Namespace Cobilas.GodotEngine.GDLua

Classes

LuaContainer

Represents an in-memory Lua code builder and executor integrated with the Godot engine.

LuaFile

Represents a Lua file that can be loaded, executed, and manipulated within the Godot engine.

LuaSerializableAttribute

Indicates that a type is serializable for Lua interoperability.

ObjectToLuaTable

Provides a base class for converting between C# objects and Lua tables.

Structs

LuaContainerConfg

Represents the configuration used to initialize a LuaContainer instance.

LuaField

Represents a Lua field with a name and value, providing type conversion capabilities and disposable pattern for resource management.

LuaFileConfg

Represents configuration settings for Lua file operations in the Godot engine.

LuaTableItem

Represents a Lua table item that can contain multiple table elements and supports enumeration.

LuaTableValue

Represents a basic Lua table element with a name and value.

Class LuaContainer

Namespace: Cobilas.GodotEngine.GDLua

Assembly: com.cobilas.godot.lua.dll

Represents an in-memory Lua code builder and executor integrated with the Godot engine.

```
public sealed class LuaContainer : IDisposable, ILuaFile
```

Inheritance

object ← LuaContainer

Implements

IDisposable, ILuaFile

Inherited Members

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

Remarks

This class implements ILuaFile and provides methods to dynamically create, manipulate, and execute Lua code through an internal StringBuilder buffer. It allows the construction of Lua scripts programmatically and their execution using NLua. Implements IDisposable for proper resource cleanup.

Constructors

LuaContainer(LuaContainerConfg)

Initializes a new instance of the LuaContainer class with the specified configuration.

```
public LuaContainer(LuaContainerConfg confg)
```

Parameters

`confg` LuaContainerConfg

The configuration used to initialize the Lua state and related options.

Exceptions

ArgumentNullException

Thrown when a required configuration property is null.

Properties

Builder

Gets the current string content of the internal StringBuilder.

```
public string Builder { get; }
```

Property Value

string

Methods

ClearBuffer()

Clears the internal Lua code buffer.

```
public LuaContainer ClearBuffer()
```

Returns

LuaContainer

The current LuaContainer instance for chaining.

Exceptions

ObjectDisposedException

Thrown when the container has been disposed.

Dispose()

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

```
public void Dispose()
```

DoString(string)

Appends a raw Lua code string to the buffer.

```
public LuaContainer DoString(string value)
```

Parameters

`value` string

The Lua code string to append.

Returns

LuaContainer

The current LuaContainer instance for chaining.

Exceptions

ObjectDisposedException

Thrown when the container has been disposed.

FlushToLua()

Executes the current Lua buffer using the NLua interpreter.

```
public LuaContainer FlushToLua()
```

Returns

LuaContainer

The current LuaContainer instance for chaining.

Exceptions

ObjectDisposedException

Thrown when the container has been disposed.

GetField(string)

Retrieves a field from the Lua file by its path.

```
public LuaField GetField(string pathField)
```

Parameters

`pathField` string

The path to the field in the Lua file.

Returns

LuaField

A LuaField containing the field data.

Exceptions

ObjectDisposedException

Thrown when the container has been disposed.

InitCLRPackage(string)

Adds a CLR import statement to the Lua buffer.

```
public LuaContainer InitCLRPackage(string import)
```

Parameters

`import` string

The CLR namespace or type to import.

Returns

LuaContainer

The current LuaContainer instance for chaining.

Exceptions

ObjectDisposedException

Thrown when the container has been disposed.

InitField(string, object)

Initializes a global Lua field with a specific value.

```
public LuaContainer InitField(string pathField, object value)
```

Parameters

`pathField` string

The Lua field path to initialize.

`value` object

The value to assign to the field.

Returns

LuaContainer

The current LuaContainer instance for chaining.

Exceptions

ObjectDisposedException

Thrown when the container has been disposed.

InitFunction(string, string, params string[])

Initializes a Lua function with the specified name, body, and arguments.

```
public LuaContainer InitFunction(string funcName, string funcBody, params string[] funcArgs)
```

Parameters

`funcName` string

The name of the function.

`funcBody` string

The Lua code representing the function body.

`funcArgs` string[]

The arguments of the function.

Returns

LuaContainer

The current LuaContainer instance for chaining.

Exceptions

ObjectDisposedException

Thrown when the container has been disposed.

InitLoaclField(string, object)

Initializes a local Lua field with a specific value.

```
public LuaContainer InitLoaclField(string pathField, object value)
```

Parameters

`pathField` string

The Lua field name to initialize locally.

`value` object

The value to assign to the field.

Returns

LuaContainer

The current LuaContainer instance for chaining.

Exceptions

ObjectDisposedException

Thrown when the container has been disposed.

InitTable(LuaTableItem)

Initializes a Lua table by appending its string representation to the buffer.

```
public LuaContainer InitTable(LuaTableItem tables)
```

Parameters

`tables` LuaTableItem

The Lua table object to append.

Returns

LuaContainer

The current LuaContainer instance for chaining.

Exceptions

ObjectDisposedException

Thrown when the container has been disposed.

InvokeFunction(string, params object[])

Invokes a function defined in the Lua file.

```
public object[] InvokeFunction(string methodName, params object[] args)
```

Parameters

`methodName` string

The name of the function to invoke.

`args` object[]

The arguments to pass to the function.

Returns

object[]

An array of objects containing the function's return values.

Exceptions

ObjectDisposedException

Thrown when the container has been disposed.

LuaTableToObject<T>(string)

Converts a Lua table to an object of the specified type.

```
public LuaField LuaTableToObject<T>(string pathField)
```

Parameters

`pathField` string

The path to the Lua table in the file.

Returns

LuaField

A LuaField containing the converted table data.

Type Parameters

`T`

The target type to convert the Lua table to.

Exceptions

ObjectDisposedException

Thrown when the container has been disposed.

LuaTableToObject<T>(string, ref T)

Converts a Lua table to an object and assigns it to the provided reference.

```
public void LuaTableToObject<T>(string pathField, ref T value)
```

Parameters

`pathField` string

The path to the Lua table in the file.

`value` T

The reference variable to assign the converted table data to.

Type Parameters

`T`

The type of the object to assign.

Exceptions

ObjectDisposedException

Thrown when the container has been disposed.

InvalidCastException

Thrown when no converter is defined for the specified type.

SetField(string, object)

Sets the value of a field in the Lua file.

```
public void SetField(string pathField, object value)
```

Parameters

`pathField` string

The path to the field in the Lua file.

`value` object

The value to assign to the field.

Exceptions

ObjectDisposedException

Thrown when the container has been disposed.

Struct LuaContainerConfg

Namespace: Cobilas.GodotEngine.GDLua

Assembly: com.cobilas.godot.lua.dll

Represents the configuration used to initialize a LuaContainer instance.

```
public readonly struct LuaContainerConfg : ILuaContainerConfg
```

Implements
ILuaContainerConfg

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.ReferenceEquals(object, object) , object.GetType()

Remarks

This structure defines options for setting up the Lua environment, including whether to load CLR packages, open Lua libraries, and reuse an existing Lua state.

Constructors

LuaContainerConfg(Lua?, bool, bool)

Represents the configuration used to initialize a LuaContainer instance.

```
public LuaContainerConfg(Lua? luaState = null, bool useCLRPackage = false, bool openLibs
= true)
```

Parameters

`luaState` Lua

`useCLRPackage` bool

`openLibs` bool

Remarks

This structure defines options for setting up the Lua environment, including whether to load CLR packages, open Lua libraries, and reuse an existing Lua state.

Properties

Default

Gets a default configuration for the LuaContainer.

```
public static LuaContainerConfg Default { get; }
```

Property Value

LuaContainerConfg

Remarks

The default configuration uses no predefined Lua state, disables CLR package loading, and enables standard Lua libraries.

LuaState

Gets the current Lua state instance.

```
public Lua? LuaState { get; }
```

Property Value

Lua

The Lua state object, or null if not initialized.

OpenLibs

Gets a value indicating whether standard Lua libraries should be loaded.

```
public bool OpenLibs { get; }
```

Property Value

bool

true to load standard Lua libraries; otherwise, false.

UseCLRPackage

Gets a value indicating whether the CLR package should be available in Lua.

```
public bool UseCLRPackage { get; }
```

Property Value

bool

true to enable CLR package access from Lua; otherwise, false.

Struct LuaField

Namespace: Cobilas.GodotEngine.GDLua

Assembly: com.cobilas.godot.lua.dll

Represents a Lua field with a name and value, providing type conversion capabilities and disposable pattern for resource management.

```
public struct LuaField : IConvertible, IDisposable
```

Implements
IConvertible, IDisposable

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.ReferenceEquals(object, object) , object.GetType()

Remarks

This struct implements IConvertible for seamless type conversions and IDisposable for proper resource cleanup when working with Lua interop.

Constructors

LuaField(string, object?)

Represents a Lua field with a name and value, providing type conversion capabilities and disposable pattern for resource management.

```
public LuaField(string fieldName, object? value)
```

Parameters

`fieldName` string

`value` object

Remarks

This struct implements IConvertible for seamless type conversions and IDisposable for proper resource cleanup when working with Lua interop.

Properties

FieldName

Gets the name of the Lua field

```
public readonly string FieldName { get; }
```

Property Value

string

The name of the field.

Exceptions

ObjectDisposedException

Thrown when the LuaField has been disposed.

Value

Gets the value of the Lua field

```
public readonly object? Value { get; }
```

Property Value

object

The field value as an object.

Exceptions

ObjectDisposedException

Thrown when the LuaField has been disposed.

Methods

Dispose()

Releases all resources used by the LuaField

```
public void Dispose()
```

Exceptions

ObjectDisposedException

Thrown when the LuaField has already been disposed.

Operators

explicit operator LuaTable(LuaField)

Converts a LuaField to a LuaTable

```
public static explicit operator LuaTable(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

LuaTable

A LuaTable representation of the field value.

Exceptions

InvalidCastException

Thrown when the field value is not a LuaTable.

explicit operator byte(LuaField)

Converts a LuaField to a byte

```
public static explicit operator byte(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

byte

A byte representation of the field value.

explicit operator char(LuaField)

Converts a LuaField to a character

```
public static explicit operator char(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

char

A character representation of the field value.

explicit operator decimal(LuaField)

Converts a LuaField to a decimal number

```
public static explicit operator decimal(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

decimal

A decimal representation of the field value.

explicit operator double(LuaField)

Converts a LuaField to a double-precision floating-point number

```
public static explicit operator double(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

double

A double representation of the field value.

explicit operator short(LuaField)

Converts a LuaField to a signed 16-bit integer

```
public static explicit operator short(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

short

A signed 16-bit integer representation of the field value.

explicit operator int(LuaField)

Converts a LuaField to a signed 32-bit integer

```
public static explicit operator int(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

int

A signed 32-bit integer representation of the field value.

explicit operator long(LuaField)

Converts a LuaField to a signed 64-bit integer

```
public static explicit operator long(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

long

A signed 64-bit integer representation of the field value.

explicit operator sbyte(LuaField)

Converts a LuaField to a signed byte

```
public static explicit operator sbyte(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

sbyte

A signed byte representation of the field value.

explicit operator float(LuaField)

Converts a LuaField to a single-precision floating-point number

```
public static explicit operator float(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

float

A float representation of the field value.

explicit operator string(LuaField)

Converts a LuaField to a string

```
public static explicit operator string(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

string

A string representation of the field value.

explicit operator TypeCode(LuaField)

Converts a LuaField to its TypeCode

```
public static explicit operator TypeCode(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

TypeCode

The TypeCode of the field value.

explicit operator ushort(LuaField)

Converts a LuaField to an unsigned 16-bit integer

```
public static explicit operator ushort(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

ushort

An unsigned 16-bit integer representation of the field value.

explicit operator uint(LuaField)

Converts a LuaField to an unsigned 32-bit integer

```
public static explicit operator uint(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

uint

An unsigned 32-bit integer representation of the field value.

explicit operator ulong(LuaField)

Converts a LuaField to an unsigned 64-bit integer

```
public static explicit operator ulong(LuaField f)
```

Parameters

`f` LuaField

The LuaField to convert.

Returns

ulong

An unsigned 64-bit integer representation of the field value.

Class LuaFile

Namespace: Cobilas.GodotEngine.GDLua

Assembly: com.cobilas.godot.lua.dll

Represents a Lua file that can be loaded, executed, and manipulated within the Godot engine.

```
public sealed class LuaFile : IDisposable, ILuaFile
```

Inheritance

object ← LuaFile

Implements

IDisposable, ILuaFile

Inherited Members

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

Remarks

This class provides methods to interact with Lua scripts, including reading fields, setting values, invoking functions, and converting Lua tables to C# objects. Implements IDisposable for proper resource cleanup.

Constructors

LuaFile(LuaFileConfg)

Initializes a new instance of the LuaFile class with the specified configuration.

```
public LuaFile(LuaFileConfg confg)
```

Parameters

`confg` LuaFileConfg

The configuration settings for the Lua file.

Exceptions

ArgumentNullException

Thrown when the file path in configuration is null.

DirectoryNotFoundException

Thrown when the directory containing the Lua file is not found.

FileNotFoundException

Thrown when the specified Lua file is not found.

LuaFile(string, bool)

Initializes a new instance of the LuaFile class with the specified file path.

```
public LuaFile(string filePath, bool refreshBuffer = false)
```

Parameters

`filePath` string

The path to the Lua file.

`refreshBuffer` bool

Whether to refresh the file buffer on each access.

Methods

Dispose()

Releases all resources used by the LuaFile instance.

```
public void Dispose()
```

GetField(string)

Retrieves a field from the Lua file by its path.

```
public LuaField GetField(string pathField)
```

Parameters

`pathField` string

The path to the field in the Lua file.

Returns

LuaField

A LuaField containing the field data.

Exceptions

ObjectDisposedException

Thrown when the LuaFile has been disposed.

InvokeFunction(string, params object[])

Invokes a function defined in the Lua file.

```
public object[] InvokeFunction(string methodName, params object[] args)
```

Parameters

`methodName` string

The name of the function to invoke.

`args` object[]

The arguments to pass to the function.

Returns

object[]

An array of objects containing the function's return values.

Exceptions

ObjectDisposedException

Thrown when the LuaFile has been disposed.

LuaTableToObject<T>(string)

Converts a Lua table to an object of the specified type.

```
public LuaField LuaTableToObject<T>(string pathField)
```

Parameters

`pathField` string

The path to the Lua table in the file.

Returns

LuaField

A LuaField containing the converted table data.

Type Parameters

`T`

The target type to convert the Lua table to.

Exceptions

ObjectDisposedException

Thrown when the LuaFile has been disposed.

LuaTableToObject<T>(string, ref T)

Converts a Lua table to an object and assigns it to the provided reference.

```
public void LuaTableToObject<T>(string pathField, ref T value)
```

Parameters

`pathField` string

The path to the Lua table in the file.

`value` T

The reference variable to assign the converted table data to.

Type Parameters

`T`

The type of the object to assign.

Exceptions

ObjectDisposedException

Thrown when the LuaFile has been disposed.

InvalidCastException

Thrown when no converter is found for the specified type.

SetField(string, object)

Sets the value of a field in the Lua file.

```
public void SetField(string pathField, object value)
```

Parameters

`pathField` string

The path to the field in the Lua file.

`value` object

The value to assign to the field.

Exceptions

ObjectDisposedException

Thrown when the LuaFile has been disposed.

Struct LuaFileConfg

Namespace: Cobilas.GodotEngine.GDLua

Assembly: com.cobilas.godot.lua.dll

Represents configuration settings for Lua file operations in the Godot engine.

```
public readonly struct LuaFileConfg : ILuaFileConfg, ILuaContainerConfg
```

Implements
ILuaFileConfg, ILuaContainerConfg

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , ValueType.ToString() , object.Equals(object, object) , object.ReferenceEquals(object, object) , object.GetType()

Remarks

This structure provides configuration options for initializing and managing Lua script files, including file paths, Lua state management, and package settings.

Constructors

LuaFileConfg(string, Lua?, bool, bool, bool)

Represents configuration settings for Lua file operations in the Godot engine.

```
public LuaFileConfg(string filePath, Lua? luaState = null, bool useCLRPackage = false, bool
refreshBuffer = false, bool openLibs = true)
```

Parameters

`filePath` string

The path to the Lua script file.

`luaState` Lua

The existing Lua state instance to use, or null to create a new one.

`useCLRPackage` bool

Whether to enable CLR package access from Lua scripts.

`refreshBuffer` bool

Whether to refresh the file buffer on each access.

`openLibs` bool

Whether to load standard Lua libraries.

Remarks

This structure provides configuration options for initializing and managing Lua script files, including file paths, Lua state management, and package settings.

Properties

FilePath

Gets the file path to the Lua script.

```
public string? FilePath { get; }
```

Property Value

string

The path to the Lua file, or null if not specified.

LuaState

Gets the current Lua state instance.

```
public Lua? LuaState { get; }
```

Property Value

Lua

The Lua state object, or null if not initialized.

OpenLibs

Gets a value indicating whether standard Lua libraries should be loaded.

```
public bool OpenLibs { get; }
```

Property Value

bool

true to load standard Lua libraries; otherwise, false.

RefreshBuffer

Gets a value indicating whether the file buffer should be refreshed on each access.

```
public bool RefreshBuffer { get; }
```

Property Value

bool

true to refresh the buffer on each access; otherwise, false.

UseCLRPackage

Gets a value indicating whether the CLR package should be available in Lua.

```
public bool UseCLRPackage { get; }
```

Property Value

bool

true to enable CLR package access from Lua; otherwise, false.

Class LuaSerializableAttribute

Namespace: Cobilas.GodotEngine.GDLua

Assembly: com.cobilas.godot.lua.dll

Indicates that a type is serializable for Lua interoperability.

```
[AttributeUsage(AttributeTargets.Class|AttributeTargets.Struct, Inherited = false,
AllowMultiple = true)]
public sealed class LuaSerializableAttribute : Attribute, _Attribute
```

Inheritance

object ← Attribute ← LuaSerializableAttribute

Implements

_Attribute

Inherited Members

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

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

Remarks

Apply this attribute to classes or structures to mark them as serializable for communication between C# and Lua environments in the Godot engine. The attribute can be applied multiple times to handle multiple target types.

Constructors

LuaSerializableAttribute(Type)

Indicates that a type is serializable for Lua interoperability.

```
public LuaSerializableAttribute(Type typeTarget)
```

Parameters

`typeTarget` Type

The target type that this serialization converter supports.

Remarks

Apply this attribute to classes or structures to mark them as serializable for communication between C# and Lua environments in the Godot engine. The attribute can be applied multiple times to handle multiple target types.

Properties

TypeTarget

Gets the target type that this serialization converter supports.

```
public Type TypeTarget { get; }
```

Property Value

Type

The Type that can be serialized to and from Lua tables.

Struct LuaTableItem

Namespace: Cobilas.GodotEngine.GDLua

Assembly: com.cobilas.godot.lua.dll

Represents a Lua table item that can contain multiple table elements and supports enumeration.

```
public readonly struct LuaTableItem : ILuaTableItem, ILuaTable,
IEnumerable<ILuaTable>, IEnumerable
```

Implements

ILuaTableItem, ILuaTable, IEnumerable<ILuaTable>, IEnumerable

Inherited Members

ValueType.Equals(object) , ValueType.GetHashCode() , object.Equals(object, object) , object.ReferenceEquals(object, object) , object.GetType()

Remarks

This structure implements ILuaTableItem to provide collection capabilities for Lua table structures with hierarchical organization and string representation.

Constructors

LuaTableItem(string, params ILuaTable[])

Represents a Lua table item that can contain multiple table elements and supports enumeration.

```
public LuaTableItem(string name, params ILuaTable[] items)
```

Parameters

`name` string

The name identifier for the table item.

`items` ILuaTable[]

The array of Lua table elements to initialize the table with.

Remarks

This structure implements ILuaTableItem to provide collection capabilities for Lua table structures with hierarchical organization and string representation.

Properties

Count

Gets the number of table elements contained in this table item.

```
public long Count { get; }
```

Property Value

long

The total count of table elements.

this[int]

Gets the table element at the specified index.

```
public ILuaTable this[int index] { get; }
```

Parameters

`index` int

The zero-based index of the table element to get.

Property Value

ILuaTable

The ILuaTable at the specified index.

this[string]

Gets the table element with the specified name.

```
public ILuaTable this[string name] { get; }
```

Parameters

`name` string

The name of the table element to get.

Property Value

ILuaTable

The ILuaTable with the specified name.

Name

Gets the name of the Lua table element.

```
public string Name { get; }
```

Property Value

string

The name identifier of the table element.

Tables

Gets an array of all table elements contained in this table item.

```
public ILuaTable[] Tables { get; }
```

Property Value

ILuaTable[]

An array of ILuaTable elements.

Methods

GetEnumerator()

Returns an enumerator that iterates through the collection.

```
public IEnumerator<ILuaTable> GetEnumerator()
```

Returns

IEnumerator <ILuaTable>

An enumerator that can be used to iterate through the collection.

ToString()

Returns a string representation of the Lua table hierarchy.

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

Returns

string

A formatted string showing the table structure with proper indentation.

Struct LuaTableValue

Namespace: Cobilas.GodotEngine.GDLua

Assembly: com.cobilas.godot.lua.dll

Represents a basic Lua table element with a name and value.

```
public readonly struct LuaTableValue : ILuaTable
```

Implements
ILuaTable

Inherited Members
ValueType.Equals(object) , ValueType.GetHashCode() , object.Equals(object, object) , object.ReferenceEquals(object, object) , object.GetType()

Remarks

This structure implements ILuaTable to provide a simple key-value pair representation for Lua table elements.

Constructors

LuaTableValue(string, object)

Represents a basic Lua table element with a name and value.

```
public LuaTableValue(string name, object value)
```

Parameters

`name` string

The name identifier of the table element.

`value` object

The value stored in the table element.

Remarks

This structure implements ILuaTable to provide a simple key-value pair representation for Lua table elements.

Properties

Name

Gets the name of the Lua table element.

```
public string Name { get; }
```

Property Value

string

The name identifier of the table element.

Value

Gets the value stored in the Lua table element.

```
public object Value { get; }
```

Property Value

object

The value contained in the table element.

Methods

ToString()

Returns a string representation of the table element in "name = value" format.

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

Returns

string

A formatted string showing the name-value pair.

Class ObjectToLuaTable

Namespace: Cobilas.GodotEngine.GDLua

Assembly: com.cobilas.godot.lua.dll

Provides a base class for converting between C# objects and Lua tables.

```
public abstract class ObjectToLuaTable
```

Inheritance

object ← ObjectToLuaTable

Inherited Members

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

Remarks

This abstract class serves as the foundation for type-specific converters that handle serialization and deserialization between C# objects and Lua tables in the Godot engine's Lua integration system.

Methods

ToLuaTable(object?, LuaTable?)

Converts a C# object to a Lua table.

```
public abstract void ToLuaTable(object? obj, LuaTable? table)
```

Parameters

`obj` object

The C# object to convert.

`table` LuaTable

The Lua table to populate with the object's data.

ToObject(object?, LuaTable?)

Converts a Lua table back to a C# object.

```
public abstract object ToObject(object? obj, LuaTable? table)
```

Parameters

`obj` object

The original object instance (may be used for context).

`table` LuaTable

The Lua table containing the serialized data.

Returns

object

A C# object reconstructed from the Lua table data.

TryGetValue(Type, out ObjectToLuaTable)

Attempts to retrieve a converter for the specified type.

```
public static bool TryGetValue(Type type, out ObjectToLuaTable value)
```

Parameters

`type` Type

The type to find a converter for.

`value` ObjectToLuaTable

When this method returns, contains the converter associated with the specified type, if found; otherwise, null.

Returns

bool

true if a converter for the specified type was found; otherwise, false.

Namespace Cobilas.GodotEngine.GDLua.Interfaces

Interfaces

ILuaContainerConfg
: Defines configuration settings for a Lua container in the Godot engine.

ILuaFile
: Defines operations for interacting with Lua files and their contents.

ILuaFileConfg
: Defines configuration settings for Lua file operations in the Godot engine.

ILuaTable
: Represents a basic Lua table element with a name and value.

ILuaTableItem
: Represents a Lua table item that can contain multiple table elements and supports enumeration.

Interface ILuaContainerConfg

Namespace: Cobilas.GodotEngine.GDLua.Interfaces

Assembly: com.cobilas.godot.lua.dll

Defines configuration settings for a Lua container in the Godot engine.

```
public interface ILuaContainerConfg
```

Remarks

This interface provides the basic configuration options required for initializing and managing Lua environments within Godot.

Properties

LuaState

Gets the current Lua state instance.

```
Lua? LuaState { get; }
```

Property Value

Lua

The Lua state object, or null if not initialized.

OpenLibs

Gets a value indicating whether standard Lua libraries should be loaded.

```
bool OpenLibs { get; }
```

Property Value

bool

true to load standard Lua libraries; otherwise, false.

UseCLRPackage

Gets a value indicating whether the CLR package should be available in Lua.

```
bool UseCLRPackage { get; }
```

Property Value

bool

true to enable CLR package access from Lua; otherwise, false.

Interface ILuaFile

Namespace: Cobilas.GodotEngine.GDLua.Interfaces

Assembly: com.cobilas.godot.lua.dll

Defines operations for interacting with Lua files and their contents.

```
public interface ILuaFile
```

Remarks

Provides methods to read, write, and invoke elements within Lua script files.

Methods

GetField(string)

Retrieves a field from the Lua file by its path.

```
LuaField GetField(string pathField)
```

Parameters

`pathField` string

The path to the field in the Lua file.

Returns

LuaField

A LuaField containing the field data.

InvokeFunction(string, params object[])

Invokes a function defined in the Lua file.

```
object[] InvokeFunction(string methodName, params object[] args)
```

Parameters

`methodName` string

The name of the function to invoke.

`args` object[]

The arguments to pass to the function.

Returns

object[]

An array of objects containing the function's return values.

LuaTableToObject<T>(string)

Converts a Lua table to an object of the specified type.

```
LuaField LuaTableToObject<T>(string pathField)
```

Parameters

`pathField` string

The path to the Lua table in the file.

Returns

LuaField

A LuaField containing the converted table data.

Type Parameters

`T`

The target type to convert the Lua table to.

LuaTableToObject<T>(string, ref T)

Converts a Lua table to an object and assigns it to the provided reference.

```
void LuaTableToObject<T>(string pathField, ref T value)
```

Parameters

`pathField` string

The path to the Lua table in the file.

`value` T

The reference variable to assign the converted table data to.

Type Parameters

`T`

The type of the object to assign.

SetField(string, object)

Sets the value of a field in the Lua file.

```
void SetField(string pathField, object value)
```

Parameters

`pathField` string

The path to the field in the Lua file.

`value` object

The value to assign to the field.

Interface ILuaFileConfg

Namespace: Cobilas.GodotEngine.GDLua.Interfaces

Assembly: com.cobilas.godot.lua.dll

Defines configuration settings for Lua file operations in the Godot engine.

```
public interface ILuaFileConfg : ILuaContainerConfg
```

Inherited Members
ILuaContainerConfg.OpenLibs , ILuaContainerConfg.LuaState , ILuaContainerConfg.UseCLRPackage

Remarks

Extends ILuaContainerConfg with file-specific configuration options for loading and executing Lua scripts from files.

Properties

FilePath

Gets the file path to the Lua script.

```
string? FilePath { get; }
```

Property Value

string

The path to the Lua file, or null if not specified.

RefreshBuffer

Gets a value indicating whether the file buffer should be refreshed on each access.

```
bool RefreshBuffer { get; }
```

Property Value

bool

true to refresh the buffer on each access; otherwise, false.

Interface ILuaTable

Namespace: Cobilas.GodotEngine.GDLua.Interfaces

Assembly: com.cobilas.godot.lua.dll

Represents a basic Lua table element with a name and value.

```
public interface ILuaTable
```

Properties

Name

Gets the name of the Lua table element.

```
string Name { get; }
```

Property Value

string

The name identifier of the table element.

Value

Gets the value stored in the Lua table element.

```
object Value { get; }
```

Property Value

object

The value contained in the table element.

Interface ILuaTableItem

Namespace: Cobilas.GodotEngine.GDLua.Interfaces

Assembly: com.cobilas.godot.lua.dll

Represents a Lua table item that can contain multiple table elements and supports enumeration.

```
public interface ILuaTableItem : ILuaTable, IEnumerable<ILuaTable>, IEnumerable
```

Inherited Members
ILuaTable.Name , ILuaTable.Value , IEnumerable<ILuaTable>.GetEnumerator()

Remarks

Extends ILuaTable to provide collection capabilities for Lua table structures.

Properties

Count

Gets the number of table elements contained in this table item.

```
long Count { get; }
```

Property Value

long

The total count of table elements.

this[int]

Gets the table element at the specified index.

```
ILuaTable this[int index] { get; }
```

Parameters

`index` int

The zero-based index of the table element to get.

Property Value

ILuaTable

The ILuaTable at the specified index.

this[string]

Gets the table element with the specified name.

```
ILuaTable this[string name] { get; }
```

Parameters

`name` string

The name of the table element to get.

Property Value

ILuaTable

The ILuaTable with the specified name.

Tables

Gets an array of all table elements contained in this table item.

```
ILuaTable[] Tables { get; }
```

Property Value

ILuaTable[]

An array of ILuaTable elements.