



Tips and Tricks: Secrets to Adobe Captivate & SCORM-Compliant Learning Management System Integration

Brian Kleeman
Technology Director and
Lead Software Developer

ICS Learning Group
8221 Ritchie Highway
Pasadena, MD 21122

410-975-9440
www.icslearninggroup.com

Overview

Until recently, anyone who wanted to author SCORM-compliant content had few choices. Not many authoring programs existed and the technical knowledge to create compliant content was and, in fact, still is beyond the reach of most training developers. Now there are many affordable, easy to use content authoring programs to create SCORM-compliant content that can be deployed to learning management systems (LMS). Adobe, a leader in the multimedia authoring and programming industry, has thrown their hat into the ring with Adobe Captivate – a SCORM-compliant authoring tool that includes screen capture, simulation, automated testing and more.

ICS Learning Group was a beta-test facility for Adobe Captivate and has an ongoing dialogue with Adobe representatives discussing how to improve Captivate and its LMS integration with relation to SCORM-compliance. Our Inquisiq Learning Management System is also featured in *Essentials of Adobe Captivate* from IconLogic and underwent extensive testing with Captivate lessons in preparation for the book's release.

Adobe Captivate and LMS Integration

When exporting content from Adobe Captivate, you have the option of making your package SCORM 1.2 Compliant. Specifics of the SCORM specification could fill an entire book (in fact, it does!), so let's just say that SCORM defines what must be included in a content package (certain files which contain certain information in a certain format) and the methods that the content package must use to communicate information (student name, score, etc.) to and from the LMS. The idea is that content authoring programs and learning management systems would all be programmed to comply with the spec and therefore be compatible with each other. Unfortunately, reality has not lived up to the vision.

The SCORM 1.2 specification is long, open to some interpretation and not always logical. Developers have had to make some assumptions and, at the same time, had to predict and hope that other developers made the same assumptions. Our experience with the workings of Adobe Captivate and the development of our own SCORM-compliant LMS has given us some insight to help you get the most out of Adobe Captivate and your LMS – even if it's not our LMS! Note that we are only focusing on SCORM 1.2. The SCORM 1.3 specification is the latest version, in various revisions; however most learning management systems and authoring tools, even those recently released, may still only support SCORM 1.2.

Making Sure Your Captivate Lesson is Compliant

Problem #1: I'll get right into it here. If you export a Captivate package that does not have any graded questions in it, it will not be SCORM-compliant. I don't mean that it just won't track because it has no grade to send; I mean it is *not* compliant. In tracing method calls from Captivate lessons, we've found that a lesson with no

questions will not make the required call to the LMS to initialize itself upon startup. It will make the *finalize* call upon exit, however any compliant LMS will throw back an error when this happens. The spec dictates that a content package must initialize itself before it can finalize itself. Makes sense, right?

A tangential problem to this is that a lesson with no questions (even if the correct initialize and finalize calls are made) has no way to tell when it's been completed, so it does not send that information to the LMS either. This secondary problem is not an issue of compliance as the SCORM specification does not require this information to be sent, but more an issue of usability. What's the point of making a SCORM-compliant lesson and loading it into an LMS if you never find out when your users have completed it?

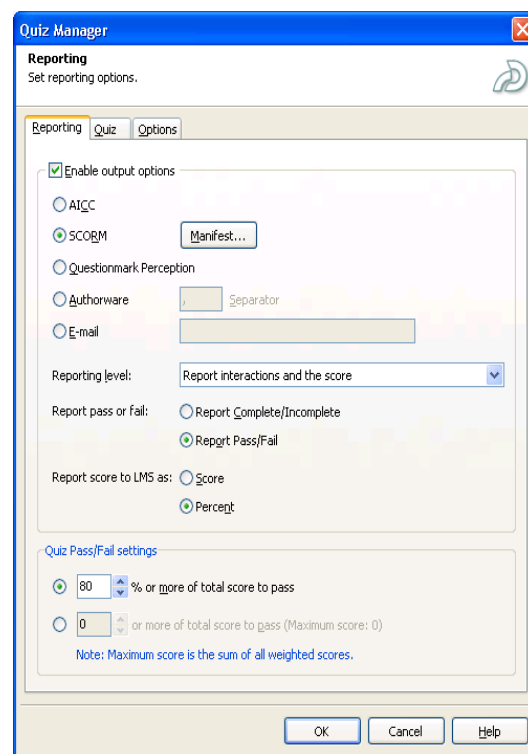
The Solution:

The resolution to both these problems is easy – just make sure that you have a graded interaction in your lesson. It can be an interaction that is actually presented as such or even a button or hot spot that you are sure your users will click while viewing the lesson. The possibilities here are endless, so be sure to test your solution, but the bottom line is that there needs to be at least one graded interaction in your lesson.

Passing the Proper Lesson Status Value

Problem #2: Adobe Captivate lets you choose whether to report 'pass/fail' or 'complete/incomplete' values for *lesson status*, but this is not an arbitrary choice. The spec dictates that this shall be determined by the lesson after querying the LMS and deciding based upon the response it receives.

When publishing with Captivate, if you select complete/incomplete, and the user fails or fails to finish the lesson, the value of 'incomplete' will be reported to the LMS. In the event that the user completes or passes the lesson, the value of 'complete' will be reported to the LMS. Likewise, if you select pass/fail, then the value of 'pass' will be used instead of 'complete' and the value of 'fail' will be used instead of 'incomplete'.



Additionally, Captivate lessons never query the LMS for the value of 'credit', which is the element that the lesson should be using to determine whether to use 'complete/incomplete' or 'pass/fail'.

The Solution:

Solving this problem may or may not even be necessary – it's a rather minor issue. The best thing to do is make sure that you coordinate the *credit* setting you use in the LMS with the *lesson status* value you select here. Lessons that are for credit should use 'pass/fail' and lessons that are not for credit should use 'complete/incomplete'. However, one thing to note, and this takes us indirectly to Problem #3 and beyond, is that the spec dictates that the LMS reevaluate the score and change this value *if you have set a mastery score*. We'll come back to this when we get to Problem #4.

Passing Score In The Proper Format

Problem #3: The 'Publish' interface in Adobe Captivate lets you choose whether to report the *score* as a raw value or as a percentage while the spec dictates that this value must be 'normalized between 0 and 100' (meaning it must be a percentage score). When you choose to report this value as a raw score, your lesson is not compliant.

Adobe tells us that they put this option for a very specific reason. The spec defines three values relating to score and all shall be normalized between 0 and 100 – *minimum score*, *maximum score* and what they call *raw score* (oddly enough, the spec calls it 'raw' score and at the same time dictates that it be normalized – no wonder everyone is confused). Logically, since they are required to be normalized between 0 and 100, *minimum score* would always be 0 and *maximum score* would always be 100 so why even use them? Because of this confusion, Adobe decided to allow the content author to decide whether to report *score* as raw or normalized.

The problem occurs when you choose to report *score* as raw and then load your content into an LMS that has been implemented according to the SCORM spec because it will expect to receive score normalized. Confusion ensues!

Scenario:

You create a Captivate lesson and choose to report *score* as a raw value. Your lesson has 5 questions and your user gets them all correct. Your lesson is going to report '5' as the score and a compliant LMS is going to interpret this as 5%. Of course, your lesson should also report a *lesson status* of 'complete' or 'passed' (see problem #2) which will truly confuse your user when they look at their stats and see that they passed/completed a lesson with a score of only 5%!

The Solution:

This is an easy one. Unless you are certain that your LMS implements *score* as a raw value, always select 'percentage' to ensure that your lesson is compliant.

Tying It All Together

Problem #4: Take a deep breath, because problem #4 might get a little confusing. The SCORM specification instructs the LMS to change the *lesson status* (the same value discussed in problem #2) when certain conditions apply. When this happens, the LMS shall use the *score* to decide how to change the *lesson status* value. If you remember though, from problem #3, you may be reporting *score* as a non-compliant raw value, so the LMS may change the *lesson status* based on bad information.

To get a better understanding of this, let's introduce *mastery score*. You set the *mastery score* by clicking the 'Manifest' button on the Publish Interface. *Mastery score* is value stored in the manifest file that is included in the content package you load into the LMS. The LMS reads this value and stores it with the lesson. If you notice, Captivate instructs that this value should be between 0 and 100, or normalized.

Now the SCORM specification instructs the LMS that if *mastery score* is set, the lesson is being taken for credit and the *lesson status* is not 'incomplete', the LMS shall change the *lesson status* to the appropriate value (complete, incomplete, pass or fail) by comparing the *score* reported from the lesson and the *mastery score* that is defined in the manifest. This occurs even if the lesson has already passed a value for *lesson status*.

The screenshot shows a 'Manifest' dialog box with the following fields and options:

- SCORM Version:** Version: 1.2 (dropdown)
- Course:** Identifier, Title, Description (text fields)
- Version:** Version: 1.0
- Duration:** Duration: (hh:mm:ss)
- Subject:** Subject:
- SCO:** Identifier, Title (text fields)
- Mastery score:** Mastery score: 80 (0-100)
- Time allowed:** Time allowed: (hh:mm:ss)
- Time limit action:** Time limit action: Exit with message (dropdown)
- Launch data:** Launch data:

Buttons: OK, Cancel, Help

The first thing to notice is that you probably should set the *mastery score* to the same value that you set passing score. That way, if the LMS re-evaluates the *lesson status*, it will use the same value as the passing score that the lesson itself does.

Now let's refer back to Problem #3. You had the option of reporting *score* as a raw value. If you chose that option, when the LMS performs this re-evaluation of *lesson status*, it is going to compare a raw *score* to the normalized *mastery score*. Since one value is normalized and the other is not, it should be clear that you will have some unexpected results from this.

Scenario:

You create a Captivate lesson with 20 questions. You choose to report *score* as a raw value (non-complaint per Problem #3, but Captivate lets you do it), choose to use 'pass/fail' for *lesson status*, enter a *mastery score* of 80% and enter a passing score of 80%. Your user gets 17 questions correct.

When the lesson finalizes, the lesson reports 'pass' to the LMS for *lesson status* and '17' for *score*. Everything looks good until the LMS sees that there is a *mastery score* and therefore it must re-evaluate the *lesson status*. The LMS looks at *score* ('17') and sees that it is less than *mastery score* ('80'), so it changes *lesson status* to 'fail'. In fact, a lesson created with these settings will *always* have its *lesson status* re-evaluated to 'fail' by the LMS because even a perfect raw *score* ('20') will always be less than the *mastery score* ('80').

The root of the problem is that Captivate prompts you to enter *mastery score* normalized, but gives you the option to report *score* as a raw value. They need to be on the same scale for the re-evaluation by the LMS to work properly.

The Solutions:

#1: Don't enter a *mastery score*. By doing this, the LMS will not re-evaluate the *lesson status* and you avoid the problem altogether. But don't forget about Problem #3 and its solution.

#2: Make sure that *mastery score* and *score* are both normalized by choosing to report *score* as a 'percentage.' You'll notice that this is also the solution to Problem #3. If you have confirmed that your LMS expects to receive *score* as raw, then use Solution #3.

#3: If you must report *score* as a raw value, then be sure to enter a raw value for *mastery score*. In our scenario, instead of entering '80' for *mastery score*, you would enter '17'. That way, when the LMS re-evaluates *lesson status*, both *score* and *mastery score* are on the same scale and the calculation is done correctly. While technically incorrect since the spec dictates that *mastery score* be normalized, we won't worry about it because you'd only use this solution in the case that your LMS is also non-compliant because it's expecting raw values for *score*. It's a workaround.

Summary

We've seen that Adobe Captivate provides a robust solution for quickly developing online training solutions. But let's not forget that we need to be mindful of the implementation of the SCORM specification by the LMS and how it's going to react to our Captivate lessons.

Review the problems and their solutions and you can be sure that your Captivate lessons are going to comply with SCORM 1.2 and function properly when loaded into a SCORM 1.2-compliant learning management system.

For more information on ICS Learning Group or how we can help you develop and integrate your online training program, visit us on the web at www.icslearninggroup.com or give us a call: (410) 975-9440.