Table of Contents

GameDriver API Documentation v1.2

October 2020

Introducing GameDriver

Welcome to the GameDriver API Documentation.

Our objective is to provide you with the tools necessary to build comprehensive automated tests for your Unity projects, and ultimately reduce the dependency on manual testing. Automated tests should cover the "must have" pieces of functionality that are considered critical for your project, and are usually the types of tests that are repeated for each and every build. For example:

- Launching the application
- Logging on to the user account
- Starting the game and selecting a level
- Creating, customizing and saving a new character
- etc...

While the implementation of each of these examples will be different for every project, there will be common methods used to automate each. In this API you will find the basic tools needed to perform many common activities to validate the functionality of your project. If something is missing, let us know on the user forums.

This document reflects the state of the API at the time of this writing. Commands, their format and input parameters are subject to change. Please refer to the version of the API documentation that accompanied your version of GameDriver for optimal behavior.

For information on how to use GameDriver HierarchyPath to identify, interact with, and query objects in your project, please refer to the "Working with HierarchyPath in GameDriver" pdf included with the GameDriver documentation.

Thank you for choosing GameDriver!

Class Api

Inheritance

System.Object

Api

Namespace: gdio.unity_api

Assembly: cs.temp.dll.dll

Syntax

```
public static class Api
```

Fields

AUTOPLAY

Declaration

```
public static bool AUTOPLAY
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

SCREENSHOT_ON_FAILURE

API uses this flag to determine if a screen capture should be taken if a checkpoint fails.

Declaration

```
public static bool SCREENSHOT_ON_FAILURE
```

Field Value

TYPE	DESCRIPTION
System.Boolean	If set to true a screen capture will automatically be taken on checkpoint failure. DEFAULT: false

Properties

ResultsPath

Declaration

```
public static string ResultsPath { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

ActivateInputField(String)

Use this function to activate a UI.InputField or TextMeshPro Input Field

Declaration

```
public static void ActivateInputField(string obj)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	The HierarchyPath to the object.

CallMethod(String, String, String[], Int32)

Use this function to call a method of a class. The class must be accessible via HierarchyPath, i.e. a Script Component on a GameObject

Declaration

```
public static bool CallMethod(string obj, string method, string[] arguments, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	The HierarchyPath to the object.
System.String	method	The method/function to be called.
System.String[]	arguments	
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

Returns

TYPE	DESCRIPTION
System.Boolean	True if the field value is found, false if it is not found before waitMaxSeconds.

CallMethod<T>(String, String, Object, Int32)

Use this function to call a method of a class. The class must be accessible via HierarchyPath, i.e. a Script Component on a GameObject.

Declaration

```
public static T CallMethod<T>(string obj, string method, object argument, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	The HierarchyPath to the object.

TYPE	NAME	DESCRIPTION
System.String	method	The method/function to be called.
System.Object	argument	The value to check.
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

Returns

TYPE	DESCRIPTION
T	Returns an object of Type T.

Type Parameters

NAME	DESCRIPTION
T	

CaptureScreen(String, Boolean, Int32)

Use this function to capture a screenshot of the game at the time of execution.

Declaration

```
public static bool CaptureScreen(string filename, bool remoteStorage = false, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	filename	The filename to use for the screenshot. Screenshots are saved to %TestFolder%\GDIOResults\resources\
System.Boolean	remoteStorage	If true, the screenshot will be stored on agent machine and not returned to the API; but the filename returned will be logged. For remote storage on the agent machine, the images will be saved in the root game folder. If a a mobile device is being tested, the resulting screenshot will be saved in the persisted data path. If false, the screenshot image is returned and saved with the given filename and the remote image on the agent machine will be deleted.
System.Int32	waitMaxSeconds	

Returns

TYPE	DESCRIPTION
System.Boolean	A png screenshot of the game at the time of execution. If a file already exists with the specified path and name, it will NOT be overwritten.

Checkpoint(Api.CheckpointStatus, String, Boolean)

Use this function to set a checkpoint in a test report.

Declaration

```
public static void Checkpoint(Api.CheckpointStatus status, string name = "", bool bForceScreenshot = false)
```

Parameters

TYPE	NAME	DESCRIPTION
Api.CheckpointStatus	status	A CheckpointStatus enum value.
System.String	name	The name of the checkpoint to be used in the report.
System.Boolean	bForceScreenshot	Take a screenshot reagrdless of the CheckpointStatus value.

Click(Api.MouseButtons, Int32, Int32, Int32, Int32)

Use this function to mouse click at a specified x,y coordinate.

Declaration

```
public static void Click(Api.MouseButtons button, int x, int y, int frames = 5, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
Api.MouseButtons	button	The name of the button See Api.MouseButtons.
System.Int32	x	The X coordinate to click at.
System.Int32	y	The Y coordinate to click at.
System.Int32	frames	The number of frames to hold the mouse button down for.
System.Int32	waitMaxSeconds	RESERVED, not currently used.

Click(Api.MouseButtons, Int32, Int32, UInt64, KeyCode[], KeyCode[], UInt64, UInt64, Int32, Int32)

Use this function to mouse click at a specified x,y coordinate.

Declaration

```
public static void Click(Api.MouseButtons button, int x, int y, ulong frames = 5UL, KeyCode[] keys = null,
KeyCode[] modifiers = null, ulong keyFrames = 5UL, ulong modKeyFrames = 3UL, int keyDelayMsec = 500, int
waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
Api.MouseButtons	button	The name of the button See Api.MouseButtons.
System.Int32	x	The X coordinate to click at.
System.Int32	y	The Y coordinate to click at.
System.UInt64	frames	
gdio.unity_api.KeyCode[]	keys	Array of keys to hold down prior to clicking.
gdio.unity_api.KeyCode[]	modifiers	Array of modifier keys to hold down prior to key press prior to mouse clicking.
System.UInt64	keyFrames	The number of frames to hold the keys down before clicking. Total press frame count is keyFrames + frames
System.UInt64	modKeyFrames	The number of frames to hold the modifier keys down before pressing the keys down. Total press frame count is modKeyFrames + keyFrames + frames
System.Int32	keyDelayMsec	Total time, in milliseconds, to wait between sending the key presses before sending the click event.
System.Int32	waitMaxSeconds	RESERVED, not currently used.

ClickObject(String, Api.MouseButtons, Int32, Int32)

Use this function to interact with an object in game using mouse-clicks.

Declaration

```
public static void ClickObject(string obj, Api.MouseButtons button, int numOfFrames = 5, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	The HierarchyPath to the object.
Api.MouseButtons	button	The name of the button See Api.MouseButtons.

TYPE	NAME	DESCRIPTION
System.Int32	numOfFrames	
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

ClickObjectEx(String, Api.MouseButtons, UInt64, KeyCode[], KeyCode[], UInt64, UInt64, Int32, Int32)

Use this function to interact with an object in game using mouse-clicks.

Declaration

```
public static void ClickObjectEx(string obj, Api.MouseButtons button, ulong numOfFrames = 5UL, KeyCode[] keys = null, KeyCode[] modifiers = null, ulong keyFrames = 5UL, ulong modKeyFrames = 3UL, int keyDelayMsec = 250, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	The HierarchyPath to the object.
Api.MouseButtons	button	The name of the button See Api.MouseButtons.
System.UInt64	numOfFrames	The number of frames to click the specific object.
gdio.unity_api.KeyCode[]	keys	Array of keys to hold down prior to clicking.
gdio.unity_api.KeyCode[]	modifiers	Array of modifier keys to hold down prior to key press prior to mouse clicking.
System.UInt64	keyFrames	The number of frames to hold the keys down before clicking. Total press frame count is keyFrames + frames
System.UInt64	modKeyFrames	The number of frames to hold the modifier keys down before pressing the keys down. Total press frame count is modKeyFrames + keyFrames + frames
System.Int32	keyDelayMsec	Total time, in milliseconds, to wait between sending the key presses before sending the click event.
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

Detach()

Use this function to detach from the attached/launched game.

Declaration

```
public static void Detach()
```

DisableKeybordHooks()

Use this function to disable keyboard hooks, to allow manual gameplay after the test is executed.

Declaration

```
public static void DisableKeybordHooks()
```

DisableMouseHooks()

Use this function to disable mouse hooks, to allow manual gameplay after the test is executed.

Declaration

```
public static void DisableMouseHooks()
```

DoubleClick(Api.MouseButtons, Int32, Int32, Int32, Int32)

Use this function to double-click a mouse button at a given x,y coordinate for a given number of frames.

Declaration

```
public static void DoubleClick(Api.MouseButtons button, int x, int y, int frames = 5, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
Api.MouseButtons	button	The name of the button See Api.MouseButtons.
System.Int32	x	The X coordinate to click at.
System.Int32	y	The Y coordinate to click at.
System.Int32	frames	The number of frames between clicks.
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

DoubleClickObject(String, Api.MouseButtons, Int32)

Use this function to interact with an object in game using mouse-clicks.

Declaration

```
public static void DoubleClickObject(string obj, Api.MouseButtons button, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	The HierarchyPath to the object.
Api.MouseButtons	button	The name of the button See Api.MouseButtons.
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

EnableKeybordHooks()

Use this function to enable keyboard hooks, in the event the option is not selected in the GDIO Agent settings in Unity at build time.

Declaration

```
public static void EnableKeybordHooks()
```

EnableMouseHooks()

Use this function to enable mouse hooks, in the event the option is not selected in the GDIO Agent settings in Unity at build time.

Declaration

```
public static void EnableMouseHooks()
```

GameAttach(Int32)

Use this function to have a Unity game running in the editor to attach to the API instance, instead of using Launch(String, String, Int32) to start the standalone game from an executable.

Declaration

```
public static bool GameAttach(int waitSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	waitSeconds	The amount of time in seconds to wait for the Unity game to attach to the API instance. DEFAULT = 60 seconds.

Returns

TYPE	DESCRIPTION
System.Boolean	Returns true if the game successfully attached in the waitSeconds period to the API instance.

GetActiveSceneName(Int32)

Use this function to get the name of the SceneManager.GetActiveScene()

Declaration

```
public static string GetActiveSceneName(int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

Returns

TYPE	DESCRIPTION
System.String	the value of the object field.

GetLastFPS()

Use this function to query the most recent FPS poll performed by the GDIO Agent. The poll rate is defined in Unity IDE under the empty GameObject where you attached the GDIO Agent (Script) > FPS Event Publish.

Declaration

```
public static double GetLastFPS()
```

Returns

TYPE	DESCRIPTION
System.Double	Returns a double representing the last polled FPS.

GetLastFPSUpdate()

Use this function to retrieve the DateTime of the last FPS poll (see: GetLastFPS())

Declaration

```
public static DateTime GetLastFPSUpdate()
```

Returns

TYPE	DESCRIPTION
System.DateTime	Returns a DateTime object representing the most recent FPS poll.

GetObjectFieldValue(String, Int32)

Use this function to query an object field value.

Declaration

```
public static string GetObjectFieldValue(string hierarchypath, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	hierarchypath	The HierarchyPath to the object.
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

Returns

TYPE	DESCRIPTION
System.String	the value of the object field.

GetObjectFieldValue<T>(String, Int32)

Use this function to query an object field value.

Declaration

```
public static T GetObjectFieldValue<T>(string hierarchypath, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	hierarchypath	The HierarchyPath to the object.
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

Returns

TYPE	DESCRIPTION
T	the value of the object field.

Type Parameters

NAME	DESCRIPTION
T	

GetObjectList(Api.ObjectListFilter, Int32)

Use this function to retrieve the current object list.

Declaration

```
public static string[] GetObjectList(Api.ObjectListFilter filter = Api.ObjectListFilter.ALL, int waitSeconds =
60)
```

Parameters

TYPE	NAME	DESCRIPTION
gdio.unity_api.Api.ObjectListFilter	filter	
System.Int32	waitSeconds	

Returns

TYPE	DESCRIPTION
System.String[]	Returns a string[] of all objects in the currently active scene.

GetObjectPosition(String, Api.CoordinateConversion, String, Int32)

Use this function to query an object position.

Declaration

```
public static string GetObjectPosition(string obj, Api.CoordinateConversion cordSpace =
Api.CoordinateConversion.None, string cameraObj = null, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	The HierarchyPath to the object.
gdio.unity_api.Api.CoordinateConversion	cordSpace	The coordinate space for the position vector to be returned in. See gdio.unity_api.Api.CoordinateConversion
System.String	cameraObj	
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

Returns

TYPE	DESCRIPTION
System.String	the position of the object.

Launch(String, String, Int32)

Use this function to launch the Unity game build from an executable. In order to automate execution, games should be set to automatically start at the desired screen resolution and require no additional external dialogs to start. These can typically be set in Unity under Build Settings > Player Settings > Resolution.

Declaration

```
public static bool Launch(string filename, string arguments = "", int waitSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	filename	The path to the executable. In Windows, this should follow the standard "C:\folder\subfolder\filename.exe" convention. In macOS, this should follow the full path to the executable binary contained within the .app container, i.e. "/Users/Applications/UnityProject.app/Contents/MacOS/UnityProject".
System.String	arguments	
System.Int32	waitSeconds	int representing the number of seconds to wait before returning false. Default = 60 seconds.

Returns

TYPE	DESCRIPTION
System.Boolean	Returns true if the API successfully launches the game and syncs with the running game.

LoadResolver(String, Int32, Int32)

Use this function to load a custom object resolver class.

Declaration

```
public static bool LoadResolver(string className, int priority = 0, int waitSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	className	
System.Int32	priority	
System.Int32	waitSeconds	

Returns

TYPE	DESCRIPTION
System.Boolean	

LoadScene(String, Int32)

Use this function load a specified scene.

Declaration

```
public static void LoadScene(string sceneName, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	sceneName	The name of the scene to load.

TYPE	NAME	DESCRIPTION
System.Int32	waitMaxSeconds	The amount of time to wait for the SceneManager to signal the scene has been loaded.

MouseDrag(Api.MouseButtons, Single, Single, Single, Single, UInt64, Int32)

Use this function to click-and-drag with the mouse.

Declaration

```
public static void MouseDrag(Api.MouseButtons button, float x1, float y1, float x2, float y2, ulong frames,
int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
Api.MouseButtons	button	Which mouse button to drag with.
System.Single	x1	The initial X coordinate, where you will begin to click-and-drag.
System.Single	y1	The initial Y coordinate, where you will begin to click-and-drag.
System.Single	x2	The end X coordinate, where you will stop dragging and release the mouse button.
System.Single	y2	The end Y coordinate, where you will stop dragging and release the moust button.
System.UInt64	frames	The number of frames to perform the drag.
System.Int32	waitMaxSeconds	The timeout for the action to be performed.

Move(String, Single, UInt64)

Use this function to send move commands to the game, as defined by the input settings in the Input Manager (under Edit > Project Settings > Input Manager).

Declaration

```
public static void Move(string axisName, float val, ulong numOfFrames)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	axisName	Refers to the Horizontal1, Horizontal2, Vertical1 or Vertical2 Input Axes, as defined in the Input Manager.

TYPE	NAME	DESCRIPTION
System.Single	val	between -1f and 1f (for normal input). Negative numbers refer to the opposite direction (i.e. if Horizontal1 turns to the right, a negative number will turn to the left). **WARNING:** Numbers greater than 1 may result in erratic behavior.
System.UInt64	numOfFrames	Refers to the number of frames to move/turn in the specified direction.

MoveMouse(Single, Single, Single, Single, UInt64, Int32)

Use this function to move the mouse from a starting point to and ending point

Declaration

```
public static void MoveMouse(float x1, float y1, float x2, float y2, ulong frames, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	x1	The initial X coordinate, where you will begin to click-and-drag.
System.Single	y1	The initial Y coordinate, where you will begin to click-and-drag.
System.Single	x2	The end X coordinate, where you will stop dragging and release the mouse button.
System.Single	y2	The end Y coordinate, where you will stop dragging and release the moust button.
System.UInt64	frames	The number of frames to perform the drag.
System.Int32	waitMaxSeconds	The timeout for the action to be performed.

MoveMouseToObject(String, UInt64, Int32)

Use this function to move the mouse from the current position to a specific object.

Declaration

```
public static void MoveMouseToObject(string hierarchypath, ulong frames, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	hierarchypath	HierarchyPath to resolve object

TYPE	NAME	DESCRIPTION
System.UInt64	frames	The number of frames to perform the drag.
System.Int32	waitMaxSeconds	The timeout for the action to be performed.

MoveToPoint(String, Vector3, Boolean, Int32)

Use this function to move a game object to a specific gdio.unity_api.Vector3 coordinate in the game. A NavMesh is REQUIRED (https://docs.unity3d.com/Manual/nav-BuildingNavMesh.html). **WARNING**: Function is not limited to moving player-controlled objects.

Declaration

```
public static void MoveToPoint(string obj, Vector3 point, bool isAsync = false, int waitSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	
gdio.unity_api.Vector3	point	A gdio.unity_api.Vector3 point to move the object to.
System.Boolean	isAsync	Whether to issue to the command as sync or async. **WARNING**: Unintended results may occur when additional move commands are issued before the object has reached the desired position.
System.Int32	waitSeconds	The number of seconds to wait for the event to complete before timing out. Default = 60 seconds.

MoveToPoint(String, String, String, String, Boolean, Int32)

Use this function to move a game object to a specific X, Y, Z coordinate in the game. Works best when a Nav Mesh is enabled in the game (https://docs.unity3d.com/Manual/nav-BuildingNavMesh.html). **WARNING**: Function is not limited to moving player-controlled objects. A NavcMesh is REQUIRED.

Declaration

```
public static void MoveToPoint(string obj, string x, string y, string z, bool isAsync = false, int waitSeconds
= 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	
System.String	x	The X coordinate to move the object to.

TYPE	NAME	DESCRIPTION
System.String	y	The Y coordinate to move the object to.
System.String	z	The Z coordinate to move the object to.
System.Boolean	isAsync	Whether to issue to the command as sync or async. **WARNING**: Unintended results may occur when additional move commands are issued before the object has reached the desired position.
System.Int32	waitSeconds	The number of seconds to wait for the event to complete before timing out. Default = 60 seconds.

PauseGame(String, Int32)

Use this function to Pause the running game if UnityAutoPlay plugin is running.

Declaration

```
public static void PauseGame(string host, int port = 110002)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	host	The hostname the game is running on.
System.Int32	port	The port the UnityAutoPlay plugin is listening on.

PressButton(String, UInt64)

Use this function to send arbitrary button states to the game.

Declaration

```
public static void PressButton(string buttonName, ulong numOfFrames)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	buttonName	The name of the button to be sent.
System.UInt64	numOfFrames	The number of frames to press the button specified.

PressKey(KeyCode, UInt64)

Use this function to send arbitrary button states to the game.

Declaration

```
public static void PressKey(KeyCode key, ulong numOfFrames)
```

Parameters

TYPE	NAME	DESCRIPTION
gdio.unity_api.KeyCode	key	The key to press. Use this overload if you need to differetiate between LEFT/RIGHT [ALT,CONTROL,SHIFT,WINDOWS_KEY]
System.UInt64	numOfFrames	The number of frames to press the key sent.

PressKey(KeyCode[], UInt64, KeyCode[], UInt64, Int32)

Use this function to send arbitrary button states to the game.

Declaration

```
public static void PressKey(KeyCode[] keys, ulong numOfFrames, KeyCode[] modifiers = null, ulong
modDelayFrames = 3UL, int keyDelayMsec = 500)
```

Parameters

TYPE	NAME	DESCRIPTION
gdio.unity_api.KeyCode[]	keys	The keys to press. Use this overload if you need to differentiate between LEFT/RIGHT [ALT,CONTROL,SHIFT,WINDOWS_KEY]
System.UInt64	numOfFrames	The number of frames to press the key sent.
gdio.unity_api.KeyCode[]	modifiers	Array of modifier keys to hold down prior to key press.
System.UInt64	modDelayFrames	
System.Int32	keyDelayMsec	Total time, in milliseconds, to wait between sending the modifiers and key presses.

PressKey(String, UInt64)

Use this function to send arbitrary button states to the game. Defaults to LEFT ALT/CTRL/SHIFT/WINDOWS(COMMAND)

Declaration

```
public static void PressKey(string keys, ulong numOfFrames)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	keys	The keys to press using the format of UnityEngine.Event.KeyboardEvent function: (https://docs.unity3d.com/ScriptReference/Event.KeyboardEvent.html)

TYPE	NAME	DESCRIPTION
System.UInt64	numOfFrames	The number of frames to press the key sent.

Quit()

Use this function to terminate the currently playing game instance that was started with Launch, WaitForGame, or GameAttach. Terminates the active API connection.

Declaration

```
public static void Quit()
```

Rotate(String, String, String, String, String)

Use this function to rotate an object by x, y and z coordinates. Each coordinate parameter refers to the positive or negative rotation along that axis.

Declaration

```
public static void Rotate(string obj, string Q_x, string Q_y, string Q_z, string Q_w)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	
System.String	Q_x	
System.String	Q_y	
System.String	Q_z	
System.String	Q_w	

SetInputFieldText(String, String, Int32)

Use this function to set the text value of a UI.InputField. This function completes the updates in a single frame and returns when complete.

Declaration

```
public static bool SetInputFieldText(string obj, string value, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	The HierarchyPath to the object.
System.String	value	The value to check.

TYPE	NAME	DESCRIPTION
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

Returns

TYPE	DESCRIPTION
System.Boolean	True if the field value is found, false if it is not found before waitMaxSeconds.

SetObjectFieldValue(String, String, String, Int32)

Use this function to set the text value of a UI.InputField. This function completes the updates in a single frame and returns when complete.

Declaration

```
public static bool SetObjectFieldValue(string obj, string field, string value, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	The HierarchyPath to the object.
System.String	field	
System.String	value	The value to check.
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

Returns

TYPE	DESCRIPTION
System.Boolean	True if the field value is found, false if it is not found before waitMaxSeconds.

SetTimescale(Single)

Use this function to set the play speed of the game. Can be used to speed up or slow down the game in order to improve test execution time. (https://docs.unity3d.com/ScriptReference/Time-timeScale.html)

Declaration

```
public static void SetTimescale(float val)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	val	The coefficient of time.

Wait(Int32)

Use this function to wait a certain amount of time before moving on to the next action in the test. Does not interrupt asynchronous actions.

Declaration

```
public static void Wait(int milliSeconds)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	milliSeconds	The amount of time to wait in milliseconds

WaitForEmptyInput(Int32)

Use this function to wait for a previous input command to complete before continuing on to the next action.

Declaration

```
public static bool WaitForEmptyInput(int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

Returns

TYPE	DESCRIPTION
System.Boolean	True when the input handler is empty, false if it is not in the timeout period.

WaitForGame(String, Int32, Int32)

Use this function to have have the API instance start the game running in the IDE and attach to the API instance, instead of using Launch(String, String, Int32) to start the standalone game from an executable. To run a test on another machine running the Unity IDE and the GDIO agent, specify the hostname or IP address as the host parameter.

Declaration

```
public static bool WaitForGame(string host = null, int port = 11000, int waitSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	host	The IP or hostname of the Unity IDE under test. Must have the GDIO agent installed and AutoPlay.dll loaded. DEFAULT = localhost.
System.Int32	port	The port number used by the agent to connect. DEFAULT = 11000.
System.Int32	waitSeconds	The amount of time in seconds to wait for the Unity game to attach to the API instance. DEFAULT = 60 seconds.

Returns

TYPE	DESCRIPTION
System.Boolean	Returns true if the game successfully attached in the waitSeconds period to the API instance.

WaitForObject(String, Int32)

Use this function to check if an object exists.

Declaration

```
public static bool WaitForObject(string obj, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	The HierarchyPath to the object.
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

Returns

TYPE	DESCRIPTION
System.Boolean	True if the object is found, false if it is not found.

WaitForObjectValue(String, String, String, Int32)

Use this function to check for an object value.

Declaration

```
public static bool WaitForObjectValue(string obj, string field, string value, int waitMaxSeconds = 60)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	The HierarchyPath to the object.
System.String	field	The field name.
System.String	value	The value to check.
System.Int32	waitMaxSeconds	The amount of time to wait in milliseconds.

Returns

TYPE	DESCRIPTION
System.Boolean	True if the field value is found, false if it is not found before waitMaxSeconds.

Events

DeserializeObjectCallback

Declaration

```
public static event Api.DeserializeObjectHandler DeserializeObjectCallback
```

Event Type

TYPE	DESCRIPTION
gdio.unity_api.Api.DeserializeObjectHandler	

SerializeObjectCallback

Declaration

```
public static event Api.SerializeObjectHandler SerializeObjectCallback
```

Event Type

TYPE	DESCRIPTION
gdio.unity_api.Api.SerializeObjectHandler	

Enum Api.CheckpointStatus

Namespace: gdio.unity_api

Assembly: cs.temp.dll.dll

Syntax

```
public enum CheckpointStatus
```

Fields

NAME	DESCRIPTION
FAIL	Fail
PASS	Pass
SUSPECT	Suspect
UNKNOWN	Unknown

Enum Api.MouseButtons

Mouse Button choices

Namespace: gdio.unity_api

Assembly: cs.temp.dll.dll

Syntax

```
public enum MouseButtons
```

Fields

NAME	DESCRIPTION
LEFT	Left Mouse Button
MIDDLE	Middle Mouse Button
RIGHT	Right Mouse Button