Unity-SCORM® Quick Start Guide

Follow the steps below to get Unity-SCORM up and running in no time at all. By following these steps, your Unity content will be SCORM-conformant and will *inform* the LMS that your course is complete. This tutorial assumes you have basic Unity authoring skills. This guide details a simple use case for SCORM-conformant Unity content. After completing this guide, you can begin using additional SCORM features to track objectives and interactions through the ScormManager interfaces.

1. Download the ScormIntegrationKit.unitypackage from www.adlnet.gov
2. Import the package into your Unity project by selecting ***Assets->Import Asset Package->Custom Package***. Choose the file you downloaded from www.adlnet.gov in step 1. Select all the files and click ***Import***.
3. You will see a new menu item called ***SCORM***. If you do not, click on the main menu bar near ***File*** to force it to refresh. From the ***SCORM*** menu, choose ***Create SCORM Manager***.
4. Decide the condition under which you wish to mark your course as complete. Add code to your scripts that executes when this condition is met. Use this code:
   ScormManager.SetCompletionStatus(completionStatusType.complete);
   ScormManager.SetSuccessStatus(successStatusType.sucess);
   ScormManager.Commit();
5. Locate your simulation's shutdown or exit procedure. Add logic to call this when you receive the message ***Scorm_Commit_Complete***. Make sure that the logic that controls this is attached to a child gameobject of the ScormManager so that it will receive the message. Call ***ScormManager.Terminate()*** as the last function in the shutdown procedure.
6. In your game startup logic, make sure the simulation does not begin or pauses until you receive the ***Scorm_Initialization_Complete*** message. Make sure the logic that controls this is attached to a child gameobject of the ScormManager so that it will receive the message.
7. Choose ***File->Build Settings*** from the main menu. In the dialog that appears, look under ***Platform*** and choose ***Web Player***. Click the button called ***Player Settings***.
8. In the inspector, choose the WebPlayer Template called ***SCORM***.
9. In the ***Build Settings*** dialog, choose ***Build***. Pick a location and export the WebPlayer.
10. Close the ***Build Settings*** dialog.
11. Choose ***SCORM->Export SCORM Package*** from the main menu. Read the warning, then click *ok*.
12. In the ***ScormPackager*** dialog, click the button titled ***Player Location*** on the top right. Choose the directory where you published the WebPlayer in step 9. Note: this directory should contain the ***WebPlayer*** folder.
13. Give your course and your scene a title in the text boxes on the dialog (Note: Select the SCORM version supported by your LMS. For additional information see www.adlnet.gov).
14. Click ***Publish*** at the bottom. Choose a filename and a location.

15. Upload this course to your LMS. This step differs based on the LMS you are using (See http://www.adlnet.gov/capabilities/scorm#tab-learn for a list of certified products and SCORM Adopters). When you launch it, you will see your Unity simulation. Play through and satisfy your success criteria. Exit your simulation – you should see the LMS mark the course as passed.

That's it! Your Unity Simulation should now report its completion and success status to the LMS. From here, you can begin tracking interactions, objectives and other SCORM data model elements. You can also access data about the student, such as name or playback preferences. You can even store data in the LMS to resume your simulation from a given state, or check other LMS features like global objectives.

For information on SCORM and SCORM conformance, see http://www.adlnet.gov/capabilities/scorm. For information on a specific version of SCORM, select the version at http://www.adlnet.gov/capabilities/scorm#tab-projects.

FAQ

What is SCORM, and why do I need it?

SCORM is the de facto e-learning standard for packaging content, communication between content and an LMS, and content sequencing. SCORM enables interoperability between e-learning content and learning management systems (LMS).

In a nutshell, what does the Unity SCORM Integration Kit do?

The Unity SCORM Integration Kit is a Research and Development project funded by the US. Department of Defense Advanced Distributed Learning Initiative. It allows simulations, built in Unity and hosted by the Unity WebPlayer, to be imported, launched, and tracked in an LMS.

How do I get the Unity SCORM Integration Kit?

Download the file from the Unity Asset Store or from the ADL website at www.adlnet.gov. You should receive a file called UnityScormIntegration.unitypackage.

How do I install the Unity SCORM Integration Kit?

The Unity SCORM Integration Kit will need to be imported into each of your Unity projects. In the main Unity GUI, choose ***Assets*** on the top menu, then select ***Import Package***, and click ***Custom Package***. Browse to and select the ***UnityScormIntegration.unitypackage*** file. In the dialog that appears, make sure all the check boxes are checked, then click Import. This will copy the necessary files into a folder called ADL in your assets folder.

How do I get access to the SCORM data within my Unity scene?

The Unity SCORM Integration Kit contains code that will add some menu options to your main menu. If you have just imported the package and do not see ***SCORM*** on your main menu, try clicking the ***File*** menu to force the menu bar to refresh. From this menu, select ***Create SCORM Manager***. You should see a message that a ScormManager prefab has been added to your scene. In your scripts, you can type ***ScormManager.GetLearnerName()*** or other functions when the ScormManager has finished its initialization process.

When should I set the Completion Status as Complete, or the Success Status as Successful?

SCORM does not specify how or when you decide that your training has been successfully completed. It's up to you and your instructional designers to determine when students have passed, what their score is, and what interactions are tracked by the system.

How can I see what functions are available in the ScormManager?

You should be able to see a list of the available functions by typing ScormManager followed by a period in the MonoDevelop GUI. These functions have descriptive names and metadata describing their function. You can view the source of the ScormManager.cs file for more information on how these functions work, and what they do.

I need to manipulate the data model in a way that is not exposed by the ScormManager. Is there a way to access the student data directly?

Yes. Call ScormManager. GetLearnerRecord(). This will return a data object that has all of the SCORM data model elements available. Be careful – you'll need to know the meaning of the SCORM terminology to make good use of this.

How do I know what all these terms mean? I see things like *Entry*, *TimeLimitAction*… Are these documented somewhere?

Of course. You can read the official SCORM Documentation at http://www.adlnet.gov/capabilities/scorm#tab-learn

How do I know when the ScormManager's initialization is complete, so I can begin communicating with it?

The ScormManager Gameobject will send a message to its children when it is ready to receive commands. This message is ***Scorm_Initialize_Complete***. The ScormManager will begin its startup procedure automatically. Be sure to place objects that need to access the ScormManager under it in the scene hierarchy, so that they may receive the startup message. Typical startup times in our testing range from 1 to 3 seconds.

I've published my game to an EXE file – how does it know what LMS to connect to?

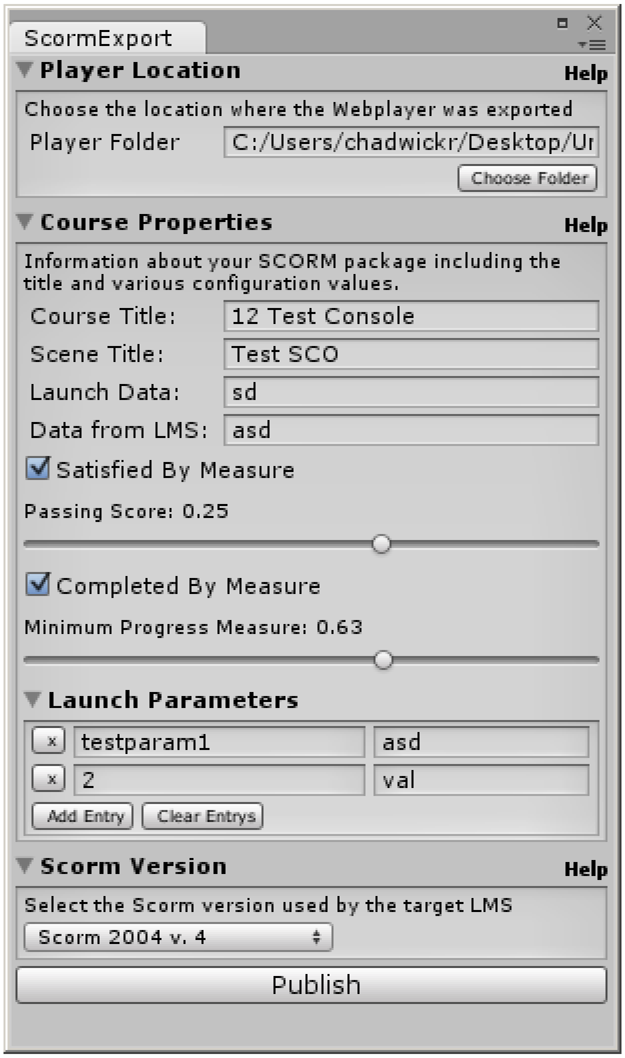
SCORM simulations must be hosted by the Unity WebPlayer and launched from the LMS to track SCORM data. There is currently no standardized way to connect a desktop executable file to a SCORM LMS. Contact ADL for advice if you cannot use the Unity WebPlayer. This integration kit is intended to work with the Unity WebPlayer only.

OK, I published as a WebPlayer – now what?

First, make sure that you selected that SCORM WebPlayer template. The SCORM system depends on some JavaScript libraries that will not be available on the standard WebPlayer template. Once you have published as a WebPlayer, note the location of the files. These should be in a folder called ***WebPlayer*** within the output folder you selected when publishing. This folder will need to be packaged in a SCORM Content Package before it can be uploaded to your LMS. We have provided a tool to create a basic content package from within the Unity GUI.

How do I use the content packaging tool?

You should have a menu item on your main menu titled ***SCORM***. Select this menu, and then select ***Export SCORM Package***. You should see a dialog like the one shown here. The tool provides a lot of feedback on the various options, and you can get an explanation by holding your mouse over a term. Be sure to select the folder where you published the WebPlayer. When you have filled out the data, choose publish at the bottom. This will export a .zip file with the proper format to upload to a SCORM LMS.



Player Location

Choose the location where the Webplayer was exported

Player Folder: C:/Users/chadwickr/Desktop/Ur

Course Properties

Information about your SCORM package including the title and various configuration values.

Course Title: 12 Test Console

Scene Title: Test SCO

Launch Data: sd

Data from LMS: asd

☑ Satisfied By Measure

Passing Score: 0.25

☑ Completed By Measure

Minimum Progress Measure: 0.63

Launch Parameters

x	testparam1	asd
x	2	val

Scorm Version

Select the Scorm version used by the target LMS

Scorm 2004 v. 4

I'm fairly good with SCORM, and would like to set up more complex sequencing rules. Can I do this through the GUI?

There is currently no support for other sequencing rules in the GUI. Instead, open the content package that is exported and locate the imsmanifest.xml. You can add your sequencing rules or other packaging options here. The ADL RELOAD Editor Version 1.1 can be used to edit a SCORM package. RELOAD is available at http://www.adlnet.gov/capabilities/scorm#tab-learn

I've published to a WebPlayer, but nothing seems to be happening. What's wrong?

First, check that you exported to the SCORM WebPlayer template. This template includes JavaScript libraries that are necessary for the SCORM/Unity integration to work. Second, ensure that the SCORM package was imported into an LMS. The SCORM/Unity integration will not work when launching the course off of your local computer with your browser – it must be delivered via the LMS. For basic testing, you can use the ScormSimulator JavaScript file. This will simulate the LMS run-time environment although you will have no control over the 'fake' data that the simulator provides, nor will the simulator actually store data you submit. To use the simulator, uncomment the appropriate line at the bottom of ScormSimulator.js in the /assets/webplayertemplates/scorm/scripts/ folder. Publish your WebPlayer and run locally. Remember to re-comment this line before you publish your final course, or the integration kit will send data to the simulator instead of the real LMS.

I need to run the WebPlayer locally in order to troubleshoot my code. However, my simulation never starts because I never receive the *Scorm_Initialization_Complete* message.

For basic testing, you can use the ScormSimulator JavaScript file. This will simulate the LMS run-time environment although you will have no control over the 'fake' data that the simulator provides, nor will the simulator actually store data you submit. To use the simulator, uncomment the appropriate line at the bottom of ScormSimulator.js in the /assets/webplayertemplates/scorm/scripts/ folder. Publish your WebPlayer and run locally. Remember to re-comment this line before you publish your final course, or the integration kit will send data to the simulator instead of the real LMS.

I have a custom WebPlayer template with slick graphics, and I don't want to use yours. Can I do this?

Sure. Just be sure to include scorm.js in your webplayer.html file. This file can be found in /assets/webplayertemplates/scorm/scripts.

When I close the simulation, no data is saved. How do I make sure that the data I send to the ScormManager is saved to the LMS?

You need to call ScormManager.Commit(); before closing the simulation. This process takes a few seconds to complete, during which we suggest you show some sort of GUI page. When the Unity SCORM Integration Kit has saved all the data to the LMS, it will send the message ***Scorm_Commit_Complete***. Be sure not to close the Unity WebPlayer until this process has finished. If you think the user might close the window without exiting your simulation correctly, you can periodically call ScormManager.Commit() while your simulation is running. It launches a separate thread and will not pause or block your game.