeighborLeft
Control.PropertyName.FocusNeighborTop
Control.PropertyName.FocusNeighborRight
Control.PropertyName.FocusNeighborBottom
Control.PropertyName.FocusNext
Control.PropertyName.FocusPrevious
Control.PropertyName.FocusMode
Control.PropertyName.MouseFilter
Control.PropertyName.MouseForcePassScrollEvents
Control.PropertyName.MouseDefaultCursorShape
Control.PropertyName.ShortcutContext
Control.PropertyName.Theme
Control.PropertyName.ThemeTypeVariation
CanvasItem.PropertyName.Visible
CanvasItem.PropertyName.Modulate
CanvasItem.PropertyName.SelfModulate
CanvasItem.PropertyName.ShowBehindParent
CanvasItem.PropertyName.TopLevel
CanvasItem.PropertyName.ClipChildren
CanvasItem.PropertyName.LightMask
CanvasItem.PropertyName.VisibilityLayer
CanvasItem.PropertyName.ZIndex
CanvasItem.PropertyName.ZAsRelative
CanvasItem.PropertyName.YSortEnabled
CanvasItem.PropertyName.TextureFilter
CanvasItem.PropertyName.TextureRepeat
CanvasItem.PropertyName.Material
CanvasItem.PropertyName.UseParentMaterial
Node.PropertyName._ImportPath
Node.PropertyName.Name
Node.PropertyName.UniqueNameInOwner
Node.PropertyName.SceneFilePath
Node.PropertyName.Owner
Node.PropertyName.Multiplayer
Node.PropertyName.ProcessMode
Node.PropertyName.ProcessPriority
Node.PropertyName.ProcessPhysicsPriority
Node.PropertyName.ProcessThreadGroup
Node.PropertyName.ProcessThreadGroupOrder
Node.PropertyName.ProcessThreadMessages
Node.PropertyName.PhysicsInterpolationMode
Node.PropertyName.AutoTranslateMode
Node.PropertyName.EditorDescription
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Controls](#)

Assembly: CatSweeper.dll

Syntax

```
public class AtlasTextureRect.PropertyName : TextureRect.PropertyName
```

Fields

_textureRef

Cached name for the '_textureRef' field.

Declaration

```
public static readonly StringName _textureRef
```

Field Value

TYPE

StringName

textureRef

Cached name for the 'textureRef' property.

Declaration

```
public static readonly StringName textureRef
```

Field Value

TYPE

StringName

Class AtlasTextureRect.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.SignalName](#) → [Node.SignalName](#) → [CanvasItem.SignalName](#) → [Control.SignalName](#) → [TextureRect.SignalName](#) → [AtlasTextureRect.SignalName](#)

Inherited Members

[Control.SignalName.Resized](#)
[Control.SignalName.GuiInput](#)
[Control.SignalName.MouseEntered](#)
[Control.SignalName.MouseExited](#)
[Control.SignalName.FocusEntered](#)
[Control.SignalName.FocusExited](#)
[Control.SignalName.SizeFlagsChanged](#)
[Control.SignalName.MinimumSizeChanged](#)
[Control.SignalName.ThemeChanged](#)
[CanvasItem.SignalName.Draw](#)
[CanvasItem.SignalName.VisibilityChanged](#)
[CanvasItem.SignalName.Hidden](#)
[CanvasItem.SignalName.ItemRectChanged](#)
[Node.SignalName.Ready](#)
[Node.SignalName.Renamed](#)
[Node.SignalName.TreeEntered](#)
[Node.SignalName.TreeExiting](#)
[Node.SignalName.TreeExited](#)
[Node.SignalName.ChildEnteredTree](#)
[Node.SignalName.ChildExitingTree](#)
[Node.SignalName.ChildOrderChanged](#)
[Node.SignalName.ReplacingBy](#)
[Node.SignalName.EditorDescriptionChanged](#)
[Node.SignalName.EditorStateChanged](#)
[GodotObject.SignalName.ScriptChanged](#)
[GodotObject.SignalName.PropertyListChanged](#)
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Syntax

```
public class AtlasTextureRect.SignalName : TextureRect.SignalName
```

© Bus Fighter. All rights reserved.

Namespace cfGodotEngine.Core

Classes

[GodotLogger](#)

© Bus Fighter. All rights reserved.

Class GodotLogger

Inheritance

[object](#) → GodotLogger

Implements

[ILogger](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Core](#)

Assembly: CatSweeper.dll

Syntax

```
public class GodotLogger : ILogger
```

Methods

Asset(bool, object)

Declaration

```
public void Asset(bool condition, object context = null)
```

Parameters

TYPE	NAME
------	------

bool	condition
------	-----------

TYPE NAME

object context

LogDebug(string, object)

Declaration

```
public void LogDebug(string message, object context = null)
```

Parameters

TYPE NAME

string message

object context

.LogError(string, object)

Declaration

```
public void LogError(string message, object context = null)
```

Parameters

TYPE NAME

string message

object context

LogException(Exception, object)

Declaration

```
public void LogException(Exception ex, object message = null)
```

Parameters

TYPE	NAME
------	------

Exception	ex
-----------	----

object	message
--------	---------

LogInfo(string, object)

Declaration

```
public void LogInfo(string message, object context = null)
```

Parameters

TYPE	NAME
------	------

string	message
--------	---------

object	context
--------	---------

LogWarning(string, object)

Declaration

```
public void LogWarning(string message, object context = null)
```

Parameters

TYPE	NAME
------	------

string	message
--------	---------

object	context
--------	---------

Implements

[ILogger](#)

Namespace cfGodotEngine.GoogleDrive

Classes

[AssetDirectFileMirror](#)

[ChangeHandler](#)

[DriveMirror](#)

[DriveMirrorSetting](#)

[DriveMirrorSetting.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[DriveMirrorSetting.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

[DriveMirrorSetting.SignalName](#)

Cached StringNames for the signals contained in this class, for fast lookup.

[DriveUtil](#)

[SettingItem](#)

[SettingItem.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[SettingItem.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

[SettingItem.SignalName](#)

Cached StringNames for the signals contained in this class, for fast lookup.

Structs

[ChangelInfo](#)

[FileHandler.DownloadRequest](#)

[FileHandler.FileItem](#)

[FolderMimeHandler](#)

[RefreshRequest](#)

[RefreshResult](#)

[RefreshStatus](#)

[SheetFileHandler](#)

[UrlInfo](#)

Interfaces

[FileHandler](#)

[IChangeHandler](#)

[IFileMirrorHandler](#)

Enums

[ChangeType](#)

[FileType](#)

© Bus Fighter. All rights reserved.

Class AssetDirectFileMirror

Inheritance

[object](#) → AssetDirectFileMirror

Implements

[IFileMirrorHandler](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public class AssetDirectFileMirror : IFileMirrorHandler
```

Constructors

AssetDirectFileMirror(ILOGGER, string)

Declaration

```
public AssetDirectFileMirror(ILOGGER logger, string assetDirectoryPath)
```

Parameters

TYPE	NAME
ILogger	logger

TYPE	NAME
------	------

string	assetDirectoryPath
--------	--------------------

Methods

RefreshFiles(DriveService, in RefreshRequest)

Declaration

```
public void RefreshFiles(DriveService driveService, in RefreshRequest request)
```

Parameters

TYPE	NAME
------	------

DriveService	driveService
RefreshRequest	request

RefreshFilesAsync(DriveService, RefreshRequest)

Declaration

```
public IEnumerable<Task<RefreshStatus>> RefreshFilesAsync(DriveService driveService, RefreshRe
```

Parameters

TYPE	NAME
------	------

DriveService	driveService
RefreshRequest	request

Returns

TYPE

IEnumerable<Task<RefreshStatus>>

Implements

© Bus Fighter. All rights reserved.

Class ChangeHandler

Inheritance

[object](#) → ChangeHandler

Implements

[IChangeHandler](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public class ChangeHandler : IChangeHandler
```

Constructors

ChangeHandler(ILogger)

Declaration

```
public ChangeHandler(ILogger logger)
```

Parameters

TYPE	NAME
ILogger	logger

Properties

ChangedFiles

Declaration

```
public IReadOnlyList<Change> ChangedFiles { get; }
```

Property Value

TYPE

IReadOnlyList<Change>

Methods

IsFileChanged(File)

Declaration

```
public bool IsFileChanged(File googleFile)
```

Parameters

TYPE NAME

File googleFile

Returns

TYPE

bool

LoadChanges(DriveService, string)

Declaration

```
public string LoadChanges(DriveService driveService, string startPageToken)
```

Parameters

TYPE	NAME
DriveService	driveService
string	startPageToken

Returns

TYPE
string

LoadChangesAsync(DriveService, string)

Declaration

```
public Task<string> LoadChangesAsync(DriveService driveService, string startPageToken)
```

Parameters

TYPE	NAME
DriveService	driveService
string	startPageToken

Returns

TYPE
Task<string>

TryGetFileChange(File, out ChangefInfo?)

Declaration

```
public bool TryGetFileChange(File googleFile, out ChangeInfo? changeInfo)
```

Parameters

TYPE	NAME
File	googleFile
ChangefInfo?	changeInfo

Returns

TYPE

bool

Implements

IChangeHandler

© Bus Fighter. All rights reserved.

Struct ChangeInfo

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public struct ChangeInfo
```

Fields

File

Declaration

```
public Optional<File> File
```

Field Value

TYPE

[Optional<File>](#)

type

Declaration

```
public ChangeType type
```

Field Value

TYPE

ChangeType

© Bus Fighter. All rights reserved.

Enum ChangeType

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public enum ChangeType
```

Fields

NAME
Modified
None
Removed

Extension Methods

[EnumExtension.hasFlag\(Enum, Enum\)](#)

© Bus Fighter. All rights reserved.

Class DriveMirror

Inheritance

[object](#) → DriveMirror

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public class DriveMirror
```

Constructors

DriveMirror(IFileMirrorHandler, ILogger)

Declaration

```
public DriveMirror(IFileMirrorHandler mirrorHandler, ILogger logger)
```

Parameters

TYPE	NAME
IFileMirrorHandler	mirrorHandler
ILogger	logger

Properties

instance

Declaration

```
public static DriveMirror instance { get; }
```

Property Value

TYPE

DriveMirror

Methods

ClearAllAndRefreshAsync()

Declaration

```
public Task<IEnumerable<RefreshStatus>> ClearAllAndRefreshAsync()
```

Returns

TYPE

Task<IEnumerable<RefreshStatus>>

ClearAllAndRefreshWithProgressBar()

Declaration

```
public Task ClearAllAndRefreshWithProgressBar()
```

Returns

TYPE

Task

Refresh()

Declaration

```
public void Refresh()
```

RefreshAsync()

Declaration

```
public Task<IEnumerable<RefreshStatus>> RefreshAsync()
```

Returns

TYPE

Task<IEnumerable<RefreshStatus>>

RefreshWithProgressBar()

Declaration

```
public Task RefreshWithProgressBar()
```

Returns

TYPE

Task

© Bus Fighter. All rights reserved.

Class DriveMirrorSetting

Inheritance

object → GodotObject → RefCounted → Resource → [Setting<DriveMirrorSetting>](#) → DriveMirrorSetting

Implements

[IDisposable](#)

Inherited Members

[Setting<DriveMirrorSetting>.GetSetting\(\)](#)

Resource._GetRid()

Resource._ResetState()

[Resource._SetPathCache\(string\)](#)

Resource._SetupLocalToScene()

[Resource.TakeOverPath\(string\)](#)

[Resource.SetPathCache\(string\)](#)

Resource.GetRid()

Resource.GetLocalScene()

Resource.SetupLocalToScene()

Resource.ResetState()

[Resource.SetIdForPath\(string, string\)](#)

[Resource.GetIdForPath\(string\)](#)

Resource.IsBuiltIn()

Resource.GenerateSceneUniqueId()

Resource.EmitChanged()

[Resource.Duplicate\(bool\)](#)

Resource.EmitSignalChanged()

Resource.EmitSignalSetupLocalToSceneRequested()

Resource.InvokeGodotClassMethod(in godot_string_name, NativeVariantPtrArgs, out godot_variant)

Resource.HasGodotClassMethod(in godot_string_name)

Resource.HasGodotClassSignal(in godot_string_name)

Resource.ResourceLocalToScene

Resource.ResourcePath

Resource.ResourceName

Resource.ResourceSceneUniqueId

Resource.Changed

Resource.SetupLocalToSceneRequested

RefCounted.InitRef()

RefCounted.Reference()

RefCounted.Unreference()

RefCounted.GetReferenceCount()

GodotObject.NotificationPostinitialize
GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsInstanceValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)
GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)

GodotObject.SetDeferred(StringName, Variant)
GodotObject.Callv(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObject.GetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
[SettingPath("res://Setting/GoogleDrive/DriveMirrorSetting.tres")]
[Tool]
[GlobalClass]
[ScriptPath("res://Modules/cfGodotEngine/google/drive/DriveMirrorSetting.cs")]
public class DriveMirrorSetting : Setting<DriveMirrorSetting>, IDisposable
```

Fields

changeChecksumToken

Declaration

```
[Export(PropertyHint.None, "")]  
public string changeChecksumToken
```

Field Value

TYPE

string

items

Declaration

```
[Export(PropertyHint.None, "")]  
public Array<SettingItem> items
```

Field Value

TYPE

Array<[SettingItem](#)>

refreshOnEnterPlayMode

Declaration

```
[Export(PropertyHint.None, "")]  
public bool refreshOnEnterPlayMode
```

Field Value

TYPE

bool

Properties

ForceRefreshAllButton

Declaration

```
[ExportToolButton("Force Refresh All")]
public Callable ForceRefreshAllButton { get; }
```

Property Value

TYPE

Callable

RefreshButton

Declaration

```
[ExportToolButton("Refresh")]
public Callable RefreshButton { get; }
```

Property Value

TYPE

Callable

serviceAccountCredentialJson

Declaration

```
public string serviceAccountCredentialJson { get; }
```

Property Value

TYPE

string

settingMap

Declaration

```
public Dictionary<string, SettingItem> settingMap { get; }
```

Property Value

TYPE

Dictionary<string, SettingItem>

Methods

ForceRefreshAll()

Declaration

```
public void ForceRefreshAll()
```

Refresh()

Declaration

```
public void Refresh()
```

Implements

IDisposable

© Bus Fighter. All rights reserved.

Class DriveMirrorSetting.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject](#).[MethodName](#) → [RefCounted](#).[MethodName](#) → [Resource](#).[MethodName](#) → [Setting<DriveMirrorSetting>.MethodName](#) → [DriveMirrorSetting](#).[MethodName](#)

Inherited Members

[Resource](#).[MethodName](#).[_GetRid](#)
[Resource](#).[MethodName](#).[_ResetState](#)
[Resource](#).[MethodName](#).[_SetPathCache](#)
[Resource](#).[MethodName](#).[_SetupLocalToScene](#)
[Resource](#).[MethodName](#).[SetPath](#)
[Resource](#).[MethodName](#).[TakeOverPath](#)
[Resource](#).[MethodName](#).[GetPath](#)
[Resource](#).[MethodName](#).[SetPathCache](#)
[Resource](#).[MethodName](#).[SetName](#)
[Resource](#).[MethodName](#).[GetName](#)
[Resource](#).[MethodName](#).[GetRid](#)
[Resource](#).[MethodName](#).[SetLocalToScene](#)
[Resource](#).[MethodName](#).[IsLocalToScene](#)
[Resource](#).[MethodName](#).[GetLocalScene](#)
[Resource](#).[MethodName](#).[SetupLocalToScene](#)
[Resource](#).[MethodName](#).[ResetState](#)
[Resource](#).[MethodName](#).[SetIdForPath](#)
[Resource](#).[MethodName](#).[GetIdForPath](#)
[Resource](#).[MethodName](#).[IsBuiltIn](#)
[Resource](#).[MethodName](#).[GenerateSceneUniqueId](#)
[Resource](#).[MethodName](#).[SetSceneUniqueId](#)
[Resource](#).[MethodName](#).[GetSceneUniqueId](#)
[Resource](#).[MethodName](#).[EmitChanged](#)
[Resource](#).[MethodName](#).[Duplicate](#)
[RefCounted](#).[MethodName](#).[InitRef](#)
[RefCounted](#).[MethodName](#).[Reference](#)
[RefCounted](#).[MethodName](#).[Unreference](#)
[RefCounted](#).[MethodName](#).[GetReferenceCount](#)
[GodotObject](#).[MethodName](#).[_Get](#)
[GodotObject](#).[MethodName](#).[_GetPropertyList](#)
[GodotObject](#).[MethodName](#).[_IterGet](#)
[GodotObject](#).[MethodName](#).[_IterInit](#)
[GodotObject](#).[MethodName](#).[_IterNext](#)

GodotObject.MethodName._Notification
GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceId
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyListChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr
GodotObject.MethodName.TrN

GodotObject.MethodName.GetTranslationDomain
GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public class DriveMirrorSetting.MethodName : Setting<DriveMirrorSetting>.MethodName
```

Fields

ForceRefreshAll

Cached name for the 'ForceRefreshAll' method.

Declaration

```
public static readonly StringName ForceRefreshAll
```

Field Value

TYPE

StringName

Refresh

Cached name for the 'Refresh' method.

Declaration

```
public static readonly StringName Refresh
```

Field Value

TYPE

StringName

UpdateSettingMap

Cached name for the 'UpdateSettingMap' method.

Declaration

```
public static readonly StringName UpdateSettingMap
```

Field Value

TYPE

StringName

© Bus Fighter. All rights reserved.

Class DriveMirrorSetting.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.PropertyName](#) → [RefCounted.PropertyName](#) → [Resource.PropertyName](#) → [Setting<DriveMirrorSetting>.PropertyName](#) → [DriveMirrorSetting.PropertyName](#)

Inherited Members

[Resource.PropertyName.ResourceLocalToScene](#)

[Resource.PropertyName.ResourcePath](#)

[Resource.PropertyName.ResourceName](#)

[Resource.PropertyName.ResourceSceneUniqueld](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public class DriveMirrorSetting.PropertyName : Setting<DriveMirrorSetting>.PropertyName
```

Fields

ForceRefreshAllButton

Cached name for the 'ForceRefreshAllButton' property.

Declaration

```
public static readonly StringName ForceRefreshAllButton
```

Field Value

TYPE

StringName

RefreshButton

Cached name for the 'RefreshButton' property.

Declaration

```
public static readonly StringName RefreshButton
```

Field Value

TYPE

StringName

_serviceAccountCredentialJsonPath

Cached name for the '_serviceAccountCredentialJsonPath' field.

Declaration

```
public static readonly StringName _serviceAccountCredentialJsonPath
```

Field Value

TYPE

StringName

changeChecksumToken

Cached name for the 'changeChecksumToken' field.

Declaration

```
public static readonly StringName changeChecksumToken
```

Field Value

TYPE

StringName

items

Cached name for the 'items' field.

Declaration

```
public static readonly StringName items
```

Field Value

TYPE

StringName

refreshOnEnterPlayMode

Cached name for the 'refreshOnEnterPlayMode' field.

Declaration

```
public static readonly StringName refreshOnEnterPlayMode
```

Field Value

TYPE

StringName

serviceAccountCredentialJson

Cached name for the 'serviceAccountCredentialJson' property.

Declaration

```
public static readonly StringName serviceAccountCredentialJson
```

Field Value

TYPE

StringName

serviceAccountCredentialJsonPath

Cached name for the 'serviceAccountCredentialJsonPath' property.

Declaration

```
public static readonly StringName serviceAccountCredentialJsonPath
```

Field Value

TYPE

StringName

© Bus Fighter. All rights reserved.

Class DriveMirrorSetting.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

[object](#) → GodotObject.SignalName → RefCounted.SignalName → Resource.SignalName → [Setting<DriveMirrorSetting>.SignalName](#) → DriveMirrorSetting.SignalName

Inherited Members

Resource.SignalName.Changed

Resource.SignalName.SetupLocalToSceneRequested

GodotObject.SignalName.ScriptChanged

GodotObject.SignalName.PropertyListChanged

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public class DriveMirrorSetting.SignalName : Setting<DriveMirrorSetting>.SignalName
```

© Bus Fighter. All rights reserved.

Class DriveUtil

Inheritance

[object](#) → DriveUtil

Inherited Members

[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public static class DriveUtil
```

Fields

MimeFileHandlers

Declaration

```
public static IReadOnlyDictionary<string, FileHandler> MimeFileHandlers
```

Field Value

TYPE

[IReadOnlyDictionary<string, FileHandler>](#)

godotLogger

Declaration

```
public static readonly ILogger godotLogger
```

Field Value

TYPE

ILogger

Methods

FormLink(string)

Declaration

```
public static string FormLink(string driveFileDialog)
```

Parameters

TYPE NAME

string driveFileDialog

Returns

TYPE

string

ParseSegments(ReadOnlyMemory<string>)

Declaration

```
public static Res<UrlInfo, Exception> ParseSegments(ReadOnlyMemory<string> segments)
```

Parameters

TYPE

NAME

ReadOnlyMemory<string>	segments
------------------------	----------

Returns

TYPE

Res<UrlInfo, Exception>

ParseUrl(string)

Declaration

```
public static Res<UrlInfo, Exception> ParseUrl(string driveLink)
```

Parameters

TYPE NAME

string	driveLink
--------	-----------

Returns

TYPE

Res<UrlInfo, Exception>

© Bus Fighter. All rights reserved.

Interface FileHandler

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public interface FileHandler
```

Methods

DownloadAsync(FilesResource, DownloadRequest)

Declaration

```
Task<IDownloadProgress> DownloadAsync(FilesResource filesResource, FileHandler.DownloadRequest
```

Parameters

TYPE	NAME
FilesResource	filesResource
FileHandler.DownloadRequest	downloadRequest

Returns

TYPE
Task<IDownloadProgress>

DownloadWithStatus(FilesResource, in DownloadRequest)

Declaration

`IDownloadProgress DownloadWithStatus(FilesResource filesResource, in FileHandler.DownloadReque`

Parameters

TYPE	NAME
FilesResource	filesResource
FileHandler.DownloadRequest	downloadRequest

Returns

TYPE
<code>IDownloadProgress</code>

© Bus Fighter. All rights reserved.

Struct FileHandler.DownloadRequest

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public struct FileHandler.DownloadRequest
```

Fields

changeHandler

Declaration

```
public IChangeHandler changeHandler
```

Field Value

TYPE

[IChangeHandler](#)

googleFileId

Declaration

```
public string googleFileDialog
```

Field Value

TYPE

string

localName

Declaration

```
public string localName
```

Field Value

TYPE

string

rootDirectoryInfo

Declaration

```
public DirectoryInfo rootDirectoryInfo
```

Field Value

TYPE

DirectoryInfo

© Bus Fighter. All rights reserved.

Struct FileHandler.FileItem

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public struct FileHandler.FileItem
```

Fields

RelativePathSegment

Declaration

```
public PathSegment RelativePathSegment
```

Field Value

TYPE

[PathSegment](#)

googleFile

Declaration

```
public File googleFile
```

Field Value

TYPE

File

© Bus Fighter. All rights reserved.

Enum FileType

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public enum FileType
```

Fields

NAME
Folder
None
Sheet

Extension Methods

[EnumExtension.hasFlag\(Enum, Enum\)](#)

© Bus Fighter. All rights reserved.

Struct FolderMimeHandler

Implements

[FileHandler](#)

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public struct FolderMimeHandler : FileHandler
```

Constructors

FolderMimeHandler(ILocator, string)

Declaration

```
public FolderMimeHandler(ILocator logger, string assetDirectoryPath)
```

Parameters

TYPE	NAME
ILocator	logger
string	assetDirectoryPath

Methods

DownloadAsync(FilesResource, DownloadRequest)

Declaration

```
public Task<IDownloadProgress> DownloadAsync(FilesResource filesResource, FileHandler.Download
```

Parameters

TYPE	NAME
FilesResource	filesResource
FileHandler.DownloadRequest	downloadRequest

Returns

TYPE
Task<IDownloadProgress>

DownloadWithStatus(FilesResource, in DownloadRequest)

Declaration

```
public IDownloadProgress DownloadWithStatus(FilesResource filesResource, in FileHandler.Downlo
```

Parameters

TYPE	NAME
FilesResource	filesResource
FileHandler.DownloadRequest	downloadRequest

Returns

TYPE
IDownloadProgress

GetFolderContent(FilesResource, string)

Declaration

```
public IEnumerable<FileHandler.FileItem> GetFolderContent(FilesResource filesResource, string
```

Parameters

TYPE	NAME
FilesResource	filesResource
string	googleFileDialog

Returns

TYPE
IEnumerable<FileHandler.FileItem>

GetFolderContentAsync(FilesResource, string)

Declaration

```
public Task<IEnumerable<FileHandler.FileItem>> GetFolderContentAsync(FilesResource filesResour
```

Parameters

TYPE	NAME
FilesResource	filesResource
string	googleFileDialog

Returns

TYPE
Task<IEnumerable<FileHandler.FileItem>>

Implements

FileHandler

Interface IChangeHandler

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IChangeHandler
```

Methods

IsFileChanged(File)

Declaration

```
bool IsFileChanged(File googleFile)
```

Parameters

TYPE	NAME
------	------

File	googleFile
------	------------

Returns

TYPE

bool

LoadChanges(DriveService, string)

Declaration

```
string LoadChanges(DriveService driveService, string startPageToken)
```

Parameters

TYPE	NAME
DriveService	driveService
string	startPageToken

Returns

TYPE
string

LoadChangesAsync(DriveService, string)

Declaration

```
Task<string> LoadChangesAsync(DriveService driveService, string startPageToken)
```

Parameters

TYPE	NAME
DriveService	driveService
string	startPageToken

Returns

TYPE
Task<string>

TryGetFileChange(File, out ChangefInfo?)

Declaration

```
bool TryGetFileChange(File googleFile, out ChangeInfo? changeInfo)
```

Parameters

TYPE	NAME
File	googleFile
ChangefInfo?	changeInfo

Returns

TYPE

bool

© Bus Fighter. All rights reserved.

Interface IFileMirrorHandler

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public interface IFileMirrorHandler
```

Methods

RefreshFiles(DriveService, in RefreshRequest)

Declaration

```
void RefreshFiles(DriveService driveService, in RefreshRequest request)
```

Parameters

TYPE	NAME
DriveService	driveService
RefreshRequest	request

RefreshFilesAsync(DriveService, RefreshRequest)

Declaration

```
IEnumerable<Task<RefreshStatus>> RefreshFilesAsync(DriveService driveService, RefreshRequest r
```

Parameters

TYPE**NAME**

DriveService driveService

RefreshRequest request

Returns**TYPE**

IEnumerable<Task<RefreshStatus>>

© Bus Fighter. All rights reserved.

Struct RefreshRequest

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public struct RefreshRequest
```

Fields

changeHandler

Declaration

```
public IChangeHandler changeHandler
```

Field Value

TYPE

[IChangeHandler](#)

getSetting

Declaration

```
public Func<File, Res<Optional<SettingItem>, Exception>> getSetting
```

Field Value

TYPE

Func<File, Res<Optional<SettingItem>, Exception>>

googleFiles

Declaration

```
public IList<File> googleFiles
```

Field Value

TYPE

IList<File>

© Bus Fighter. All rights reserved.

Struct RefreshResult

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public struct RefreshResult
```

Fields

newChangeChecksumToken

Declaration

```
public string newChangeChecksumToken
```

Field Value

TYPE

[string](#)

Struct RefreshStatus

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public struct RefreshStatus
```

Constructors

RefreshStatus(File, IDownloadProgress, float)

Declaration

```
public RefreshStatus(File file, IDownloadProgress status, float progress)
```

Parameters

TYPE	NAME
File	file
IDownloadProgress	status
float	progress

Fields

file

Declaration

```
public readonly File file
```

Field Value

TYPE

File

progress

Declaration

```
public readonly float progress
```

Field Value

TYPE

float

status

Declaration

```
public readonly IDownloadProgress status
```

Field Value

TYPE

IDownloadProgress

Class SettingItem

Inheritance

[object](#) → GodotObject → RefCounted → Resource → SettingItem

Implements

[IDisposable](#)

Inherited Members

Resource._GetRid()

Resource._ResetState()

Resource._SetPathCache(string)

Resource._SetupLocalToScene()

Resource.TakeOverPath(string)

Resource.SetPathCache(string)

Resource.GetRid()

Resource.GetLocalScene()

Resource.SetupLocalToScene()

Resource.ResetState()

Resource.SetIdForPath(string, string)

Resource.GetIdForPath(string)

Resource.IsBuiltIn()

Resource.GenerateSceneUniqueId()

Resource.EmitChanged()

Resource.Duplicate(bool)

Resource.EmitSignalChanged()

Resource.EmitSignalSetupLocalToSceneRequested()

Resource.InvokeGodotClassMethod(in godot_string_name, NativeVariantPtrArgs, out godot_variant)

Resource.HasGodotClassMethod(in godot_string_name)

Resource.HasGodotClassSignal(in godot_string_name)

Resource.ResourceLocalToScene

Resource.ResourcePath

Resource.ResourceName

Resource.ResourceSceneUniqueId

Resource.Changed

Resource.SetupLocalToSceneRequested

RefCounted.InitRef()

RefCounted.Reference()

RefCounted.Unreference()

RefCounted.GetReferenceCount()

GodotObject.NotificationPostInitialize

GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)
GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)

GodotObject.Callv(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObject.GetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Namespace: **cfGodotEngine.GoogleDrive**

Assembly: CatSweeper.dll

Syntax

```
[Tool]
[GlobalClass]
[ScriptPath("res://Modules/cfGodotEngine/google/drive/SettingItem.cs")]
public class SettingItem : Resource, IDisposable
```

Fields

assetPath

Declaration

```
[Export(PropertyHint.None, "")]  
public string assetPath
```

Field Value

TYPE

string

driveLink

Declaration

```
[Export(PropertyHint.None, "")]  
public string driveLink
```

Field Value

TYPE

string

Properties

fileName

Declaration

```
[Export(PropertyHint.None, "")]  
public string fileName { get; set; }
```

Property Value

TYPE

string

Implements

[IDisposable](#)

© Bus Fighter. All rights reserved.

Class SettingItem.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.MethodName](#) → [RefCounted.MethodName](#) → [Resource.MethodName](#) → [SettingItem.MethodName](#)

Inherited Members

[Resource.MethodName._GetRid](#)
[Resource.MethodName._ResetState](#)
[Resource.MethodName._SetPathCache](#)
[Resource.MethodName._SetupLocalToScene](#)
[Resource.MethodName.SetPath](#)
[Resource.MethodName.TakeOverPath](#)
[Resource.MethodName.GetPath](#)
[Resource.MethodName.SetPathCache](#)
[Resource.MethodNameSetName](#)
[Resource.MethodName.GetName](#)
[Resource.MethodName.GetRid](#)
[Resource.MethodName.SetLocalToScene](#)
[Resource.MethodName.IsLocalToScene](#)
[Resource.MethodName.GetLocalScene](#)
[Resource.MethodName.SetupLocalToScene](#)
[Resource.MethodName.ResetState](#)
[Resource.MethodName.SetIdForPath](#)
[Resource.MethodName.GetIdForPath](#)
[Resource.MethodName.IsBuiltIn](#)
[Resource.MethodName.GenerateSceneUniqueId](#)
[Resource.MethodName.SetSceneUniqueId](#)
[Resource.MethodName.GetSceneUniqueId](#)
[Resource.MethodName.EmitChanged](#)
[Resource.MethodName.Duplicate](#)
[RefCounted.MethodName.InitRef](#)
[RefCounted.MethodName.Reference](#)
[RefCounted.MethodName.Unreference](#)
[RefCounted.MethodName.GetReferenceCount](#)
[GodotObject.MethodName._Get](#)
[GodotObject.MethodName._GetPropertyList](#)
[GodotObject.MethodName._IterGet](#)
[GodotObject.MethodName._IterInit](#)
[GodotObject.MethodName._IterNext](#)
[GodotObject.MethodName._Notification](#)

GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceId
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr
GodotObject.MethodName.TrN
GodotObject.MethodName.GetTranslationDomain

GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfGodotEngine.GoogleDrive`

Assembly: CatSweeper.dll

Syntax

```
public class SettingItem.MethodName : Resource.MethodName
```

© Bus Fighter. All rights reserved.

Class SettingItem.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.PropertyName](#) → [RefCounted.PropertyName](#) → [Resource.PropertyName](#) → [SettingItem.PropertyName](#)

Inherited Members

[Resource.PropertyName.ResourceLocalToScene](#)

[Resource.PropertyName.ResourcePath](#)

[Resource.PropertyName.ResourceName](#)

[Resource.PropertyName.ResourceSceneUniqueld](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public class SettingItem.PropertyName : Resource.PropertyName
```

Fields

_fileName

Cached name for the '_fileName' field.

Declaration

```
public static readonly StringName _fileName
```

Field Value

TYPE

StringName

assetPath

Cached name for the 'assetPath' field.

Declaration

```
public static readonly StringName assetPath
```

Field Value

TYPE

StringName

driveLink

Cached name for the 'driveLink' field.

Declaration

```
public static readonly StringName driveLink
```

Field Value

TYPE

StringName

fileName

Cached name for the 'fileName' property.

Declaration

```
public static readonly StringName fileName
```

TYPE

StringName

© Bus Fighter. All rights reserved.

Class SettingItem.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

[object](#) → GodotObject.SignalName → RefCounted.SignalName → Resource.SignalName → SettingItem.SignalName

Inherited Members

Resource.SignalName.Changed

Resource.SignalName.SetupLocalToSceneRequested

GodotObject.SignalName.ScriptChanged

GodotObject.SignalName.PropertyListChanged

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public class SettingItem.SignalName : Resource.SignalName
```

© Bus Fighter. All rights reserved.

Struct SheetFileHandler

Implements

[FileHandler](#)

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public struct SheetFileHandler : FileHandler
```

Methods

DownloadAsync(FilesResource, DownloadRequest)

Declaration

```
public Task<IDownloadProgress> DownloadAsync(FilesResource filesResource, FileHandler.Download
```

Parameters

TYPE	NAME
FilesResource	filesResource
FileHandler.DownloadRequest	downloadRequest

Returns

Task<IDownloadProgress>

DownloadWithStatus(FilesResource, in DownloadRequest)

Declaration

```
public IDownloadProgress DownloadWithStatus(FilesResource filesResource, in FileHandler.DownloadRe
```

Parameters

TYPE	NAME
FilesResource	filesResource
FileHandler.DownloadRequest	downloadRequest

Returns

TYPE
IDownloadProgress

Implements

FileHandler

© Bus Fighter. All rights reserved.

Struct UrlInfo

Inherited Members

[ValueType.Equals\(object\)](#)
[ValueType.GetHashCode\(\)](#)
[ValueType.ToString\(\)](#)
[object.Equals\(object, object\)](#)
[object.GetType\(\)](#)
[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfGodotEngine.GoogleDrive](#)

Assembly: CatSweeper.dll

Syntax

```
public struct UrlInfo
```

Fields

fileId

Declaration

```
public string fileId
```

Field Value

TYPE

string

fileType

Declaration

```
public FileType fileType
```

Field Value

TYPE

[FileType](#)

Properties

Empty

Declaration

```
public static UrlInfo Empty { get; }
```

Property Value

TYPE

[UrlInfo](#)

© Bus Fighter. All rights reserved.

Namespace cfGodotEngine.Info

Classes

[ResourceInfoLoader<TInfo>](#)

[ResourceInfo<TInfo>](#)

[ResourceInfo<TInfo>.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[ResourceInfo<TInfo>.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

[ResourceInfo<TInfo>.SignalName](#)

Cached StringNames for the signals contained in this class, for fast lookup.

© Bus Fighter. All rights reserved.

Class ResourceInfoLoader<TInfo>

Inheritance

[object](#) → ResourceInfoLoader<TInfo>

Implements

[IValueLoader<TInfo>](#)

Inherited Members

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public class ResourceInfoLoader<TInfo> : IValueLoader<TInfo>
```

Type Parameters

NAME

TInfo

Constructors

ResourceInfoLoader(string)

Declaration

```
public ResourceInfoLoader(string resourcePath)
```

Parameters

TYPE	NAME
------	------

string	resourcePath
--------	--------------

Methods

Load(out List<TInfo>)

Declaration

```
public ObjectPool<List<TInfo>>.Handle Load(out List<TInfo> values)
```

Parameters

TYPE	NAME
------	------

List<TInfo>	values
-------------	--------

Returns

TYPE

ObjectPool<List<TInfo>>.Handle

LoadAsync(CancellationToken)

Declaration

```
public Task<List<TInfo>> LoadAsync(CancellationToken cancellationToken)
```

Parameters

TYPE	NAME
------	------

CancellationToken	cancellationToken
-------------------	-------------------

Returns

TYPE

Task<List<TInfo>>

Implements

[IValueLoader<TInfo>](#)

© Bus Fighter. All rights reserved.

Class ResourceInfo<TInfo>

Inheritance

[object](#) → [GodotObject](#) → [RefCounted](#) → [Resource](#) → [ResourceInfo<TInfo>](#)

Implements

[IDisposable](#)

Inherited Members

[Resource._GetRid\(\)](#)

[Resource._ResetState\(\)](#)

[**Resource._SetPathCache\(string\)**](#)

[Resource._SetupLocalToScene\(\)](#)

[**Resource.TakeOverPath\(string\)**](#)

[**Resource.SetPathCache\(string\)**](#)

[Resource.GetRid\(\)](#)

[Resource.GetLocalScene\(\)](#)

[Resource.SetupLocalToScene\(\)](#)

[Resource.ResetState\(\)](#)

[**Resource.SetIdForPath\(string, string\)**](#)

[**Resource.GetIdForPath\(string\)**](#)

[Resource.IsBuiltIn\(\)](#)

[Resource.GenerateSceneUniqueId\(\)](#)

[Resource.EmitChanged\(\)](#)

[**Resource.Duplicate\(bool\)**](#)

[Resource.EmitSignalChanged\(\)](#)

[Resource.EmitSignalSetupLocalToSceneRequested\(\)](#)

[Resource.InvokeGodotClassMethod\(in godot_string_name, NativeVariantPtrArgs, out godot_variant\)](#)

[Resource.HasGodotClassMethod\(in godot_string_name\)](#)

[Resource.HasGodotClassSignal\(in godot_string_name\)](#)

[Resource.ResourceLocalToScene](#)

[Resource.ResourcePath](#)

[Resource.ResourceName](#)

[Resource.ResourceSceneUniqueId](#)

[Resource.Changed](#)

[Resource.SetupLocalToSceneRequested](#)

[RefCounted.InitRef\(\)](#)

[RefCounted.Reference\(\)](#)

[RefCounted.Unreference\(\)](#)

[RefCounted.GetReferenceCount\(\)](#)

[GodotObject.NotificationPostInitialize](#)

GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)
GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)

GodotObject.Callv(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObject.GetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Namespace: [cfGodotEngine.Info](#)

Assembly: CatSweeper.dll

Syntax

```
[ScriptPath("res://Modules/cfGodotEngine/info/ResourceInfo.cs")]
public abstract class ResourceInfo<TInfo> : Resource, IDisposable
```

Type Parameters

NAME

TInfo

Properties

Declaration

```
public abstract IEnumerable<TInfo> GetInfos { get; }
```

Property Value

TYPE

IEnumerable<TInfo>

Implements

IDisposable

© Bus Fighter. All rights reserved.

Class ResourceInfo<TInfo>.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject](#).[MethodName](#) → [RefCounted](#).[MethodName](#) → [Resource](#).[MethodName](#) → [ResourceInfo<TInfo>.MethodName](#)

Inherited Members

[Resource](#).[MethodName](#).[_GetRid](#)
[Resource](#).[MethodName](#).[_ResetState](#)
[Resource](#).[MethodName](#).[_SetPathCache](#)
[Resource](#).[MethodName](#).[_SetupLocalToScene](#)
[Resource](#).[MethodName](#).[SetPath](#)
[Resource](#).[MethodName](#).[TakeOverPath](#)
[Resource](#).[MethodName](#).[GetPath](#)
[Resource](#).[MethodName](#).[SetPathCache](#)
[Resource](#).[MethodName](#).[SetName](#)
[Resource](#).[MethodName](#).[GetName](#)
[Resource](#).[MethodName](#).[GetRid](#)
[Resource](#).[MethodName](#).[SetLocalToScene](#)
[Resource](#).[MethodName](#).[IsLocalToScene](#)
[Resource](#).[MethodName](#).[GetLocalScene](#)
[Resource](#).[MethodName](#).[SetupLocalToScene](#)
[Resource](#).[MethodName](#).[ResetState](#)
[Resource](#).[MethodName](#).[SetIdForPath](#)
[Resource](#).[MethodName](#).[GetIdForPath](#)
[Resource](#).[MethodName](#).[IsBuiltIn](#)
[Resource](#).[MethodName](#).[GenerateSceneUniqueId](#)
[Resource](#).[MethodName](#).[SetSceneUniqueId](#)
[Resource](#).[MethodName](#).[GetSceneUniqueId](#)
[Resource](#).[MethodName](#).[EmitChanged](#)
[Resource](#).[MethodName](#).[Duplicate](#)
[RefCounted](#).[MethodName](#).[InitRef](#)
[RefCounted](#).[MethodName](#).[Reference](#)
[RefCounted](#).[MethodName](#).[Unreference](#)
[RefCounted](#).[MethodName](#).[GetReferenceCount](#)
[GodotObject](#).[MethodName](#).[_Get](#)
[GodotObject](#).[MethodName](#).[_GetPropertyList](#)
[GodotObject](#).[MethodName](#).[_IterGet](#)
[GodotObject](#).[MethodName](#).[_IterInit](#)
[GodotObject](#).[MethodName](#).[_IterNext](#)

GodotObject.MethodName._Notification
GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceId
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyListChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr
GodotObject.MethodName.TrN

GodotObject.MethodName.GetTranslationDomain
GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfGodotEngine.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public class ResourceInfo<TInfo>.MethodName : Resource.MethodName
```

© Bus Fighter. All rights reserved.

Class ResourceInfo<TInfo>.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.PropertyName](#) → [RefCounted.PropertyName](#) → [Resource.PropertyName](#) → [ResourceInfo<TInfo>.PropertyName](#)

Inherited Members

[Resource.PropertyName.ResourceLocalToScene](#)

[Resource.PropertyName.ResourcePath](#)

[Resource.PropertyName.ResourceName](#)

[Resource.PropertyName.ResourceSceneUniqueld](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Info](#)

Assembly: CatSweeper.dll

Syntax

```
public class ResourceInfo<TInfo>.PropertyName : Resource.PropertyName
```

© Bus Fighter. All rights reserved.

Class ResourceInfo<TInfo>.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.SignalName](#) → [RefCounted.SignalName](#) → [Resource.SignalName](#) → [ResourceInfo<TInfo>.SignalName](#)

Inherited Members

[Resource.SignalName.Changed](#)

[Resource.SignalName.SetupLocalToSceneRequested](#)

[GodotObject.SignalName.ScriptChanged](#)

[GodotObject.SignalName.PropertyListChanged](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Info](#)

Assembly: [CatSweeper.dll](#)

Syntax

```
public class ResourceInfo<TInfo>.SignalName : Resource.SignalName
```

© Bus Fighter. All rights reserved.

Namespace cfGodotEngine.SceneManagement

Classes

[GodotSceneManager](#)

[GodotSceneManager.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[GodotSceneManager.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

[GodotSceneManager.SignalName](#)

Cached StringNames for the signals contained in this class, for fast lookup.

Interfaces

[ISceneManager<TScene>](#)

Enums

[LoadSceneMode](#)

Class GodotSceneManager

Inheritance

object → GodotObject → Node → [MonoInstance<GodotSceneManager>](#) → GodotSceneManager

Implements

[ISceneManager<Node>](#)

[IService](#)

[IDisposable](#)

Inherited Members

[MonoInstance<GodotSceneManager>.Instance](#)

Node.NotificationEnterTree

Node.NotificationExitTree

Node.NotificationMovedInParent

Node.NotificationReady

Node.NotificationPaused

Node.NotificationUnpaused

Node.NotificationPhysicsProcess

Node.NotificationProcess

Node.NotificationParented

Node.NotificationUnparented

Node.NotificationSceneInstantiated

Node.NotificationDragBegin

Node.NotificationDragEnd

Node.NotificationPathRenamed

Node.NotificationChildOrderChanged

Node.NotificationInternalProcess

Node.NotificationInternalPhysicsProcess

Node.NotificationPostEnterTree

Node.NotificationDisabled

Node.NotificationEnabled

Node.NotificationResetPhysicsInterpolation

Node.NotificationEditorPreSave

Node.NotificationEditorPostSave

Node.NotificationWMMouseEnter

Node.NotificationWMMouseExit

Node.NotificationWMWindowFocusIn

Node.NotificationWMWindowFocusOut

Node.NotificationWMCloseRequest

Node.NotificationWMGoBackRequest

Node.NotificationWMSizeChanged
Node.NotificationWMDpiChange
Node.NotificationVpMouseEnter
Node.NotificationVpMouseExit
Node.NotificationWMPositionChanged
Node.NotificationOsMemoryWarning
Node.NotificationTranslationChanged
Node.NotificationWMAbout
Node.NotificationCrash
Node.NotificationOslmeUpdate
Node.NotificationApplicationResumed
Node.NotificationApplicationPaused
Node.NotificationApplicationFocusIn
Node.NotificationApplicationFocusOut
Node.NotificationTextServerChanged
Node.GetNode<T>(NodePath)
Node.GetNodeOrNull<T>(NodePath)
Node.GetChild<T>(int, bool)
Node.GetChildOrNull<T>(int, bool)
Node.GetOwner<T>()
Node.GetOwnerOrNull<T>()
Node.GetParent<T>()
Node.GetParentOrNull<T>()
Node._EnterTree()
Node._ExitTree()
Node._GetConfigurationWarnings()
Node._Input(InputEvent)
Node._PhysicsProcess(double)
Node._Ready()
Node._ShortcutInput(InputEvent)
Node._UnhandledInput(InputEvent)
Node._UnhandledKeyInput(InputEvent)
Node.PrintOrphanNodes()
Node.AddSibling(Node, bool)
Node.AddChild(Node, bool, Node.InternalMode)
Node.RemoveChild(Node)
Node.Reparent(Node, bool)
Node.GetChildCount(bool)
Node.GetChildren(bool)
Node.GetChild(int, bool)
Node.HasNode(NodePath)
Node.GetNode(NodePath)
Node.GetNodeOrNull(NodePath)
Node.GetParent()
Node.FindChild(string, bool, bool)
Node.FindChildren(string, string, bool, bool)
Node.FindParent(string)
Node.HasNodeAndResource(NodePath)
Node.GetNodeAndResource(NodePath)
Node.IsInsideTree()

Node.IsPartOfEditedScene()
Node.IsAncestorOf(Node)
Node.IsGreater Than(Node)
Node.GetPath()
Node.GetPathTo(Node, bool)
Node.AddToGroup(StringName, bool)
Node.RemoveFromGroup(StringName)
Node.IsInGroup(StringName)
Node.MoveChild(Node, int)
Node.GetGroups()
Node.GetIndex(bool)
Node.PrintTree()
Node.PrintTreePretty()
Node.GetTreeString()
Node.GetTreeStringPretty()
Node.PropagateNotification(int)
Node.PropagateCall(StringName, Array, bool)
Node.SetPhysicsProcess(bool)
Node.GetPhysicsProcessDelta Time()
Node.IsPhysicsProcessing()
Node.GetProcessDelta Time()
Node.SetProcess(bool)
Node.IsProcessing()
Node.SetProcessInput(bool)
Node.IsProcessingInput()
Node.SetProcessShortcutInput(bool)
Node.IsProcessingShortcutInput()
Node.SetProcessUnhandledInput(bool)
Node.IsProcessingUnhandledInput()
Node.SetProcessUnhandledKeyInput(bool)
Node.IsProcessingUnhandledKeyInput()
Node.CanProcess()
Node.SetDisplayFolded(bool)
Node.IsDisplayedFolded()
Node.SetProcessInternal(bool)
Node.IsProcessingInternal()
Node.SetPhysicsProcessInternal(bool)
Node.IsPhysicsProcessingInternal()
Node.IsPhysicsInterpolated()
Node.IsPhysicsInterpolatedAndEnabled()
Node.ResetPhysicsInterpolation()
Node.SetTranslationDomainInherited()
Node.GetWindow()
Node.GetLastExclusiveWindow()
Node.GetTree()
Node.CreateTween()
Node.Duplicate(int)
Node.ReplaceBy(Node, bool)
Node.setSceneInstanceLoadPlaceholder(bool)
Node.GetSceneInstanceLoadPlaceholder()

Node.setEditableInstance(Node, bool)
Node.setEditableInstance(Node)
Node.setViewport()
Node.queueFree()
Node.requestReady()
Node.isNodeReady()
Node.setMultiplayerAuthority(int, bool)
Node.getMultiplayerAuthority()
Node.isMultiplayerAuthority()
Node.rpcConfig(StringName, Variant)
Node.getRpcConfig()
Node.Atr(string, StringName)
Node.AtrN(string, StringName, int, StringName)
Node.rpc(StringName, params Variant[])
Node.Rpc(StringName, ReadOnlySpan<Variant>)
Node.RpcId(long, StringName, params Variant[])
Node.RpcId(long, StringName, ReadOnlySpan<Variant>)
Node.updateConfigurationWarnings()
Node.callDeferredThreadGroup(StringName, params Variant[])
Node.CallDeferredThreadGroup(StringName, ReadOnlySpan<Variant>)
Node.setDeferredThreadGroup(StringName, Variant)
Node.NotifyDeferredThreadGroup(int)
Node.callThreadSafe(StringName, params Variant[])
Node.CallThreadSafe(StringName, ReadOnlySpan<Variant>)
Node.setThreadSafe(StringName, Variant)
Node.NotifyThreadSafe(int)
Node.emitSignalReady()
Node.emitSignalRenamed()
Node.emitSignalTreeEntered()
Node.emitSignalTreeExiting()
Node.emitSignalTreeExited()
Node.emitSignalChildEnteredTree(Node)
Node.emitSignalChildExitingTree(Node)
Node.emitSignalChildOrderChanged()
Node.emitSignalReplacingBy(Node)
Node.emitSignalEditorDescriptionChanged(Node)
Node.emitSignalEditorStateChanged()
Node.invokeGodotClassMethod(in godot_string_name, NativeVariantPtrArgs, out godot_variant)
Node.hasGodotClassMethod(in godot_string_name)
Node.hasGodotClassSignal(in godot_string_name)
Node.name
Node.uniqueNameInOwner
Node.sceneFilePath
Node.owner
Node.multiplayer
Node.processMode
Node.processPriority
Node.processPhysicsPriority
Node.processThreadGroup
Node.processThreadGroupOrder

Node.ProcessThreadMessages
Node.PhysicsInterpolationMode
Node.AutoTranslateMode
Node.EditorDescription
Node.Ready
Node.Renamed
Node.TreeEntered
Node.TreeExiting
Node.TreeExited
Node.ChildEnteredTree
Node.ChildExitingTree
Node.ChildOrderChanged
Node.ReplacingBy
Node.EditorDescriptionChanged
Node.EditorStateChanged
GodotObject.NotificationPostinitialize
GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)

GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)
GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)
GodotObject.Callv(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObjectGetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Syntax

```
[ScriptPath("res://Modules/cfGodotEngine/scene/GodotSceneManager.cs")]
public class GodotSceneManager : MonoInstance<GodotSceneManager>, ISceneManager<Node>, IServic
```

Constructors

GodotSceneManager()

Declaration

```
public GodotSceneManager()
```

Methods

GetScene(string)

Declaration

```
public Node GetScene(string sceneName)
```

Parameters

TYPE	NAME
------	------

string	sceneName
--------	-----------

Returns

TYPE

Node

LoadScene(string, LoadSceneMode)

Declaration

```
public Node LoadScene(string sceneKey, LoadSceneMode mode = LoadSceneMode.Single)
```

Parameters

TYPE	NAME
string	sceneKey
LoadSceneMode	mode

Returns

TYPE
Node

LoadSceneAsync(string, LoadSceneMode, IProgress<float>)

Declaration

```
public Task<Node> LoadSceneAsync(string sceneKey, LoadSceneMode mode = LoadSceneMode.Single, I
```

Parameters

TYPE	NAME
string	sceneKey
LoadSceneMode	mode
IProgress<float>	progress

Returns

TYPE
Task<Node>

_Process(double)

Called during the processing step of the main loop. Processing happens at every frame and as fast as possible, so the `delta` time since the previous frame is not constant. `delta` is in seconds.

It is only called if processing is enabled, which is done automatically if this method is overridden, and can be toggled with `Set Process(bool)`.

Processing happens in order of [Godot.Node.ProcessPriority](#), lower priority values are called first. Nodes with the same priority are processed in tree order, or top to bottom as seen in the editor (also known as pre-order traversal).

Corresponds to the [Godot.Node.NotificationProcess](#) notification in [_Notification\(int\)](#).

Note: This method is only called if the node is present in the scene tree (i.e. if it's not an orphan).

Note: `delta` will be larger than expected if running at a framerate lower than [Godot.Engine.PhysicsTicksPerSecond](#) / [Godot.Engine.MaxPhysicsStepsPerFrame](#) FPS. This is done to avoid "spiral of death" scenarios where performance would plummet due to an ever-increasing number of physics steps per frame. This behavior affects both [_Process\(double\)](#) and [_PhysicsProcess\(double\)](#). As a result, avoid using `delta` for time measurements in real-world seconds. Use the [Godot.Time](#) singleton's methods for this purpose instead, such as [Godot.Time.GetTicksUseC\(\)](#).

Declaration

```
public override void _Process(double delta)
```

Parameters

TYPE	NAME
double	delta

Overrides

[Node._Process\(double\)](#)

Implements

[ISceneManager<TScene>](#)
[IService](#)
[IDisposable](#)

Extension Methods

[NodeUtil.DontDestroyOnLoad\(Node\)](#)

© Bus Fighter. All rights reserved.

Class GodotSceneManager.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

`object` → `GodotObject.MethodName` → `Node.MethodName` → `MonoInstance<GodotSceneManager>.MethodName` → `GodotSceneManager.MethodName`

Inherited Members

`Node.MethodName._EnterTree`
`Node.MethodName._ExitTree`
`Node.MethodName._GetConfigurationWarnings`
`Node.MethodName._Input`
`Node.MethodName._PhysicsProcess`
`Node.MethodName._Ready`
`Node.MethodName._ShortcutInput`
`Node.MethodName._UnhandledInput`
`Node.MethodName._UnhandledKeyInput`
`Node.MethodName.PrintOrphanNodes`
`Node.MethodName.AddSibling`
`Node.MethodName.SetName`
`Node.MethodName.GetName`
`Node.MethodName.AddChild`
`Node.MethodName.RemoveChild`
`Node.MethodName.Reparent`
`Node.MethodName.GetChildCount`
`Node.MethodName.GetChildren`
`Node.MethodName.GetChild`
`Node.MethodName.HasNode`
`Node.MethodName.GetNode`
`Node.MethodName.GetNodeOrNull`
`Node.MethodName.GetParent`
`Node.MethodName.FindChild`
`Node.MethodName.FindChildren`
`Node.MethodName.FindParent`
`Node.MethodName.HasNodeAndResource`
`Node.MethodName.GetNodeAndResource`
`Node.MethodName.IsInsideTree`
`Node.MethodName.IsPartOfEditedScene`
`Node.MethodName.IsAncestorOf`
`Node.MethodName.IsGreaterThanOrEqual`
`Node.MethodName.GetPath`

Node.MethodName.GetPathTo
Node.MethodName.AddToGroup
Node.MethodName.RemoveFromGroup
Node.MethodName.IsInGroup
Node.MethodName.MoveChild
Node.MethodName.GetGroups
Node.MethodName.SetOwner
Node.MethodName.GetOwner
Node.MethodName.GetIndex
Node.MethodName.PrintTree
Node.MethodName.PrintTreePretty
Node.MethodName.GetTreeString
Node.MethodName.GetTreeStringPretty
Node.MethodName.SetSceneFilePath
Node.MethodName.GetSceneFilePath
Node.MethodName.PropagateNotification
Node.MethodName.PropagateCall
Node.MethodName.SetPhysicsProcess
Node.MethodName.GetPhysicsProcessDeltaTime
Node.MethodName.IsPhysicsProcessing
Node.MethodName.GetProcessDeltaTime
Node.MethodName.SetProcess
Node.MethodName.SetProcessPriority
Node.MethodName.GetProcessPriority
Node.MethodName.SetPhysicsProcessPriority
Node.MethodName.GetPhysicsProcessPriority
Node.MethodName.IsProcessing
Node.MethodName.SetProcessInput
Node.MethodName.IsProcessingInput
Node.MethodName.SetProcessShortcutInput
Node.MethodName.IsProcessingShortcutInput
Node.MethodName.SetProcessUnhandledInput
Node.MethodName.IsProcessingUnhandledInput
Node.MethodName.SetProcessUnhandledKeyInput
Node.MethodName.IsProcessingUnhandledKeyInput
Node.MethodName.SetProcessMode
Node.MethodName.GetProcessMode
Node.MethodName.CanProcess
Node.MethodName.SetProcessThreadGroup
Node.MethodName.GetProcessThreadGroup
Node.MethodName.SetProcessThreadMessages
Node.MethodName.GetProcessThreadMessages
Node.MethodName.SetProcessThreadGroupOrder
Node.MethodName.GetProcessThreadGroupOrder
Node.MethodName.SetDisplayFolded
Node.MethodName.IsDisplayedFolded
Node.MethodName.SetProcessInternal
Node.MethodName.IsProcessingInternal
Node.MethodName.SetPhysicsProcessInternal
Node.MethodName.IsPhysicsProcessingInternal

Node.MethodName.SetPhysicsInterpolationMode
Node.MethodName.GetPhysicsInterpolationMode
Node.MethodName.IsPhysicsInterpolated
Node.MethodName.IsPhysicsInterpolatedAndEnabled
Node.MethodName.ResetPhysicsInterpolation
Node.MethodName.SetAutoTranslateMode
Node.MethodName.GetAutoTranslateMode
Node.MethodName.SetTranslationDomainInherited
Node.MethodName.GetWindow
Node.MethodName.GetLastExclusiveWindow
Node.MethodName.GetTree
Node.MethodName.CreateTween
Node.MethodName.Duplicate
Node.MethodName.ReplaceBy
Node.MethodName.SetSceneInstanceLoadPlaceholder
Node.MethodName.GetSceneInstanceLoadPlaceholder
Node.MethodName.SetEditableInstance
Node.MethodName.IsEditableInstance
Node.MethodName.GetViewport
Node.MethodName.QueueFree
Node.MethodName.RequestReady
Node.MethodName.IsNodeReady
Node.MethodName.SetMultiplayerAuthority
Node.MethodName.GetMultiplayerAuthority
Node.MethodName.IsMultiplayerAuthority
Node.MethodName.GetMultiplayer
Node.MethodName.RpcConfig
Node.MethodName.GetRpcConfig
Node.MethodName.SetEditorDescription
Node.MethodName.GetEditorDescription
Node.MethodName._SetImportPath
Node.MethodName._GetImportPath
Node.MethodName.SetUniqueNameInOwner
Node.MethodName.IsUniqueNameInOwner
Node.MethodName.Atr
Node.MethodName.AtrN
Node.MethodName.Rpc
Node.MethodName.RpcId
Node.MethodName.UpdateConfigurationWarnings
Node.MethodName.CallDeferredThreadGroup
Node.MethodName.SetDeferredThreadGroup
Node.MethodName.NotifyDeferredThreadGroup
Node.MethodName.CallThreadSafe
Node.MethodName.SetThreadSafe
Node.MethodName.NotifyThreadSafe
GodotObject.MethodName._Get
GodotObject.MethodName._GetPropertyList
GodotObject.MethodName._IterGet
GodotObject.MethodName._IterInit
GodotObject.MethodName._IterNext

GodotObject.MethodName._Notification
GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceId
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyListChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr
GodotObject.MethodName.TrN

GodotObject.MethodName.GetTranslationDomain
GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfGodotEngine.SceneManagement](#)

Assembly: CatSweeper.dll

Syntax

```
public class GodotSceneManager.MethodName : MonoInstance<GodotSceneManager>.MethodName
```

Fields

GetScene

Cached name for the 'GetScene' method.

Declaration

```
public static readonly StringName GetScene
```

Field Value

TYPE

StringName

GetSceneTree

Cached name for the 'GetSceneTree' method.

Declaration

```
public static readonly StringName GetSceneTree
```

Field Value

TYPE

StringName

LoadScene

Cached name for the 'LoadScene' method.

Declaration

```
public static readonly StringName LoadScene
```

Field Value

TYPE

StringName

ShowScene

Cached name for the 'ShowScene' method.

Declaration

```
public static readonly StringName ShowScene
```

Field Value

TYPE

StringName

_Process

Cached name for the '_Process' method.

Declaration

```
public static readonly StringName _Process
```

TYPE

StringName

© Bus Fighter. All rights reserved.

Class GodotSceneManager.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.PropertyName](#) → [Node.PropertyName](#) → [MonoInstance<GodotSceneManager>.PropertyName](#) → [GodotSceneManager.PropertyName](#)

Inherited Members

[Node.PropertyName._ImportPath](#)
[Node.PropertyName.Name](#)
[Node.PropertyName.UniqueNameInOwner](#)
[Node.PropertyName.SceneFilePath](#)
[Node.PropertyName.Owner](#)
[Node.PropertyName.Multiplayer](#)
[Node.PropertyName.ProcessMode](#)
[Node.PropertyName.ProcessPriority](#)
[Node.PropertyName.ProcessPhysicsPriority](#)
[Node.PropertyName.ProcessThreadGroup](#)
[Node.PropertyName.ProcessThreadGroupOrder](#)
[Node.PropertyName.ProcessThreadMessages](#)
[Node.PropertyName.PhysicsInterpolationMode](#)
[Node.PropertyName.AutoTranslateMode](#)
[Node.PropertyName.EditorDescription](#)
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfGodotEngine.SceneManagement](#)

Assembly: CatSweeper.dll

Syntax

```
public class GodotSceneManager.PropertyName : MonoInstance<GodotSceneManager>.PropertyName
```

Fields

progressArray

Cached name for the 'progressArray' field.

Declaration

```
public static readonly StringName progressArray
```

Field Value

TYPE

StringName

© Bus Fighter. All rights reserved.

Class GodotSceneManager.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.SignalName](#) → [Node.SignalName](#) → [MonoInstance<GodotSceneManager>.SignalName](#) → [GodotSceneManager.SignalName](#)

Inherited Members

[Node.SignalName.Ready](#)
[Node.SignalName.Renamed](#)
[Node.SignalName.TreeEntered](#)
[Node.SignalName.TreeExiting](#)
[Node.SignalName.TreeExited](#)
[Node.SignalName.ChildEnteredTree](#)
[Node.SignalName.ChildExitingTree](#)
[Node.SignalName.ChildOrderChanged](#)
[Node.SignalName.ReplacingBy](#)
[Node.SignalName.EditorDescriptionChanged](#)
[Node.SignalName.EditorStateChanged](#)
[GodotObject.SignalName.ScriptChanged](#)
[GodotObject.SignalName.PropertyListChanged](#)
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfGodotEngine.SceneManagement](#)

Assembly: CatSweeper.dll

Syntax

```
public class GodotSceneManager.SignalName : MonoInstance<GodotSceneManager>.SignalName
```

Interface ISceneManager<TScene>

Inherited Members

[IDisposable.Dispose\(\)](#)

Namespace: [cfGodotEngine.SceneManagement](#)

Assembly: CatSweeper.dll

Syntax

```
public interface ISceneManager<TScene> : IService, IDisposable
```

Type Parameters

NAME

TScene

Methods

GetScene(string)

Declaration

```
TScene GetScene(string sceneName)
```

Parameters

TYPE NAME

string sceneName

Returns

TYPE

TScene

LoadScene(string, LoadSceneMode)

Declaration

```
TScene LoadScene(string sceneKey, LoadSceneMode mode = LoadSceneMode.Single)
```

Parameters

TYPE	NAME
string	sceneKey
LoadSceneMode	mode

Returns

TYPE
TScene

LoadSceneAsync(string, LoadSceneMode, IProgress<float>)

Declaration

```
Task<TScene> LoadSceneAsync(string sceneKey, LoadSceneMode mode = LoadSceneMode.Single, IProgr
```

Parameters

TYPE	NAME
string	sceneKey
LoadSceneMode	mode
IProgress<float>	progress

Returns

TYPE
Task<TScene>

Enum LoadSceneMode

Namespace: [cfGodotEngine.SceneManagement](#)

Assembly: CatSweeper.dll

Syntax

```
public enum LoadSceneMode
```

Fields

NAME
Additive
Single

Extension Methods

[EnumExtension.hasFlag\(Enum, Enum\)](#)

© Bus Fighter. All rights reserved.

Namespace cfGodotEngine.Util

Classes

[Application](#)

[MonoInstance<T>](#)

[MonoInstance<T>.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[MonoInstance<T>.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

[MonoInstance<T>.SignalName](#)

Cached StringNames for the signals contained in this class, for fast lookup.

[NodeUtil](#)

[SettingPath](#)

[Setting<T>](#)

[Setting<T>.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[Setting<T>.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

[Setting<T>.SignalName](#)

Cached StringNames for the signals contained in this class, for fast lookup.

[StateExecutionException<TStatId>](#)

[StateMachineNode<TStatId, TState, TStateMachine>](#)

[StateMachineNode<TStatId, TState, TStateMachine>.MethodName](#)

Cached StringNames for the methods contained in this class, for fast lookup.

[StateMachineNode<TStatId, TState, TStateMachine>.PropertyName](#)

Cached StringNames for the properties and fields contained in this class, for fast lookup.

StateMachineNode<TStatId, TState, TStateMachine>.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

StateNode<TStatId, TState, TStateMachine>

StateNode<TStatId, TState, TStateMachine>.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

StateNode<TStatId, TState, TStateMachine>.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

StateNode<TStatId, TState, TStateMachine>.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

© Bus Fighter. All rights reserved.

Class Application

Inheritance

`object` → Application

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public static class Application
```

Properties

assetDataPath

Declaration

```
public static string assetDataPath { get; }
```

Property Value

TYPE

`string`

exportDataPath

Declaration

```
public static string exportDataPath { get; }
```

Property Value

TYPE

string

persistentDataPath

Declaration

```
public static string persistentDataPath { get; }
```

Property Value

TYPE

string

Methods

GetGlobalizePath(string)

Declaration

```
public static string GetGlobalizePath(string path)
```

Parameters

TYPE NAME

string path

Returns

© Bus Fighter. All rights reserved.

Class MonoInstance<T>

Inheritance

[object](#) → [GodotObject](#) → [Node](#) → [MonoInstance<T>](#) → [AsyncResourceLoader](#) → [GodotSceneManager](#)

Implements

[IDisposable](#)

Inherited Members

Node.NotificationEnterTree
Node.NotificationExitTree
Node.NotificationMovedInParent
Node.NotificationReady
Node.NotificationPaused
Node.NotificationUnpaused
Node.NotificationPhysicsProcess
Node.NotificationProcess
Node.NotificationParented
Node.NotificationUnparented
Node.NotificationSceneInstantiated
Node.NotificationDragBegin
Node.NotificationDragEnd
Node.NotificationPathRenamed
Node.NotificationChildOrderChanged
Node.NotificationInternalProcess
Node.NotificationInternalPhysicsProcess
Node.NotificationPostEnterTree
Node.NotificationDisabled
Node.NotificationEnabled
Node.NotificationResetPhysicsInterpolation
Node.NotificationEditorPreSave
Node.NotificationEditorPostSave
Node.NotificationWMMouseEnter
Node.NotificationWMMouseExit
Node.NotificationWMWindowFocusIn
Node.NotificationWMWindowFocusOut
Node.NotificationWMCloseRequest
Node.NotificationWMGoBackRequest
Node.NotificationWMSizeChanged
Node.NotificationWMDpiChange
Node.NotificationVpMouseEnter

Node.NotificationVpMouseExit
Node.NotificationWMPositionChanged
Node.NotificationOsMemoryWarning
Node.NotificationTranslationChanged
Node.NotificationWMAbout
Node.NotificationCrash
Node.NotificationOslmeUpdate
Node.NotificationApplicationResumed
Node.NotificationApplicationPaused
Node.NotificationApplicationFocusIn
Node.NotificationApplicationFocusOut
Node.NotificationTextServerChanged
Node.GetNode<T>(NodePath)
Node.GetNodeOrNull<T>(NodePath)
Node.GetChild<T>(int, bool)
Node.GetChildOrNull<T>(int, bool)
Node.GetOwner<T>()
Node.GetOwnerOrNull<T>()
Node.GetParent<T>()
Node.GetParentOrNull<T>()
Node._EnterTree()
Node._ExitTree()
Node._GetConfigurationWarnings()
Node._Input(InputEvent)
Node._PhysicsProcess(double)
Node._Process(double)
Node._Ready()
Node._ShortcutInput(InputEvent)
Node._UnhandledInput(InputEvent)
Node._UnhandledKeyInput(InputEvent)
Node.PrintOrphanNodes()
Node.AddSibling(Node, bool)
Node.AddChild(Node, bool, Node.InternalMode)
Node.RemoveChild(Node)
Node.Reparent(Node, bool)
Node.GetChildCount(bool)
Node.GetChildren(bool)
Node.GetChild(int, bool)
Node.HasNode(NodePath)
Node.GetNode(NodePath)
Node.GetNodeOrNull(NodePath)
Node.GetParent()
Node.FindChild(string, bool, bool)
Node.FindChildren(string, string, bool, bool)
Node.FindParent(string)
Node.HasNodeAndResource(NodePath)
Node.GetNodeAndResource(NodePath)
Node.IsInsideTree()
Node.IsPartOfEditedScene()
Node.IsAncestorOf(Node)

Node.IsGreater Than(Node)
Node.GetPath()
Node.GetPath To(Node, bool)
Node.AddToGroup(StringName, bool)
Node.RemoveFromGroup(StringName)
Node.IsInGroup(StringName)
Node.MoveChild(Node, int)
Node.GetGroups()
Node.GetIndex(bool)
Node.PrintTree()
Node.PrintTreePretty()
Node.GetTreeString()
Node.GetTreeStringPretty()
Node.PropagateNotification(int)
Node.PropagateCall(StringName, Array, bool)
Node.SetPhysicsProcess(bool)
Node.GetPhysicsProcessDelta Time()
Node.IsPhysicsProcessing()
Node.GetProcessDelta Time()
Node.SetProcess(bool)
Node.IsProcessing()
Node.SetProcessInput(bool)
Node.IsProcessingInput()
Node.SetProcessShortcutInput(bool)
Node.IsProcessingShortcutInput()
Node.SetProcessUnhandledInput(bool)
Node.IsProcessingUnhandledInput()
Node.SetProcessUnhandledKeyInput(bool)
Node.IsProcessingUnhandledKeyInput()
Node.CanProcess()
Node.SetDisplayFolded(bool)
Node.IsDisplayedFolded()
Node.SetProcessInternal(bool)
Node.IsProcessingInternal()
Node.SetPhysicsProcessInternal(bool)
Node.IsPhysicsProcessingInternal()
Node.IsPhysicsInterpolated()
Node.IsPhysicsInterpolatedAndEnabled()
Node.ResetPhysicsInterpolation()
Node.SetTranslationDomainInherited()
Node.GetWindow()
Node.GetLastExclusiveWindow()
Node.GetTree()
Node.CreateTween()
Node.Duplicate(int)
Node.ReplaceBy(Node, bool)
Node.setSceneInstanceLoadPlaceholder(bool)
Node.GetSceneInstanceLoadPlaceholder()
Node.setEditableInstance(Node, bool)
Node.setEditableInstance(Node)

```
Node.GetViewport()
Node.QueueFree()
Node.RequestReady()
Node.IsNodeReady()
Node.SetMultiplayerAuthority(int, bool)
Node.GetMultiplayerAuthority()
Node.IsMultiplayerAuthority()
Node.RpcConfig(StringName, Variant)
Node.GetRpcConfig()
Node.Atr(string, StringName)
Node.AtrN(string, StringName, int, StringName)
Node.Rpc(StringName, params Variant[])
Node.Rpc(StringName, ReadOnlySpan<Variant>)
Node.RpcId(long, StringName, params Variant[])
Node.RpcId(long, StringName, ReadOnlySpan<Variant>)
Node.UpdateConfigurationWarnings()
Node.CallDeferredThreadGroup(StringName, params Variant[])
Node.CallDeferredThreadGroup(StringName, ReadOnlySpan<Variant>)
Node.SetDeferredThreadGroup(StringName, Variant)
Node.NotifyDeferredThreadGroup(int)
Node.CallThreadSafe(StringName, params Variant[])
Node.CallThreadSafe(StringName, ReadOnlySpan<Variant>)
Node.SetThreadSafe(StringName, Variant)
Node.NotifyThreadSafe(int)
Node.EmitSignalReady()
Node.EmitSignalRenamed()
Node.EmitSignalTreeEntered()
Node.EmitSignalTreeExiting()
Node.EmitSignalTreeExited()
Node.EmitSignalChildEnteredTree(Node)
Node.EmitSignalChildExitingTree(Node)
Node.EmitSignalChildOrderChanged()
Node.EmitSignalReplacingBy(Node)
Node.EmitSignalEditorDescriptionChanged(Node)
Node.EmitSignalEditorStateChanged()
Node.InvokeGodotClassMethod(in godot_string_name, NativeVariantPtrArgs, out godot_variant)
Node.HasGodotClassMethod(in godot_string_name)
Node.HasGodotClassSignal(in godot_string_name)
Node.Name
Node.UniqueNameInOwner
Node.SceneFilePath
Node.Owner
Node.Multiplayer
Node.ProcessMode
Node.ProcessPriority
Node.ProcessPhysicsPriority
Node.ProcessThreadGroup
Node.ProcessThreadGroupOrder
Node.ProcessThreadMessages
Node.PhysicsInterpolationMode
```

Node.AutoTranslateMode
Node.EditorDescription
Node.Ready
Node.Renamed
Node.TreeEntered
Node.TreeExiting
Node.TreeExited
Node.ChildEnteredTree
Node.ChildExitingTree
Node.ChildOrderChanged
Node.ReplacingBy
Node.EditorDescriptionChanged
Node.EditorStateChanged
GodotObject.NotificationPostinitialize
GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)

GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)
GodotObject.Cally(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObjectGetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Namespace: **cfGodotEngine.Util**

Assembly: CatSweeper.dll

Syntax

```
[ScriptPath("res://Modules/cfGodotEngine/util/MonoInstance.cs")]
public abstract class MonoInstance<T> : Node, IDisposable where T : MonoInstance<T>, new()
```

Type Parameters

NAME

T

Constructors

MonoInstance()

Declaration

```
protected MonoInstance()
```

Properties

Instance

Declaration

```
public static T Instance { get; }
```

Property Value

TYPE

T

Implements

IDisposable

Extension Methods

[NodeUtil.DontDestroyOnLoad\(Node\)](#)

© Bus Fighter. All rights reserved.

Class MonoInstance<T>.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject](#).[MethodName](#) → [Node](#).[MethodName](#) → [MonoInstance<T>](#).[MethodName](#) →
[AsyncResourceLoader](#).[MethodName](#) → [GodotSceneManager](#).[MethodName](#)

Inherited Members

[Node](#).[MethodName](#).[_EnterTree](#)
[Node](#).[MethodName](#).[_ExitTree](#)
[Node](#).[MethodName](#).[_GetConfigurationWarnings](#)
[Node](#).[MethodName](#).[_Input](#)
[Node](#).[MethodName](#).[_PhysicsProcess](#)
[Node](#).[MethodName](#).[_Process](#)
[Node](#).[MethodName](#).[_Ready](#)
[Node](#).[MethodName](#).[_ShortcutInput](#)
[Node](#).[MethodName](#).[_UnhandledInput](#)
[Node](#).[MethodName](#).[_UnhandledKeyInput](#)
[Node](#).[MethodName](#).[PrintOrphanNodes](#)
[Node](#).[MethodName](#).[AddSibling](#)
[Node](#).[MethodName](#).[SetName](#)
[Node](#).[MethodName](#).[GetName](#)
[Node](#).[MethodName](#).[AddChild](#)
[Node](#).[MethodName](#).[RemoveChild](#)
[Node](#).[MethodName](#).[Reparent](#)
[Node](#).[MethodName](#).[GetChildCount](#)
[Node](#).[MethodName](#).[GetChildren](#)
[Node](#).[MethodName](#).[GetChild](#)
[Node](#).[MethodName](#).[HasNode](#)
[Node](#).[MethodName](#).[GetNode](#)
[Node](#).[MethodName](#).[GetNodeOrNull](#)
[Node](#).[MethodName](#).[GetParent](#)
[Node](#).[MethodName](#).[FindChild](#)
[Node](#).[MethodName](#).[FindChildren](#)
[Node](#).[MethodName](#).[FindParent](#)
[Node](#).[MethodName](#).[HasNodeAndResource](#)
[Node](#).[MethodName](#).[GetNodeAndResource](#)
[Node](#).[MethodName](#).[IsInsideTree](#)
[Node](#).[MethodName](#).[IsPartOfEditedScene](#)
[Node](#).[MethodName](#).[IsAncestorOf](#)
[Node](#).[MethodName](#).[IsGreaterThan](#)

Node.MethodName.GetPath
Node.MethodName.GetPathTo
Node.MethodName.AddToGroup
Node.MethodName.RemoveFromGroup
Node.MethodName.IsInGroup
Node.MethodName.MoveChild
Node.MethodName.GetGroups
Node.MethodName.SetOwner
Node.MethodName.GetOwner
Node.MethodName.GetIndex
Node.MethodName.PrintTree
Node.MethodName.PrintTreePretty
Node.MethodName.GetTreeString
Node.MethodName.GetTreeStringPretty
Node.MethodName.SetSceneFilePath
Node.MethodName.GetSceneFilePath
Node.MethodName.PropagateNotification
Node.MethodName.PropagateCall
Node.MethodName.SetPhysicsProcess
Node.MethodName.GetPhysicsProcessDeltaTime
Node.MethodName.IsPhysicsProcessing
Node.MethodName.GetProcessDeltaTime
Node.MethodName.SetProcess
Node.MethodName.SetProcessPriority
Node.MethodName.GetProcessPriority
Node.MethodName.SetPhysicsProcessPriority
Node.MethodName.GetPhysicsProcessPriority
Node.MethodName.IsProcessing
Node.MethodName.SetProcessInput
Node.MethodName.IsProcessingInput
Node.MethodName.SetProcessShortcutInput
Node.MethodName.IsProcessingShortcutInput
Node.MethodName.SetProcessUnhandledInput
Node.MethodName.IsProcessingUnhandledInput
Node.MethodName.SetProcessUnhandledKeyInput
Node.MethodName.IsProcessingUnhandledKeyInput
Node.MethodName.SetProcessMode
Node.MethodName.GetProcessMode
Node.MethodName.CanProcess
Node.MethodName.SetProcessThreadGroup
Node.MethodName.GetProcessThreadGroup
Node.MethodName.SetProcessThreadMessages
Node.MethodName.GetProcessThreadMessages
Node.MethodName.SetProcessThreadGroupOrder
Node.MethodName.GetProcessThreadGroupOrder
Node.MethodName.SetDisplayFolded
Node.MethodName.IsDisplayedFolded
Node.MethodName.SetProcessInternal
Node.MethodName.IsProcessingInternal
Node.MethodName.SetPhysicsProcessInternal

Node.MethodName.IsPhysicsProcessingInternal
Node.MethodName.SetPhysicsInterpolationMode
Node.MethodName.GetPhysicsInterpolationMode
Node.MethodName.IsPhysicsInterpolated
Node.MethodName.IsPhysicsInterpolatedAndEnabled
Node.MethodName.ResetPhysicsInterpolation
Node.MethodName.SetAutoTranslateMode
Node.MethodName.GetAutoTranslateMode
Node.MethodName.SetTranslationDomainInherited
Node.MethodName.GetWindow
Node.MethodName.GetLastExclusiveWindow
Node.MethodName.GetTree
Node.MethodName.CreateTween
Node.MethodName.Duplicate
Node.MethodName.ReplaceBy
Node.MethodName.SetSceneInstanceLoadPlaceholder
Node.MethodName.GetSceneInstanceLoadPlaceholder
Node.MethodName.SetEditableInstance
Node.MethodName.IsEditableInstance
Node.MethodName.GetViewport
Node.MethodName.QueueFree
Node.MethodName.RequestReady
Node.MethodName.IsNodeReady
Node.MethodName.SetMultiplayerAuthority
Node.MethodName.GetMultiplayerAuthority
Node.MethodName.IsMultiplayerAuthority
Node.MethodName.GetMultiplayer
Node.MethodName.RpcConfig
Node.MethodName.GetRpcConfig
Node.MethodName.SetEditorDescription
Node.MethodName.GetEditorDescription
Node.MethodName._SetImportPath
Node.MethodName._GetImportPath
Node.MethodName.SetUniqueNameInOwner
Node.MethodName.IsUniqueNameInOwner
Node.MethodName.Atr
Node.MethodName.AtrN
Node.MethodName.Rpc
Node.MethodName.RpcId
Node.MethodName.UpdateConfigurationWarnings
Node.MethodName.CallDeferredThreadGroup
Node.MethodName.SetDeferredThreadGroup
Node.MethodName.NotifyDeferredThreadGroup
Node.MethodName.CallThreadSafe
Node.MethodName.SetThreadSafe
Node.MethodName.NotifyThreadSafe
GodotObject.MethodName._Get
GodotObject.MethodName._GetPropertyList
GodotObject.MethodName._IterGet
GodotObject.MethodName._IterInit

GodotObject.MethodName._IterNext
GodotObject.MethodName._Notification
GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceId
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr

GodotObject.MethodName.TrN
GodotObject.MethodName.GetTranslationDomain
GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfGodotEngine.Util`

Assembly: `CatSweeper.dll`

Syntax

```
public class MonoInstance<T>.MethodName : Node.MethodName
```

© Bus Fighter. All rights reserved.

Class MonoInstance<T>.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.PropertyName](#) → [Node.PropertyName](#) → [MonoInstance<T>.PropertyName](#) → [AsyncResourceLoader.PropertyName](#) → [GodotSceneManager.PropertyName](#)

Inherited Members

[Node.PropertyName._ImportPath](#)
[Node.PropertyName.Name](#)
[Node.PropertyName.UniqueNameInOwner](#)
[Node.PropertyName.SceneFilePath](#)
[Node.PropertyName.Owner](#)
[Node.PropertyName.Multiplayer](#)
[Node.PropertyName.ProcessMode](#)
[Node.PropertyName.ProcessPriority](#)
[Node.PropertyName.ProcessPhysicsPriority](#)
[Node.PropertyName.ProcessThreadGroup](#)
[Node.PropertyName.ProcessThreadGroupOrder](#)
[Node.PropertyName.ProcessThreadMessages](#)
[Node.PropertyName.PhysicsInterpolationMode](#)
[Node.PropertyName.AutoTranslateMode](#)
[Node.PropertyName.EditorDescription](#)
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class MonoInstance<T>.PropertyName : Node.PropertyName
```

Class MonoInstance<T>.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.SignalName](#) → [Node.SignalName](#) → [MonoInstance<T>.SignalName](#) → [AsyncResourceLoader.SignalName](#) → [GodotSceneManager.SignalName](#)

Inherited Members

[Node.SignalName.Ready](#)
[Node.SignalName.Renamed](#)
[Node.SignalName.TreeEntered](#)
[Node.SignalName.TreeExiting](#)
[Node.SignalName.TreeExited](#)
[Node.SignalName.ChildEnteredTree](#)
[Node.SignalName.ChildExitingTree](#)
[Node.SignalName.ChildOrderChanged](#)
[Node.SignalName.ReplacingBy](#)
[Node.SignalName.EditorDescriptionChanged](#)
[Node.SignalName.EditorStateChanged](#)
[GodotObject.SignalName.ScriptChanged](#)
[GodotObject.SignalName.PropertyListChanged](#)
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class MonoInstance<T>.SignalName : Node.SignalName
```

Class NodeUtil

Inheritance

`object` → `NodeUtil`

Inherited Members

`object.Equals(object)`
`object.Equals(object, object)`
`object.GetHashCode()`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public static class NodeUtil
```

Methods

DontDestroyOnLoad(Node)

Declaration

```
public static void DontDestroyOnLoad(this Node node)
```

Parameters

TYPE	NAME
------	------

Node	node
------	------

GetSceneTree()

Declaration

```
public static SceneTree GetSceneTree()
```

Returns

TYPE

SceneTree

© Bus Fighter. All rights reserved.

Class SettingPath

Inheritance

object → [Attribute](#) → SettingPath

Inherited Members

[Attribute.Equals\(object\)](#)
[Attribute.GetCustomAttribute\(Assembly, Type\)](#)
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#)
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#)
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#)
[Attribute.GetCustomAttribute\(Module, Type\)](#)
[Attribute.GetCustomAttribute\(Module, Type, bool\)](#)
[Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#)
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#)
[Attribute.GetCustomAttributes\(Assembly\)](#)
[Attribute.GetCustomAttributes\(Assembly, bool\)](#)
[Attribute.GetCustomAttributes\(Assembly, Type\)](#)
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#)
[Attribute.GetCustomAttributes\(MemberInfo\)](#)
[Attribute.GetCustomAttributes\(MemberInfo, bool\)](#)
[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#)
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#)
[Attribute.GetCustomAttributes\(Module\)](#)
[Attribute.GetCustomAttributes\(Module, bool\)](#)
[Attribute.GetCustomAttributes\(Module, Type\)](#)
[Attribute.GetCustomAttributes\(Module, Type, bool\)](#)
[Attribute.GetCustomAttributes\(ParameterInfo\)](#)
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#)
[Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#)
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#)
[Attribute.GetHashCode\(\)](#)
[Attribute.IsDefaultAttribute\(\)](#)
[Attribute.IsDefined\(Assembly, Type\)](#)
[Attribute.IsDefined\(Assembly, Type, bool\)](#)
[Attribute.IsDefined\(MemberInfo, Type\)](#)
[Attribute.IsDefined\(MemberInfo, Type, bool\)](#)
[Attribute.IsDefined\(Module, Type\)](#)
[Attribute.IsDefined\(Module, Type, bool\)](#)
[Attribute.IsDefined\(ParameterInfo, Type\)](#)
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#)
[Attribute.Match\(object\)](#)

`Attribute.TypeId`
`object.Equals(object, object)`
`object.GetType()`
`object.MemberwiseClone()`
`object.ReferenceEquals(object, object)`
`object.ToString()`

Namespace: `cfGodotEngine.Util`

Assembly: `CatSweeper.dll`

Syntax

```
public class SettingPath : Attribute
```

Constructors

SettingPath(string)

Declaration

```
public SettingPath(string path)
```

Parameters

TYPE	NAME
<code>string</code>	<code>path</code>

Fields

path

Declaration

```
public readonly string path
```

Field Value

TYPE
<code>string</code>

Class Setting<T>

Inheritance

[object](#) → [GodotObject](#) → [RefCounted](#) → [Resource](#) → [Setting<T>](#) → [DriveMirrorSetting](#)

Implements

[IDisposable](#)

Inherited Members

[Resource._GetRid\(\)](#)

[Resource._ResetState\(\)](#)

[Resource._SetPathCache\(string\)](#)

[Resource._SetupLocalToScene\(\)](#)

[Resource.TakeOverPath\(string\)](#)

[Resource.SetPathCache\(string\)](#)

[Resource.GetRid\(\)](#)

[Resource.GetLocalScene\(\)](#)

[Resource.SetupLocalToScene\(\)](#)

[Resource.ResetState\(\)](#)

[Resource.SetIdForPath\(string, string\)](#)

[Resource.GetIdForPath\(string\)](#)

[Resource.IsBuiltIn\(\)](#)

[Resource.GenerateSceneUniqueId\(\)](#)

[Resource.EmitChanged\(\)](#)

[Resource.Duplicate\(bool\)](#)

[Resource.EmitSignalChanged\(\)](#)

[Resource.EmitSignalSetupLocalToSceneRequested\(\)](#)

[Resource.InvokeGodotClassMethod\(in godot_string_name, NativeVariantPtrArgs, out godot_variant\)](#)

[Resource.HasGodotClassMethod\(in godot_string_name\)](#)

[Resource.HasGodotClassSignal\(in godot_string_name\)](#)

[Resource.ResourceLocalToScene](#)

[Resource.ResourcePath](#)

[Resource.ResourceName](#)

[Resource.ResourceSceneUniqueId](#)

[Resource.Changed](#)

[Resource.SetupLocalToSceneRequested](#)

[RefCounted.InitRef\(\)](#)

[RefCounted.Reference\(\)](#)

[RefCounted.Unreference\(\)](#)

[RefCounted.GetReferenceCount\(\)](#)

[GodotObject.NotificationPostInitialize](#)

GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)
GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)

GodotObject.Callv(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObject.GetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
[ScriptPath("res://Modules/cfGodotEngine/util/Setting.cs")]
public abstract class Setting<T> : Resource, IDisposable where T : Setting<T>
```

Type Parameters

NAME

T

Methods

GetSetting()

Declaration

```
public static T GetSetting()
```

Returns

TYPE

T

Implements

[IDisposable](#)

© Bus Fighter. All rights reserved.

Class Setting<T>.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject](#).[MethodName](#) → [RefCounted](#).[MethodName](#) → [Resource](#).[MethodName](#) → [Setting<T>](#).[MethodName](#) → [DriveMirrorSetting](#).[MethodName](#)

Inherited Members

[Resource](#).[MethodName](#).[_GetRid](#)
[Resource](#).[MethodName](#).[_ResetState](#)
[Resource](#).[MethodName](#).[_SetPathCache](#)
[Resource](#).[MethodName](#).[_SetupLocalToScene](#)
[Resource](#).[MethodName](#).[SetPath](#)
[Resource](#).[MethodName](#).[TakeOverPath](#)
[Resource](#).[MethodName](#).[GetPath](#)
[Resource](#).[MethodName](#).[SetPathCache](#)
[Resource](#).[MethodName](#).[SetName](#)
[Resource](#).[MethodName](#).[GetName](#)
[Resource](#).[MethodName](#).[GetRid](#)
[Resource](#).[MethodName](#).[SetLocalToScene](#)
[Resource](#).[MethodName](#).[IsLocalToScene](#)
[Resource](#).[MethodName](#).[GetLocalScene](#)
[Resource](#).[MethodName](#).[SetupLocalToScene](#)
[Resource](#).[MethodName](#).[ResetState](#)
[Resource](#).[MethodName](#).[SetIdForPath](#)
[Resource](#).[MethodName](#).[GetIdForPath](#)
[Resource](#).[MethodName](#).[IsBuiltIn](#)
[Resource](#).[MethodName](#).[GenerateSceneUniqueId](#)
[Resource](#).[MethodName](#).[SetSceneUniqueId](#)
[Resource](#).[MethodName](#).[GetSceneUniqueId](#)
[Resource](#).[MethodName](#).[EmitChanged](#)
[Resource](#).[MethodName](#).[Duplicate](#)
[RefCounted](#).[MethodName](#).[InitRef](#)
[RefCounted](#).[MethodName](#).[Reference](#)
[RefCounted](#).[MethodName](#).[Unreference](#)
[RefCounted](#).[MethodName](#).[GetReferenceCount](#)
[GodotObject](#).[MethodName](#).[_Get](#)
[GodotObject](#).[MethodName](#).[_GetPropertyList](#)
[GodotObject](#).[MethodName](#).[_IterGet](#)
[GodotObject](#).[MethodName](#).[_IterInit](#)
[GodotObject](#).[MethodName](#).[_IterNext](#)

GodotObject.MethodName._Notification
GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceId
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyListChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr
GodotObject.MethodName.TrN

GodotObject.MethodName.GetTranslationDomain
GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class Setting<T>.MethodName : Resource.MethodName
```

© Bus Fighter. All rights reserved.

Class Setting<T>.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.PropertyName](#) → [RefCounted\(PropertyName\)](#) → [Resource\(PropertyName\)](#) → [Setting<T>.PropertyName](#) → [DriveMirrorSetting\(PropertyName\)](#)

Inherited Members

[Resource\(PropertyName.ResourceLocalToScene\)](#)

[Resource\(PropertyName.ResourcePath\)](#)

[Resource\(PropertyName.ResourceName\)](#)

[Resource\(PropertyName.ResourceSceneUniqueld\)](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class Setting<T>.PropertyName : Resource(PropertyName)
```

© Bus Fighter. All rights reserved.

Class Setting<T>.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

[object](#) → GodotObject.SignalName → RefCounted.SignalName → Resource.SignalName → Setting<T>.SignalName → [DriveMirrorSetting.SignalName](#)

Inherited Members

Resource.SignalName.Changed

Resource.SignalName.SetupLocalToSceneRequested

GodotObject.SignalName.ScriptChanged

GodotObject.SignalName.PropertyListChanged

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.GetType\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class Setting<T>.SignalName : Resource.SignalName
```

© Bus Fighter. All rights reserved.

Class StateExecutionException<TStatId>

Inheritance

[object](#) → [Exception](#) → [StateExecutionException<TStatId>](#)

Implements

[ISerializable](#)

Inherited Members

[Exception.GetBaseException\(\)](#)

[Exception.GetType\(\)](#)

[Exception.ToString\(\)](#)

[Exception.Data](#)

[Exception.HelpLink](#)

[Exception.HResult](#)

[Exception.InnerException](#)

[Exception.Message](#)

[Exception.Source](#)

[Exception.StackTrace](#)

[Exception.TargetSite](#)

[Exception.SerializeObjectState](#)

[object.Equals\(object\)](#)

[object.Equals\(object, object\)](#)

[object.GetHashCode\(\)](#)

[object.MemberwiseClone\(\)](#)

[object.ReferenceEquals\(object, object\)](#)

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class StateExecutionException<TStateId> : Exception, ISerializable
```

Type Parameters

NAME

TStateId

Constructors

StateExecutionException(TStatId, Exception)

Declaration

```
public StateExecutionException(TStateId stateId, Exception innerException)
```

Parameters

TYPE	NAME
TStatId	stateId
Exception	innerException

Implements

ISerializable

© Bus Fighter. All rights reserved.

Class StateMachineNode<TStatId, TState, TStateMachine>

Inheritance

[object](#) → [GodotObject](#) → [Node](#) → [StateMachineNode<TStatId, TState, TStateMachine>](#)

Implements

[IDisposable](#)

[IStateMachine<TStatId>](#)

Inherited Members

[Node.NotificationEnterTree](#)
[Node.NotificationExitTree](#)
[Node.NotificationMovedInParent](#)
[Node.NotificationReady](#)
[Node.NotificationPaused](#)
[Node.NotificationUnpaused](#)
[Node.NotificationPhysicsProcess](#)
[Node.NotificationProcess](#)
[Node.NotificationParented](#)
[Node.NotificationUnparented](#)
[Node.NotificationSceneInstantiated](#)
[Node.NotificationDragBegin](#)
[Node.NotificationDragEnd](#)
[Node.NotificationPathRenamed](#)
[Node.NotificationChildOrderChanged](#)
[Node.NotificationInternalProcess](#)
[Node.NotificationInternalPhysicsProcess](#)
[Node.NotificationPostEnterTree](#)
[Node.NotificationDisabled](#)
[Node.NotificationEnabled](#)
[Node.NotificationResetPhysicsInterpolation](#)
[Node.NotificationEditorPreSave](#)
[Node.NotificationEditorPostSave](#)
[Node.NotificationWMMouseEnter](#)
[Node.NotificationWMMouseExit](#)
[Node.NotificationWMWindowFocusIn](#)
[Node.NotificationWMWindowFocusOut](#)
[Node.NotificationWMCloseRequest](#)
[Node.NotificationWMGoBackRequest](#)
[Node.NotificationWMSizeChanged](#)

Node.NotificationWMDpiChange
Node.NotificationVpMouseEnter
Node.NotificationVpMouseExit
Node.NotificationWMPositionChanged
Node.NotificationOsMemoryWarning
Node.NotificationTranslationChanged
Node.NotificationWMAbout
Node.NotificationCrash
Node.NotificationOslmeUpdate
Node.NotificationApplicationResumed
Node.NotificationApplicationPaused
Node.NotificationApplicationFocusIn
Node.NotificationApplicationFocusOut
Node.NotificationTextServerChanged
Node.GetNode<T>(NodePath)
Node.GetNodeOrNull<T>(NodePath)
Node.GetChild<T>(int, bool)
Node.GetChildOrNull<T>(int, bool)
Node.GetOwner<T>()
Node.GetOwnerOrNull<T>()
Node.GetParent<T>()
Node.GetParentOrNull<T>()
Node._EnterTree()
Node._ExitTree()
Node._GetConfigurationWarnings()
Node._Input(InputEvent)
Node._PhysicsProcess(double)
Node._ShortcutInput(InputEvent)
Node._UnhandledInput(InputEvent)
Node._UnhandledKeyInput(InputEvent)
Node.PrintOrphanNodes()
Node.AddSibling(Node, bool)
Node.AddChild(Node, bool, Node.InternalMode)
Node.RemoveChild(Node)
Node.Reparent(Node, bool)
Node.GetChildCount(bool)
Node.GetChildren(bool)
Node.GetChild(int, bool)
Node.HasNode(NodePath)
Node.GetNode(NodePath)
Node.GetNodeOrNull(NodePath)
Node.GetParent()
Node.FindChild(string, bool, bool)
Node.FindChildren(string, string, bool, bool)
Node.FindParent(string)
Node.HasNodeAndResource(NodePath)
Node.GetNodeAndResource(NodePath)
Node.IsInsideTree()
Node.IsPartOfEditedScene()
Node.IsAncestorOf(Node)

Node.IsGreater Than(Node)
Node.GetPath()
Node.GetPath To(Node, bool)
Node.AddToGroup(StringName, bool)
Node.RemoveFromGroup(StringName)
Node.IsInGroup(StringName)
Node.MoveChild(Node, int)
Node.GetGroups()
Node.GetIndex(bool)
Node.PrintTree()
Node.PrintTreePretty()
Node.GetTreeString()
Node.GetTreeStringPretty()
Node.PropagateNotification(int)
Node.PropagateCall(StringName, Array, bool)
Node.SetPhysicsProcess(bool)
Node.GetPhysicsProcessDelta Time()
Node.IsPhysicsProcessing()
Node.GetProcessDelta Time()
Node.SetProcess(bool)
Node.IsProcessing()
Node.SetProcessInput(bool)
Node.IsProcessingInput()
Node.SetProcessShortcutInput(bool)
Node.IsProcessingShortcutInput()
Node.SetProcessUnhandledInput(bool)
Node.IsProcessingUnhandledInput()
Node.SetProcessUnhandledKeyInput(bool)
Node.IsProcessingUnhandledKeyInput()
Node.CanProcess()
Node.SetDisplayFolded(bool)
Node.IsDisplayedFolded()
Node.SetProcessInternal(bool)
Node.IsProcessingInternal()
Node.SetPhysicsProcessInternal(bool)
Node.IsPhysicsProcessingInternal()
Node.IsPhysicsInterpolated()
Node.IsPhysicsInterpolatedAndEnabled()
Node.ResetPhysicsInterpolation()
Node.SetTranslationDomainInherited()
Node.GetWindow()
Node.GetLastExclusiveWindow()
Node.GetTree()
Node.CreateTween()
Node.Duplicate(int)
Node.ReplaceBy(Node, bool)
Node.setSceneInstanceLoadPlaceholder(bool)
Node.GetSceneInstanceLoadPlaceholder()
Node.setEditableInstance(Node, bool)
Node.setEditableInstance(Node)

```
Node.GetViewport()
Node.QueueFree()
Node.RequestReady()
Node.IsNodeReady()
Node.SetMultiplayerAuthority(int, bool)
Node.GetMultiplayerAuthority()
Node.IsMultiplayerAuthority()
Node.RpcConfig(StringName, Variant)
Node.GetRpcConfig()
Node.Atr(string, StringName)
Node.AtrN(string, StringName, int, StringName)
Node.Rpc(StringName, params Variant[])
Node.Rpc(StringName, ReadOnlySpan<Variant>)
Node.RpcId(long, StringName, params Variant[])
Node.RpcId(long, StringName, ReadOnlySpan<Variant>)
Node.UpdateConfigurationWarnings()
Node.CallDeferredThreadGroup(StringName, params Variant[])
Node.CallDeferredThreadGroup(StringName, ReadOnlySpan<Variant>)
Node.SetDeferredThreadGroup(StringName, Variant)
Node.NotifyDeferredThreadGroup(int)
Node.CallThreadSafe(StringName, params Variant[])
Node.CallThreadSafe(StringName, ReadOnlySpan<Variant>)
Node.SetThreadSafe(StringName, Variant)
Node.NotifyThreadSafe(int)
Node.EmitSignalReady()
Node.EmitSignalRenamed()
Node.EmitSignalTreeEntered()
Node.EmitSignalTreeExiting()
Node.EmitSignalTreeExited()
Node.EmitSignalChildEnteredTree(Node)
Node.EmitSignalChildExitingTree(Node)
Node.EmitSignalChildOrderChanged()
Node.EmitSignalReplacingBy(Node)
Node.EmitSignalEditorDescriptionChanged(Node)
Node.EmitSignalEditorStateChanged()
Node.InvokeGodotClassMethod(in godot_string_name, NativeVariantPtrArgs, out godot_variant)
Node.HasGodotClassMethod(in godot_string_name)
Node.HasGodotClassSignal(in godot_string_name)
Node.Name
Node.UniqueNameInOwner
Node.SceneFilePath
Node.Owner
Node.Multiplayer
Node.ProcessMode
Node.ProcessPriority
Node.ProcessPhysicsPriority
Node.ProcessThreadGroup
Node.ProcessThreadGroupOrder
Node.ProcessThreadMessages
Node.PhysicsInterpolationMode
```

Node.AutoTranslateMode
Node.EditorDescription
Node.Ready
Node.Renamed
Node.TreeEntered
Node.TreeExiting
Node.TreeExited
Node.ChildEnteredTree
Node.ChildExitingTree
Node.ChildOrderChanged
Node.ReplacingBy
Node.EditorDescriptionChanged
Node.EditorStateChanged
GodotObject.NotificationPostinitialize
GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)

GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)
GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)
GodotObject.Cally(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObjectGetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Namespace: **cfGodotEngine.Util**

Assembly: CatSweeper.dll

Syntax

```
[ScriptPath("res://Modules/cfGodotEngine/util/StateMachineNode/StateMachineNode.cs")]
public abstract class StateMachineNode<TStateId, TState, TStateMachine> : Node, IDisposable, I
```

Type Parameters

NAME

TStateId

TState

TStateMachine

Properties

allState

Declaration

```
protected IEnumerable<TState> allState { get; }
```

Property Value

TYPE

IEnumerable<TState>

currentState

Declaration

```
protected TState currentState { get; }
```

Property Value

TYPE

TState

currentStatId

Declaration

```
public TStateId currentStateId { get; }
```

Property Value

TYPE

TStatId

lastState

Declaration

```
protected TState lastState { get; }
```

Property Value

TYPE

TState

lastStatId

Declaration

```
public TStateId lastStateId { get; }
```

Property Value

TYPE

TStatId

Methods

CanGoToState(TStatId, StateParam)

Declaration

```
public bool CanGoToState(TStateId id, StateParam param)
```

Parameters

TYPE	NAME
TStateld	id
StateParam	param

Returns

TYPE
bool

ForceGoToState(TStateld, StateParam)

Declaration

```
public void ForceGoToState(TStateId nextStateId, StateParam param = null)
```

Parameters

TYPE	NAME
TStateld	nextStateId
StateParam	param

GetStateUnsafe(TStateld)

Declaration

```
public TState GetStateUnsafe(TStateId id)
```

Parameters

TYPE	NAME
TStateld	id

Returns

TYPE

TState

GetStateUnsafe<T>(TStatId)

Declaration

```
public T GetStateUnsafe<T>(TStateId id) where T : TState
```

Parameters

TYPE **NAME**

TStatId id

Returns

TYPE

T

Type Parameters

NAME

T

RegisterState(TState)

Declaration

```
public void RegisterState(TState state)
```

Parameters

TYPE **NAME**

TState state

SubscribeAfterStateChange(Action<StateChangeRecord<TStatId>>)

Declaration

```
public Subscription SubscribeAfterStateChange(Action<StateChangeRecord<TStateId>> listener)
```

Parameters

TYPE	NAME
Action<StateChangeRecord<TStatId>>	listener

Returns

TYPE
Subscription

SubscribeBeforeStateChange(Action<StateChangeRecord<TStatId>>)

Declaration

```
public Subscription SubscribeBeforeStateChange(Action<StateChangeRecord<TStateId>> listener)
```

Parameters

TYPE	NAME
Action<StateChangeRecord<TStatId>>	listener

Returns

TYPE
Subscription

TryGetState(TStatId, out TState)

Declaration

```
public bool TryGetState(TStateId id, out TState monoState)
```

Parameters

TYPE	NAME
TStatId	id
TState	monoState

Returns

TYPE

bool

TryGetState<T>(TStatId, out T)

Declaration

```
public bool TryGetState<T>(TStateId id, out T state) where T : TState
```

Parameters

TYPE **NAME**

TStatId id

T state

Returns

TYPE

bool

Type Parameters

NAME

T

TryGoToState(TStatId, StateParam)

Declaration

```
public bool TryGoToState(TStateId nextStateId, StateParam param = null)
```

Parameters

TYPE **NAME**

TStatId nextStateId

StateParam param

Returns

TYPE

bool

_Process(double)

Called during the processing step of the main loop. Processing happens at every frame and as fast as possible, so the `delta` time since the previous frame is not constant. `delta` is in seconds.

It is only called if processing is enabled, which is done automatically if this method is overridden, and can be toggled with `SetProcess(bool)`.

Processing happens in order of `Godot.Node.ProcessPriority`, lower priority values are called first. Nodes with the same priority are processed in tree order, or top to bottom as seen in the editor (also known as pre-order traversal).

Corresponds to the `Godot.Node.NotificationProcess` notification in `_Notification(int)`.

Note: This method is only called if the node is present in the scene tree (i.e. if it's not an orphan).

Note: `delta` will be larger than expected if running at a framerate lower than `Godot.Engine.PhysicsTicksPerSecond` / `Godot.Engine.MaxPhysicsStepsPerFrame` FPS. This is done to avoid "spiral of death" scenarios where performance would plummet due to an ever-increasing number of physics steps per frame. This behavior affects both `_Process(double)` and `_PhysicsProcess(double)`. As a result, avoid using `delta` for time measurements in real-world seconds. Use the `Godot.Time` singleton's methods for this purpose instead, such as `Godot.Time.GetTicksUseC()`.

Declaration

```
public override void _Process(double delta)
```

Parameters

TYPE	NAME
double	delta

Overrides

[Node._Process\(double\)](#)

_Ready()

Called when the node is "ready", i.e. when both the node and its children have entered the scene tree. If the node has children, their `Godot.Node._Ready()` callbacks get triggered first, and the parent node will receive the ready notification afterwards.

Corresponds to the `Godot.Node.NotificationReady` notification in `_Notification(int)`. See also the `@onready` annotation for variables.

Usually used for initialization. For even earlier initialization, `Godot.GodotObject.GodotObject()` may be used. See also `Godot.Node._EnterTree()`.

Note: This method may be called only once for each node. After removing a node from the scene tree and adding it again, `Godot.Node._Ready()` will **not** be called a second time. This can be bypassed by requesting another call with `Godot.Node.RequestReady()`, which may be called anywhere before adding the node again.

Declaration

```
public override void _Ready()
```

Overrides

`Godot.Node._Ready()`

__Process(double)

Declaration

```
protected virtual void __Process(double delta)
```

Parameters

TYPE	NAME
double	delta

__Ready()

Declaration

```
protected virtual void __Ready()
```

Implements

`IDisposable`
`IStateMachine<TStateId>`

Extension Methods

© Bus Fighter. All rights reserved.

Class StateMachineNode<TStatId, TState, TStateMachine>.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject](#).[MethodName](#) → [Node](#).[MethodName](#) → [StateMachineNode<TStatId, TState, TStateMachine>.MethodName](#)

Inherited Members

[Node](#).[MethodName](#).[_EnterTree](#)
[Node](#).[MethodName](#).[_ExitTree](#)
[Node](#).[MethodName](#).[_GetConfigurationWarnings](#)
[Node](#).[MethodName](#).[_Input](#)
[Node](#).[MethodName](#).[_PhysicsProcess](#)
[Node](#).[MethodName](#).[_ShortcutInput](#)
[Node](#).[MethodName](#).[_UnhandledInput](#)
[Node](#).[MethodName](#).[_UnhandledKeyInput](#)
[Node](#).[MethodName](#).[PrintOrphanNodes](#)
[Node](#).[MethodName](#).[AddSibling](#)
[Node](#).[MethodName](#).[SetName](#)
[Node](#).[MethodName](#).[GetName](#)
[Node](#).[MethodName](#).[AddChild](#)
[Node](#).[MethodName](#).[RemoveChild](#)
[Node](#).[MethodName](#).[Reparent](#)
[Node](#).[MethodName](#).[GetChildCount](#)
[Node](#).[MethodName](#).[GetChildren](#)
[Node](#).[MethodName](#).[GetChild](#)
[Node](#).[MethodName](#).[HasNode](#)
[Node](#).[MethodName](#).[GetNode](#)
[Node](#).[MethodName](#).[GetNodeOrNull](#)
[Node](#).[MethodName](#).[GetParent](#)
[Node](#).[MethodName](#).[FindChild](#)
[Node](#).[MethodName](#).[FindChildren](#)
[Node](#).[MethodName](#).[FindParent](#)
[Node](#).[MethodName](#).[HasNodeAndResource](#)
[Node](#).[MethodName](#).[GetNodeAndResource](#)
[Node](#).[MethodName](#).[IsInsideTree](#)
[Node](#).[MethodName](#).[IsPartOfEditedScene](#)
[Node](#).[MethodName](#).[IsAncestorOf](#)
[Node](#).[MethodName](#).[IsGreaterThan](#)

Node.MethodName.GetPath
Node.MethodName.GetPathTo
Node.MethodName.AddToGroup
Node.MethodName.RemoveFromGroup
Node.MethodName.IsInGroup
Node.MethodName.MoveChild
Node.MethodName.GetGroups
Node.MethodName.SetOwner
Node.MethodName.GetOwner
Node.MethodName.GetIndex
Node.MethodName.PrintTree
Node.MethodName.PrintTreePretty
Node.MethodName.GetTreeString
Node.MethodName.GetTreeStringPretty
Node.MethodName.SetSceneFilePath
Node.MethodName.GetSceneFilePath
Node.MethodName.PropagateNotification
Node.MethodName.PropagateCall
Node.MethodName.SetPhysicsProcess
Node.MethodName.GetPhysicsProcessDeltaTime
Node.MethodName.IsPhysicsProcessing
Node.MethodName.GetProcessDeltaTime
Node.MethodName.SetProcess
Node.MethodName.SetProcessPriority
Node.MethodName.GetProcessPriority
Node.MethodName.SetPhysicsProcessPriority
Node.MethodName.GetPhysicsProcessPriority
Node.MethodName.IsProcessing
Node.MethodName.SetProcessInput
Node.MethodName.IsProcessingInput
Node.MethodName.SetProcessShortcutInput
Node.MethodName.IsProcessingShortcutInput
Node.MethodName.SetProcessUnhandledInput
Node.MethodName.IsProcessingUnhandledInput
Node.MethodName.SetProcessUnhandledKeyInput
Node.MethodName.IsProcessingUnhandledKeyInput
Node.MethodName.SetProcessMode
Node.MethodName.GetProcessMode
Node.MethodName.CanProcess
Node.MethodName.SetProcessThreadGroup
Node.MethodName.GetProcessThreadGroup
Node.MethodName.SetProcessThreadMessages
Node.MethodName.GetProcessThreadMessages
Node.MethodName.SetProcessThreadGroupOrder
Node.MethodName.GetProcessThreadGroupOrder
Node.MethodName.SetDisplayFolded
Node.MethodName.IsDisplayedFolded
Node.MethodName.SetProcessInternal
Node.MethodName.IsProcessingInternal
Node.MethodName.SetPhysicsProcessInternal

Node.MethodName.IsPhysicsProcessingInternal
Node.MethodName.SetPhysicsInterpolationMode
Node.MethodName.GetPhysicsInterpolationMode
Node.MethodName.IsPhysicsInterpolated
Node.MethodName.IsPhysicsInterpolatedAndEnabled
Node.MethodName.ResetPhysicsInterpolation
Node.MethodName.SetAutoTranslateMode
Node.MethodName.GetAutoTranslateMode
Node.MethodName.SetTranslationDomainInherited
Node.MethodName.GetWindow
Node.MethodName.GetLastExclusiveWindow
Node.MethodName.GetTree
Node.MethodName.CreateTween
Node.MethodName.Duplicate
Node.MethodName.ReplaceBy
Node.MethodName.SetSceneInstanceLoadPlaceholder
Node.MethodName.GetSceneInstanceLoadPlaceholder
Node.MethodName.SetEditableInstance
Node.MethodName.IsEditableInstance
Node.MethodName.GetViewport
Node.MethodName.QueueFree
Node.MethodName.RequestReady
Node.MethodName.IsNodeReady
Node.MethodName.SetMultiplayerAuthority
Node.MethodName.GetMultiplayerAuthority
Node.MethodName.IsMultiplayerAuthority
Node.MethodName.GetMultiplayer
Node.MethodName.RpcConfig
Node.MethodName.GetRpcConfig
Node.MethodName.SetEditorDescription
Node.MethodName.GetEditorDescription
Node.MethodName._SetImportPath
Node.MethodName._GetImportPath
Node.MethodName.SetUniqueNameInOwner
Node.MethodName.IsUniqueNameInOwner
Node.MethodName.Atr
Node.MethodName.AtrN
Node.MethodName.Rpc
Node.MethodName.RpcId
Node.MethodName.UpdateConfigurationWarnings
Node.MethodName.CallDeferredThreadGroup
Node.MethodName.SetDeferredThreadGroup
Node.MethodName.NotifyDeferredThreadGroup
Node.MethodName.CallThreadSafe
Node.MethodName.SetThreadSafe
Node.MethodName.NotifyThreadSafe
GodotObject.MethodName._Get
GodotObject.MethodName._GetPropertyList
GodotObject.MethodName._IterGet
GodotObject.MethodName._IterInit

GodotObject.MethodName._IterNext
GodotObject.MethodName._Notification
GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceId
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr

GodotObject.MethodName.TrN
GodotObject.MethodName.GetTranslationDomain
GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: **cfGodotEngine.Util**

Assembly: CatSweeper.dll

Syntax

```
public class StateMachineNode<TStateId, TState, TStateMachine>.MethodName : Node.MethodName
```

Fields

_Process

Cached name for the '_Process' method.

Declaration

```
public static readonly StringName _Process
```

Field Value

TYPE

StringName

_Ready

Cached name for the '_Ready' method.

Declaration

```
public static readonly StringName _Ready
```

Field Value

TYPE

StringName

__Process

Cached name for the '__Process' method.

Declaration

```
public static readonly StringName __Process
```

Field Value

TYPE

StringName

__Ready

Cached name for the '__Ready' method.

Declaration

```
public static readonly StringName __Ready
```

Field Value

TYPE

StringName

Class StateMachineNode<TStatId, TState, TStateMachine>.PropertyName

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.PropertyName](#) → [Node.PropertyName](#) → [StateMachineNode<TStatId, TState, TStateMachine>.PropertyName](#)

Inherited Members

[Node.PropertyName._ImportPath](#)
[Node.PropertyName.Name](#)
[Node.PropertyName.UniqueNameInOwner](#)
[Node.PropertyName.SceneFilePath](#)
[Node.PropertyName.Owner](#)
[Node.PropertyName.Multiplayer](#)
[Node.PropertyName.ProcessMode](#)
[Node.PropertyName.ProcessPriority](#)
[Node.PropertyName.ProcessPhysicsPriority](#)
[Node.PropertyName.ProcessThreadGroup](#)
[Node.PropertyName.ProcessThreadGroupOrder](#)
[Node.PropertyName.ProcessThreadMessages](#)
[Node.PropertyName.PhysicsInterpolationMode](#)
[Node.PropertyName.AutoTranslateMode](#)
[Node.PropertyName.EditorDescription](#)
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class StateMachineNode<TStateId, TState, TStateMachine>.PropertyName : Node.PropertyName
```


Class StateMachineNode<TStatId, TState, TStateMachine>.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

[object](#) → GodotObject.SignalName → Node.SignalName → StateMachineNode<TStatId, TState, TStateMachine>.SignalName

Inherited Members

Node.SignalName.Ready
Node.SignalName.Renamed
Node.SignalName.TreeEntered
Node.SignalName.TreeExiting
Node.SignalName.TreeExited
Node.SignalName.ChildEnteredTree
Node.SignalName.ChildExitingTree
Node.SignalName.ChildOrderChanged
Node.SignalName.ReplacingBy
Node.SignalName.EditorDescriptionChanged
Node.SignalName.EditorStateChanged
GodotObject.SignalName.ScriptChanged
GodotObject.SignalName.PropertyListChanged
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class StateMachineNode<TStateId, TState, TStateMachine>.SignalName : Node.SignalName
```

Class StateNode<TStatId, TState, TStateMachine>

Inheritance

[object](#) → [GodotObject](#) → [Node](#) → [StateNode<TStatId, TState, TStateMachine>](#)

Implements

[IDisposable](#)

Inherited Members

Node.NotificationEnterTree
Node.NotificationExitTree
Node.NotificationMovedInParent
Node.NotificationReady
Node.NotificationPaused
Node.NotificationUnpaused
Node.NotificationPhysicsProcess
Node.NotificationProcess
Node.NotificationParented
Node.NotificationUnparented
Node.NotificationSceneInstantiated
Node.NotificationDragBegin
Node.NotificationDragEnd
Node.NotificationPathRenamed
Node.NotificationChildOrderChanged
Node.NotificationInternalProcess
Node.NotificationInternalPhysicsProcess
Node.NotificationPostEnterTree
Node.NotificationDisabled
Node.NotificationEnabled
Node.NotificationResetPhysicsInterpolation
Node.NotificationEditorPreSave
Node.NotificationEditorPostSave
Node.NotificationWMMouseEnter
Node.NotificationWMMouseExit
Node.NotificationWMWindowFocusIn
Node.NotificationWMWindowFocusOut
Node.NotificationWMCloseRequest
Node.NotificationWMGoBackRequest
Node.NotificationWMSizeChanged
Node.NotificationWMDpiChange
Node.NotificationVpMouseEnter

Node.NotificationVpMouseExit
Node.NotificationWMPositionChanged
Node.NotificationOsMemoryWarning
Node.NotificationTranslationChanged
Node.NotificationWMAbout
Node.NotificationCrash
Node.NotificationOslmeUpdate
Node.NotificationApplicationResumed
Node.NotificationApplicationPaused
Node.NotificationApplicationFocusIn
Node.NotificationApplicationFocusOut
Node.NotificationTextServerChanged
Node.GetNode<T>(NodePath)
Node.GetNodeOrNull<T>(NodePath)
Node.GetChild<T>(int, bool)
Node.GetChildOrNull<T>(int, bool)
Node.GetOwner<T>()
Node.GetOwnerOrNull<T>()
Node.GetParent<T>()
Node.GetParentOrNull<T>()
Node._EnterTree()
Node._ExitTree()
Node._GetConfigurationWarnings()
Node._Input(InputEvent)
Node._PhysicsProcess(double)
Node._ShortcutInput(InputEvent)
Node._UnhandledInput(InputEvent)
Node._UnhandledKeyInput(InputEvent)
Node.PrintOrphanNodes()
Node.AddSibling(Node, bool)
Node.AddChild(Node, bool, Node.InternalMode)
Node.RemoveChild(Node)
Node.Reparent(Node, bool)
Node.GetChildCount(bool)
Node.GetChildren(bool)
Node.GetChild(int, bool)
Node.HasNode(NodePath)
Node.GetNode(NodePath)
Node.GetNodeOrNull(NodePath)
Node.GetParent()
Node.FindChild(string, bool, bool)
Node.FindChildren(string, string, bool, bool)
Node.FindParent(string)
Node.HasNodeAndResource(NodePath)
Node.GetNodeAndResource(NodePath)
Node.IsInsideTree()
Node.IsPartOfEditedScene()
Node.IsAncestorOf(Node)
Node.IsGreater Than(Node)
Node.GetPath()

Node.GetPathTo(Node, bool)
Node.AddToGroup(StringName, bool)
Node.RemoveFromGroup(StringName)
Node.IsInGroup(StringName)
Node.MoveChild(Node, int)
Node.GetGroups()
Node.GetIndex(bool)
Node.PrintTree()
Node.PrintTreePretty()
Node.GetTreeString()
Node.GetTreeStringPretty()
Node.PropagateNotification(int)
Node.PropagateCall(StringName, Array, bool)
Node.SetPhysicsProcess(bool)
Node.GetPhysicsProcessDeltaTime()
Node.IsPhysicsProcessing()
Node.GetProcessDeltaTime()
Node.SetProcess(bool)
Node.IsProcessing()
Node.SetProcessInput(bool)
Node.IsProcessingInput()
Node.SetProcessShortcutInput(bool)
Node.IsProcessingShortcutInput()
Node.SetProcessUnhandledInput(bool)
Node.IsProcessingUnhandledInput()
Node.SetProcessUnhandledKeyInput(bool)
Node.IsProcessingUnhandledKeyInput()
Node.CanProcess()
Node.SetDisplayFolded(bool)
Node.IsDisplayedFolded()
Node.SetProcessInternal(bool)
Node.IsProcessingInternal()
Node.SetPhysicsProcessInternal(bool)
Node.IsPhysicsProcessingInternal()
Node.IsPhysicsInterpolated()
Node.IsPhysicsInterpolatedAndEnabled()
Node.ResetPhysicsInterpolation()
Node.SetTranslationDomainInherited()
Node.GetWindow()
Node.GetLastExclusiveWindow()
Node.GetTree()
Node.CreateTween()
Node.Duplicate(int)
Node.ReplaceBy(Node, bool)
Node.setSceneInstanceLoadPlaceholder(bool)
Node.GetSceneInstanceLoadPlaceholder()
Node.setEditableInstance(Node, bool)
Node.setEditableInstance(Node)
Node.GetViewport()
Node.QueueFree()

```
Node.RequestReady()
Node.IsNodeReady()
Node.SetMultiplayerAuthority(int, bool)
Node.GetMultiplayerAuthority()
Node.IsMultiplayerAuthority()
Node.RpcConfig(StringName, Variant)
Node.GetRpcConfig()
Node.Atr(string, StringName)
Node.AtrN(string, StringName, int, StringName)
Node.Rpc(StringName, params Variant[])
Node.Rpc(StringName, ReadOnlySpan<Variant>)
Node.RpcId(long, StringName, params Variant[])
Node.RpcId(long, StringName, ReadOnlySpan<Variant>)
Node.UpdateConfigurationWarnings()
Node.CallDeferredThreadGroup(StringName, params Variant[])
Node.CallDeferredThreadGroup(StringName, ReadOnlySpan<Variant>)
Node.SetDeferredThreadGroup(StringName, Variant)
Node.NotifyDeferredThreadGroup(int)
Node.CallThreadSafe(StringName, params Variant[])
Node.CallThreadSafe(StringName, ReadOnlySpan<Variant>)
Node.SetThreadSafe(StringName, Variant)
Node.NotifyThreadSafe(int)
Node.EmitSignalReady()
Node.EmitSignalRenamed()
Node.EmitSignalTreeEntered()
Node.EmitSignalTreeExiting()
Node.EmitSignalTreeExited()
Node.EmitSignalChildEnteredTree(Node)
Node.EmitSignalChildExitingTree(Node)
Node.EmitSignalChildOrderChanged()
Node.EmitSignalReplacingBy(Node)
Node.EmitSignalEditorDescriptionChanged(Node)
Node.EmitSignalEditorStateChanged()
Node.InvokeGodotClassMethod(in godot_string_name, NativeVariantPtrArgs, out godot_variant)
Node.HasGodotClassMethod(in godot_string_name)
Node.HasGodotClassSignal(in godot_string_name)
Node.Name
Node.UniqueNameInOwner
Node.SceneFilePath
Node.Owner
Node.Multiplayer
Node.ProcessMode
Node.ProcessPriority
Node.ProcessPhysicsPriority
Node.ProcessThreadGroup
Node.ProcessThreadGroupOrder
Node.ProcessThreadMessages
Node.PhysicsInterpolationMode
Node.AutoTranslateMode
Node.EditorDescription
```

Node.Ready
Node.Renamed
Node.TreeEntered
Node.TreeExiting
Node.TreeExited
Node.ChildEnteredTree
Node.ChildExitingTree
Node.ChildOrderChanged
Node.ReplacingBy
Node.EditorDescriptionChanged
Node.EditorStateChanged
GodotObject.NotificationPostinitialize
GodotObject.NotificationPredelete
GodotObject.NotificationExtensionReloaded
GodotObject.InstanceFromId(ulong)
GodotObject.IsInstanceIdValid(ulong)
GodotObject.IsValid(GodotObject)
GodotObject.WeakRef(GodotObject)
GodotObject.Dispose()
GodotObject.Dispose(bool)
GodotObject.ToString()
GodotObject.ToSignal(GodotObject, StringName)
GodotObject._Get(StringName)
GodotObject._GetPropertyList()
GodotObject._IterGet(Variant)
GodotObject._IterInit(Array)
GodotObject._IterNext(Array)
GodotObject._Notification(int)
GodotObject._PropertyCanRevert(StringName)
GodotObject._PropertyGetRevert(StringName)
GodotObject._Set(StringName, Variant)
GodotObject._ValidateProperty(Dictionary)
GodotObject.Free()
GodotObject.GetClass()
GodotObject.IsClass(string)
GodotObject.Set(StringName, Variant)
GodotObject.Get(StringName)
GodotObject.SetIndexed(NodePath, Variant)
GodotObject.GetIndexed(NodePath)
GodotObject.GetPropertyList()
GodotObject.GetMethodList()
GodotObject.PropertyCanRevert(StringName)
GodotObject.PropertyGetRevert(StringName)
GodotObject.Notification(int, bool)
GodotObject.GetInstanceId()
GodotObject.SetScript(Variant)
GodotObject.GetScript()
GodotObject.SetMeta(StringName, Variant)
GodotObject.RemoveMeta(StringName)
GodotObject.GetMeta(StringName, Variant)

GodotObject.HasMeta(StringName)
GodotObject.GetMetaList()
GodotObject.AddUserSignal(string, Array)
GodotObject.HasUserSignal(StringName)
GodotObject.RemoveUserSignal(StringName)
GodotObject.EmitSignal(StringName, params Variant[])
GodotObject.EmitSignal(StringName, ReadOnlySpan<Variant>)
GodotObject.Call(StringName, params Variant[])
GodotObject.Call(StringName, ReadOnlySpan<Variant>)
GodotObject.CallDeferred(StringName, params Variant[])
GodotObject.CallDeferred(StringName, ReadOnlySpan<Variant>)
GodotObject.SetDeferred(StringName, Variant)
GodotObject.Callv(StringName, Array)
GodotObject.HasMethod(StringName)
GodotObject.GetMethodArgumentCount(StringName)
GodotObject.HasSignal(StringName)
GodotObject.GetSignalList()
GodotObject.GetSignalConnectionList(StringName)
GodotObject.GetIncomingConnections()
GodotObject.Connect(StringName, Callable, uint)
GodotObject.Disconnect(StringName, Callable)
GodotObject.IsConnected(StringName, Callable)
GodotObject.HasConnections(StringName)
GodotObject.SetBlockSignals(bool)
GodotObject.IsBlockingSignals()
GodotObject.NotifyPropertyListChanged()
GodotObject.SetMessageTranslation(bool)
GodotObject.CanTranslateMessages()
GodotObject.Tr(StringName, StringName)
GodotObject.TrN(StringName, StringName, int, StringName)
GodotObject.GetTranslationDomain()
GodotObject.SetTranslationDomain(StringName)
GodotObject.IsQueuedForDeletion()
GodotObject.CancelFree()
GodotObject.EmitSignalScriptChanged()
GodotObject.EmitSignalPropertyListChanged()
GodotObject.NativeInstance
GodotObject.ScriptChanged
GodotObject.PropertyListChanged
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
[ScriptPath("res://Modules/cfGodotEngine/util/StateMachineNode/StateNode.cs")]
public abstract class StateNode<TStateId, TState, TStateMachine> : Node, IDisposable where TSt
```

Type Parameters

NAME

TStateId

TState

TStateMachine

Properties

Id

Declaration

```
public abstract TStateId Id { get; }
```

Property Value

TYPE

TStateld

stateMachine

Declaration

```
protected TStateMachine stateMachine { get; }
```

Property Value

TYPE

TStateMachine

Methods

CanUpdate()

Declaration

```
public virtual bool CanUpdate()
```

Returns

TYPE

bool

IsReady(StateParam)

Declaration

```
public virtual bool IsReady(StateParam param)
```

Parameters

TYPE	NAME
StateParam	param

Returns

TYPE

bool

SetStateMachine(TStateMachine)

Declaration

```
public void SetStateMachine(TStateMachine stateMachine)
```

Parameters

TYPE	NAME
TStateMachine	stateMachine

_OnEndContext()

Declaration

```
protected virtual void _OnEndContext()
```

_Process(double)

Declaration

```
public virtual void _Process(double delta)
```

Parameters

TYPE	NAME
double	delta

_Ready()

Declaration

```
public virtual void _Ready()
```

_StartContext(StateParam)

Declaration

```
protected abstract void _StartContext(StateParam param)
```

Parameters

TYPE	NAME
StateParam	param

Implements

Extension Methods

`NodeUtil.DontDestroyOnLoad(Node)`

© Bus Fighter. All rights reserved.

Class StateNode<TStatId, TState, TStateMachine>.MethodName

Cached StringNames for the methods contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject](#).[MethodName](#) → [Node](#).[MethodName](#) → [StateNode<TStatId, TState, TStateMachine>.MethodName](#)

Inherited Members

[Node](#).[MethodName._EnterTree](#)
[Node](#).[MethodName._ExitTree](#)
[Node](#).[MethodName._GetConfigurationWarnings](#)
[Node](#).[MethodName._Input](#)
[Node](#).[MethodName._PhysicsProcess](#)
[Node](#).[MethodName._ShortcutInput](#)
[Node](#).[MethodName._UnhandledInput](#)
[Node](#).[MethodName._UnhandledKeyInput](#)
[Node](#).[MethodName.PrintOrphanNodes](#)
[Node](#).[MethodName.AddSibling](#)
[Node](#).[MethodName.SetName](#)
[Node](#).[MethodName.GetName](#)
[Node](#).[MethodName.AddChild](#)
[Node](#).[MethodName.RemoveChild](#)
[Node](#).[MethodName.Reparent](#)
[Node](#).[MethodName.GetChildCount](#)
[Node](#).[MethodName.GetChildren](#)
[Node](#).[MethodName.GetChild](#)
[Node](#).[MethodName.HasNode](#)
[Node](#).[MethodName.GetNode](#)
[Node](#).[MethodName.GetNodeOrNull](#)
[Node](#).[MethodName.GetParent](#)
[Node](#).[MethodName.FindChild](#)
[Node](#).[MethodName.FindChildren](#)
[Node](#).[MethodName.FindParent](#)
[Node](#).[MethodName.HasNodeAndResource](#)
[Node](#).[MethodName.GetNodeAndResource](#)
[Node](#).[MethodName.IsInsideTree](#)
[Node](#).[MethodName.IsPartOfEditedScene](#)
[Node](#).[MethodName.IsAncestorOf](#)
[Node](#).[MethodName.IsGreaterThan](#)
[Node](#).[MethodName.GetPath](#)

Node.MethodName.GetPathTo
Node.MethodName.AddToGroup
Node.MethodName.RemoveFromGroup
Node.MethodName.IsInGroup
Node.MethodName.MoveChild
Node.MethodName.GetGroups
Node.MethodName.SetOwner
Node.MethodName.GetOwner
Node.MethodName.GetIndex
Node.MethodName.PrintTree
Node.MethodName.PrintTreePretty
Node.MethodName.GetTreeString
Node.MethodName.GetTreeStringPretty
Node.MethodName.SetSceneFilePath
Node.MethodName.GetSceneFilePath
Node.MethodName.PropagateNotification
Node.MethodName.PropagateCall
Node.MethodName.SetPhysicsProcess
Node.MethodName.GetPhysicsProcessDeltaTime
Node.MethodName.IsPhysicsProcessing
Node.MethodName.GetProcessDeltaTime
Node.MethodName.SetProcess
Node.MethodName.SetProcessPriority
Node.MethodName.GetProcessPriority
Node.MethodName.SetPhysicsProcessPriority
Node.MethodName.GetPhysicsProcessPriority
Node.MethodName.IsProcessing
Node.MethodName.SetProcessInput
Node.MethodName.IsProcessingInput
Node.MethodName.SetProcessShortcutInput
Node.MethodName.IsProcessingShortcutInput
Node.MethodName.SetProcessUnhandledInput
Node.MethodName.IsProcessingUnhandledInput
Node.MethodName.SetProcessUnhandledKeyInput
Node.MethodName.IsProcessingUnhandledKeyInput
Node.MethodName.SetProcessMode
Node.MethodName.GetProcessMode
Node.MethodName.CanProcess
Node.MethodName.SetProcessThreadGroup
Node.MethodName.GetProcessThreadGroup
Node.MethodName.SetProcessThreadMessages
Node.MethodName.GetProcessThreadMessages
Node.MethodName.SetProcessThreadGroupOrder
Node.MethodName.GetProcessThreadGroupOrder
Node.MethodName.SetDisplayFolded
Node.MethodName.IsDisplayedFolded
Node.MethodName.SetProcessInternal
Node.MethodName.IsProcessingInternal
Node.MethodName.SetPhysicsProcessInternal
Node.MethodName.IsPhysicsProcessingInternal

Node.MethodName.SetPhysicsInterpolationMode
Node.MethodName.GetPhysicsInterpolationMode
Node.MethodName.IsPhysicsInterpolated
Node.MethodName.IsPhysicsInterpolatedAndEnabled
Node.MethodName.ResetPhysicsInterpolation
Node.MethodName.SetAutoTranslateMode
Node.MethodName.GetAutoTranslateMode
Node.MethodName.SetTranslationDomainInherited
Node.MethodName.GetWindow
Node.MethodName.GetLastExclusiveWindow
Node.MethodName.GetTree
Node.MethodName.CreateTween
Node.MethodName.Duplicate
Node.MethodName.ReplaceBy
Node.MethodName.SetSceneInstanceLoadPlaceholder
Node.MethodName.GetSceneInstanceLoadPlaceholder
Node.MethodName.SetEditableInstance
Node.MethodName.IsEditableInstance
Node.MethodName.GetViewport
Node.MethodName.QueueFree
Node.MethodName.RequestReady
Node.MethodName.IsNodeReady
Node.MethodName.SetMultiplayerAuthority
Node.MethodName.GetMultiplayerAuthority
Node.MethodName.IsMultiplayerAuthority
Node.MethodName.GetMultiplayer
Node.MethodName.RpcConfig
Node.MethodName.GetRpcConfig
Node.MethodName.SetEditorDescription
Node.MethodName.GetEditorDescription
Node.MethodName._SetImportPath
Node.MethodName._GetImportPath
Node.MethodName.SetUniqueNameInOwner
Node.MethodName.IsUniqueNameInOwner
Node.MethodName.Atr
Node.MethodName.AtrN
Node.MethodName.Rpc
Node.MethodName.RpcId
Node.MethodName.UpdateConfigurationWarnings
Node.MethodName.CallDeferredThreadGroup
Node.MethodName.SetDeferredThreadGroup
Node.MethodName.NotifyDeferredThreadGroup
Node.MethodName.CallThreadSafe
Node.MethodName.SetThreadSafe
Node.MethodName.NotifyThreadSafe
GodotObject.MethodName._Get
GodotObject.MethodName._GetPropertyList
GodotObject.MethodName._IterGet
GodotObject.MethodName._IterInit
GodotObject.MethodName._IterNext

GodotObject.MethodName._Notification
GodotObject.MethodName._PropertyCanRevert
GodotObject.MethodName._PropertyGetRevert
GodotObject.MethodName._Set
GodotObject.MethodName._ValidateProperty
GodotObject.MethodName.Free
GodotObject.MethodName.GetClass
GodotObject.MethodName.IsClass
GodotObject.MethodName.Set
GodotObject.MethodName.Get
GodotObject.MethodName.SetIndexed
GodotObject.MethodName.GetIndexed
GodotObject.MethodName.GetPropertyList
GodotObject.MethodName.GetMethodList
GodotObject.MethodName.PropertyCanRevert
GodotObject.MethodName.PropertyGetRevert
GodotObject.MethodName.Notification
GodotObject.MethodName.GetInstanceId
GodotObject.MethodName.SetScript
GodotObject.MethodName.GetScript
GodotObject.MethodName.SetMeta
GodotObject.MethodName.RemoveMeta
GodotObject.MethodName.GetMeta
GodotObject.MethodName.HasMeta
GodotObject.MethodName.GetMetaList
GodotObject.MethodName.AddUserSignal
GodotObject.MethodName.HasUserSignal
GodotObject.MethodName.RemoveUserSignal
GodotObject.MethodName.EmitSignal
GodotObject.MethodName.Call
GodotObject.MethodName.CallDeferred
GodotObject.MethodName.SetDeferred
GodotObject.MethodName.Callv
GodotObject.MethodName.HasMethod
GodotObject.MethodName.GetMethodArgumentCount
GodotObject.MethodName.HasSignal
GodotObject.MethodName.GetSignalList
GodotObject.MethodName.GetSignalConnectionList
GodotObject.MethodName.GetIncomingConnections
GodotObject.MethodName.Connect
GodotObject.MethodName.Disconnect
GodotObject.MethodName.IsConnected
GodotObject.MethodName.HasConnections
GodotObject.MethodName.SetBlockSignals
GodotObject.MethodName.IsBlockingSignals
GodotObject.MethodName.NotifyPropertyListChanged
GodotObject.MethodName.SetMessageTranslation
GodotObject.MethodName.CanTranslateMessages
GodotObject.MethodName.Tr
GodotObject.MethodName.TrN

GodotObject.MethodName.GetTranslationDomain
GodotObject.MethodName.SetTranslationDomain
GodotObject.MethodName.IsQueuedForDeletion
GodotObject.MethodName.CancelFree
object.Equals(object)
object.Equals(object, object)
object.GetHashCode()
object.GetType()
object.MemberwiseClone()
object.ReferenceEquals(object, object)
object.ToString()

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class StateNode<TStateId, TState, TStateMachine>.MethodName : Node.MethodName
```

Fields

CanUpdate

Cached name for the 'CanUpdate' method.

Declaration

```
public static readonly StringName CanUpdate
```

Field Value

TYPE

StringName

OnEndContext

Cached name for the 'OnEndContext' method.

Declaration

```
public static readonly StringName OnEndContext
```

Field Value

TYPE

StringName

_OnEndContext

Cached name for the '_OnEndContext' method.

Declaration

```
public static readonly StringName _OnEndContext
```

Field Value

TYPE

StringName

_Process

Cached name for the '_Process' method.

Declaration

```
public static readonly StringName _Process
```

Field Value

TYPE

StringName

_Ready

Cached name for the '_Ready' method.

Declaration

```
public static readonly StringName _Ready
```

TYPE

StringName

© Bus Fighter. All rights reserved.

Class `StateNode<TStatId, TState, TStateMachine>.PropertyName`

Cached StringNames for the properties and fields contained in this class, for fast lookup.

Inheritance

`object` → `GodotObject.PropertyName` → `Node.PropertyName` → `StateNode<TStatId, TState, TStateMachine>.PropertyName`

Inherited Members

`Node.PropertyName._ImportPath`

`Node.PropertyName.Name`

`Node.PropertyName.UniqueNameInOwner`

`Node.PropertyName.SceneFilePath`

`Node.PropertyName.Owner`

`Node.PropertyName.Multiplayer`

`Node.PropertyName.ProcessMode`

`Node.PropertyName.ProcessPriority`

`Node.PropertyName.ProcessPhysicsPriority`

`Node.PropertyName.ProcessThreadGroup`

`Node.PropertyName.ProcessThreadGroupOrder`

`Node.PropertyName.ProcessThreadMessages`

`Node.PropertyName.PhysicsInterpolationMode`

`Node.PropertyName.AutoTranslateMode`

`Node.PropertyName.EditorDescription`

`object.Equals(object)`

`object.Equals(object, object)`

`object.GetHashCode()`

`object.GetType()`

`object.MemberwiseClone()`

`object.ReferenceEquals(object, object)`

`object.ToString()`

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class StateNode<TStateId, TState, TStateMachine>.PropertyName : Node.PropertyName
```

stateNodes

Cached name for the 'stateNodes' field.

Declaration

```
public static readonly StringName stateNodes
```

Field Value

TYPE

StringName

© Bus Fighter. All rights reserved.

Class StateNode<TStatId, TState, TStateMachine>.SignalName

Cached StringNames for the signals contained in this class, for fast lookup.

Inheritance

[object](#) → [GodotObject.SignalName](#) → [Node.SignalName](#) → [StateNode<TStatId, TState, TStateMachine>.SignalName](#)

Inherited Members

[Node.SignalName.Ready](#)
[Node.SignalName.Renamed](#)
[Node.SignalName.TreeEntered](#)
[Node.SignalName.TreeExiting](#)
[Node.SignalName.TreeExited](#)
[Node.SignalName.ChildEnteredTree](#)
[Node.SignalName.ChildExitingTree](#)
[Node.SignalName.ChildOrderChanged](#)
[Node.SignalName.ReplacingBy](#)
[Node.SignalName.EditorDescriptionChanged](#)
[Node.SignalName.EditorStateChanged](#)
[GodotObject.SignalName.ScriptChanged](#)
[GodotObject.SignalName.PropertyListChanged](#)
[object.Equals\(object\)](#)
[object.Equals\(object, object\)](#)
[object.GetHashCode\(\)](#)
[object.GetType\(\)](#)
[object.MemberwiseClone\(\)](#)
[object.ReferenceEquals\(object, object\)](#)
[object.ToString\(\)](#)

Namespace: [cfGodotEngine.Util](#)

Assembly: CatSweeper.dll

Syntax

```
public class StateNode<TStateId, TState, TStateMachine>.SignalName : Node.SignalName
```