

Scorm API

[Supported Platforms](#)

[Features](#)

[Getting started](#)

[Upgrade guide](#)

[Code changes](#)

[Template required changes](#)

[Scorm Build](#)

[Scorm Package Test](#)

[Scorm Cloud](#)

[Scorm 1.2 Player](#)

[API Reference](#)

[General](#)

[Generic getters and setters](#)

[LearnerId](#)

[LearnerName](#)

[LessonLocation](#)

[Credit](#)

[LessonStatus](#)

[Entry](#)

[RawScore](#)

[MaxScore](#)

[MinScore](#)

[TotalTime](#)

[LessonMode](#)

[Exit](#)

[SessionTime](#)

[SuspendData](#)

[Comments](#)

[CommentsFromLMS](#)

[Objectives](#)

[Language](#)

[FAQ](#)

Supported Platforms

Scorm packages run on LMS that run on Web Browsers.

- **WebGL** support.
- **WebPlayer** support **has been removed**. The last Unity compatible version was 5.3. If you still need it I can share you the last 1.x version of the plugin.

Features

- [Scorm 1.2](#) and [Scorm 2004](#) support. Implemented 1.2/2004 equivalent API, If you need something specific of Scorm 2004 that is not available in 1.2, feel free to ask and you will have it as soon as possible.
- **Builds Scorm packages automatically.** You get a “*.zip” with your Scorm package ready to upload to a LMS.
- **Full object oriented API** (you don't deserialize anything, it's already done it for you).
- **C# Source code** included.
- **Documentation** available.
- **Easy integration** to your Unity project.
- Well **organized and structured** code.
- Provides a **in editor implementation** for testing purposes.
- **Examples** available. Includes an example to test the full API with a console view.

Getting started

```
IScornService scormService;

#if UNITY_EDITOR
    scormService = new ScormPlayerPrefsService(); // PlayerPrefs implementation (for
editor testing)
#else
    scormService = new ScormService(); // Real implementation
#endif

bool result = scormService.Initialize(Scorm.Version.Scorm_1_2); // Begins
communication with the LMS

if (result) {
    scormService.SetMinScore(0); // Sets a min score of 0
    scormService.SetMaxScore(10); // Sets a max score of 10
    scormService.SetRawScore(6.5f); // Sets a score of 6.5

    scormService.Commit(); // Commit the pending changes

    scormService.Finish(); // Ends communication with the LMS
}
```

Upgrade guide

A heavy code rewritten happened between 1.x and 2.x versions so you need to follow this guide if you want to upgrade an existing project.

Before updating remove “Assets/ScormAPI” folder and then import the new package. I have removed some files between 1.x and 2.x so if you don’t do this you will mix old and new version together creating many errors.

Example code has also been upgraded so maybe you can see there the required changes.

Code changes

- *ScormAPI* is now *ScormService*.
- *ScormService* is no longer a *MonoBehaviour* so you don't need to drag it as a component.
- *ScormService* is no longer following Singleton pattern,, it's up to the developer if he wants to use it that way.

For allowing an easier upgrade you can copy this code in a new script file. This way you can used it as a Singleton like before.

```
public class Singleton<T> where T : class, new() {  
  
    private static T instance = null;  
  
    public static T Instance {  
        get {  
            if (instance == null) {  
                instance = new T();  
            }  
  
            return instance;  
        }  
    }  
}  
  
public class ScormAPI : Singleton<ScormService> { }
```

- *ScormService* implements an interface to make it easier to fake it in editor and be able to test it there.
- To establish a new connection with the LMS you need to call:

```
scormService.Initialize(Scorm.Version.Scorm_1_2); // Scorm 1.2
scormService.Initialize(Scorm.Version.Scorm_2004); // Scorm 2004
```

- `IsInitialized` is now `IsConnected`.
- Enums are no longer inside *ScormAPI* class.
- All events have been removed. "Get*" methods are no longer "void" and return directly the value that was send previously in the event.
- *GetCommentsFromLMS* becomes *GetCommentsFromLms*.
- *Exit* becomes *SetExitReason*.
- *OnUserClosed* event has been removed because there is no way to ensure that when users closes a browser tab or the entire browser it will send that pending request. You need to save your data on regular intervals or have checkpoints like in games to ensure that user doesn't lost any data.

Template required changes

You need to remove this block in *index.html* file because it no longer requires template customization (that's why I don't provide any custom templates anymore):

```
<script type="text/javascript" src="js/ostynscormtime.js"></script>
<script type="text/javascript" src="js/scorm.js"></script>
<script type="text/javascript">
    var processedUnload = false;

    var scorm = new Scorm("Scorm %UNITY_CUSTOM_SCORM_VERSION%");

    window.onbeforeunload = function(e) {
        doUnload();
    };

    window.onunload = function(e) {
        doUnload();
    };

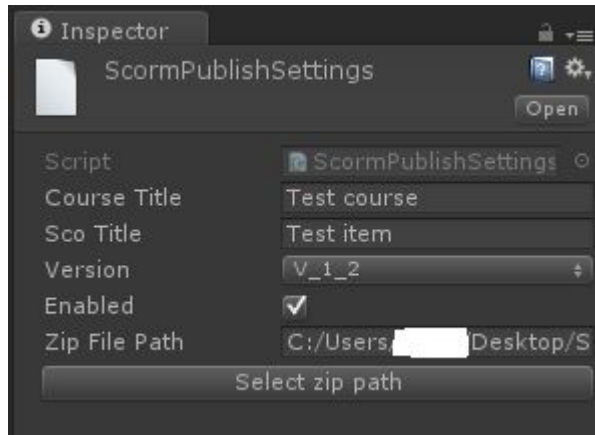
    function doUnload() {
        if (processedUnload) return;

        processedUnload = true;

        SendMessage("ScormAPI", "Close");
    }
</script>
```

Scorm Build

It can build a “*.zip” file ready to upload to a LMS:



1. Insert your Course title in “Resources > ScormPublishSettings.asset > Course Title” field.
2. Insert your Sco title in “Resources > ScormPublishSettings.asset > Sco Title” field.
3. Select your Scorm version in “Resources > ScormPublishSettings.asset > Version” field.
4. Enable publish in “Resources > ScormPublishSettings.asset > Enabled” field.
5. Insert your “*.zip” file path in “Resources > ScormPublishSettings.asset > Zip file path” field (or click the button and select the path).

Every time you build to WebGL/WebPlayer platform, it will generate a “*.zip” file in the desired path.

Scorm Package Test

If you don't have or can't have access to a LMS you can test it with the following applications:

Scorm Cloud

Supports 1.2/2004:

1. You can create a free account and upload a package if it doesn't exceed 100MB:

<http://scorm.com/scorm-solved/scorm-cloud-features/content-just-works/testing-content-with-scorm-cloud/>

2. Upload your Scorm package and test it!

Scorm 1.2 Player

Supports only 1.2.

1. Download it:

<http://www.reload.ac.uk/scormplayer.html>

2. Run "reload-scorm-player.jar".
3. Open your Scorm 1.2 package and name it.
4. Press "Play" to test it!

Note: It's a really old software that has some issues like not allowing to have double quotes on string values. For example, you can encounter issues testing suspend data if you store JSON data.

API Reference

Implemented all the equivalent 1.2/2004 API:

<http://scorm.com/scorm-explained/technical-scorm/run-time/run-time-reference/>

General

Properties
public string IsConnected { get; } public Version Version { get; }
Methods
public void Initialize (Version version); public void Commit (); public void Finish ();

Generic getters and setters

Methods
public string GetStringValue (string element); bool SetValue (string element, string value); int GetIntValue (string element); bool SetValue (string element, int value); float GetFloatValue (string element); bool SetValue (string element, float value);

LearnerId

Methods
public string GetLearnerId ();

LearnerName

Methods
public string GetLearnerName ();

LessonLocation

Methods

```
public string GetLessonLocation();  
public bool SetLessonLocation(string value);
```

Credit

Methods

```
public CreditType GetCredit();
```

LessonStatus

Methods

```
public LessonStatus GetLessonStatus();  
public bool SetLessonStatus(LessonStatus value);
```

Entry

Methods

```
public EntryType GetEntry();
```

RawScore

Methods

```
public float GetRawScore();  
public bool SetRawScore(float value);
```

MaxScore

Methods

```
public float GetMaxScore();  
public bool SetMaxScore(float value);
```

MinScore

Methods

```
public float GetMinScore();  
public bool SetMinScore(float value);
```

TotalTime

Methods

```
public int GetTotalTime();
```

LessonMode

Methods

```
public LessonMode GetLessonMode();
```

Exit

Methods

```
public bool SetExitReason(ExitReason value);
```

SessionTime

Methods

```
public bool SetSessionTime(int milliseconds);
```

SuspendData

Methods

```
public string GetSuspendData();  
public bool SetSuspendData(string value);
```

Comments

Methods

```
public string GetComments();
```

CommentsFromLMS

Methods
public string GetCommentsFromLms ();

Objectives

Methods
public List<ObjectiveData> GetObjectives (); public void SetObjective (ObjectiveData value);

Language

Methods
public string GetLanguage ();

FAQ

- 1. I get this message when uploading a Scorm to Docebo LMS: "Some of the archive files are not allowed". What can I do?**

Unity generates a ".htaccess" file that it's not a valid file for Docebo. You must remove it from the output Scorm zip file.