The Electronic Staff Record Programme

National Health Service

ESR-NHS0167 – ESR e-Learning Content Guidance and Standards

Information Classification: Public

Owner:	Nick Adcock
Author:	Matthew Lye
Creation Date:	10 March 2008
Last Updated:	23 March 2022
Next Update:	23 March 2023
Version:	11

Approvals:

Name	Lee Pacey
Title	Director of Development and Operations

1 DOCUMENT CONTROL

1.1 CHANGE RECORD

Date	Author	Version	Change Reference
04/02/2008	Iain Harding	0.1	Initial Draft
16/02/2008	Iain Harding	0.2	Redrafted after comments
20/02/2008	Iain Harding	0.3	Review of document
21/02/2008	Iain Harding	0.4	Second review of document
21/02/2008	Iain Harding	0.5	Awaiting approval and Sign Off
29/02/2008	Stuart Fox	0.51	Updated document following feedback
06/02/2008	Iain Harding	0.52	Updated following feedback from e-LfH
10/03/08	Stuart Fox	0.9	Amended to v 0.9 for official approvals process
11/03/08	Stuart Fox	1.0	v1.0 issued for sign-off
25/03/08	Stuart Fox	1.1	Classification changed to Commercial in Confidence
25/03/08	Stuart Fox	2.0	Issued for Sign-off
06/11/08	Nick Moseley	2.1	Addition of section covering packaging of content
11/03/09	James Haddon	2.2	Addition of section covering screen resolution and amendments to SCORM calls.
24/02/10	James Haddon	3.0	Removed irrelevant sections, added checklist
19/10/10	James Haddon	3.1	Amended examples, various minor changes.
28/10/10	James Haddon	4.0	Issued for sign-off
25/07/11	James Haddon	4.1	Updated SCORM Calls, updated screens/formatting
1/9/11	Julie Bickerton	5.0	Uplifted to version 5.0 for publication
14/08/2013	Matthew Lye	5.1	Amended to include SCORM 2004 information, various minor changes.
20/08/2013	Matthew Lye	6.0	Uplifted to version 6.0 for approval
16/09/2013	Nick Adcock	6.0	Approved
02/12/2014	Matthew Lye	6.1	Amended to include information about JSP adapter, various minor changes
09/12/14	Nick Adcock	7.0	Approved
27/05/15	Matthew Lye	7.1	Amended due to change of ESR Service Provider
19/06/2015	Nick Adcock	8.0	Approved and uplifted
26/04/2016	Matthew Lye	8.1	Annual review,
02/03/2018	Matthew Lye	8.2	Annual review, addition of advice for e-Learning packages and testing local contentin TPLY using Non JRE
01/06/2018	Matthew Lye	8.3	Added section on common issues for e-Learning developers
08/11/2018	Nick Adcock	9.0	Approved
23/11/2019	Matthew Lye	9.1	Updates following review
08/01/2019	Nick Adcock	10.0	Approved for uplift following review
17/09/2021	Matt Lye	10.1	Reviewed, no changes
23/03/2022	Nick Adcock	11	Version uplift

1.2 REVIEWERS

Name	Position
James Haddon	NHS ESR Development Team
Chris Moorley	NHS ESR Development Team
Nicole Stacey	NHS ESR Development Team
	NHS ESR Functional Account Managers

1.3 DISTRIBUTION

Copy No.	Name	Location
1	Library Master	NHS Document Library

2 CONTENTS

3 PURPOSE AND BACKGROUND

3.1 INTRODUCTION

ESR provides the ability for NHS Organisations to deliver e-Learning content to individuals with a person record.

E-Learning content from national providers is available together with the ability for Organisations to deliver locally produced e-Learning content.

This document provides guidance and information to e-Learning developers to enable them to procure or create content, internally through their own development teams, and communicate effectively with external suppliers.

3.2 CONTENT STANDARDS

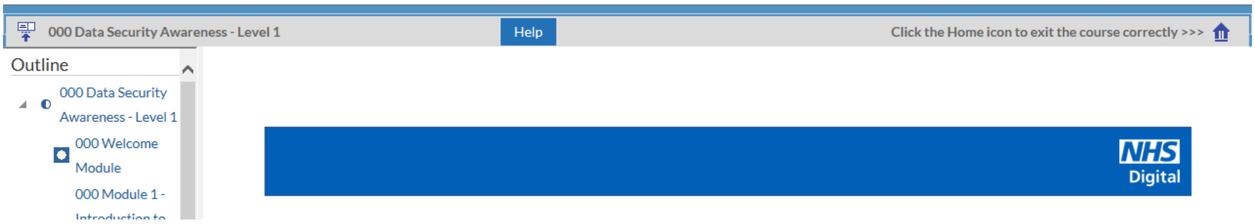
Where organisations are creating web based content they should try to ensure that it meets with guidelines defined for HTML content by the World Wide Web Consortium (W3C) and with Web Content Accessibility Guidelines (WCAG) for accessibility.

Please note: NHS Digital has developed a Standard in order to help comply with the Equality Act 2010 in the design, development and delivery of training materials. More information on these standards can be found at https://digital.nhs.uk/services/training-quality-improvement/education-and-training-standards-and-benchmarking

4 HOW ESR PLAYS CONTENT

4.1 The OLM Player

e-Learning is displayed in ESR in the OLM Player. This generates the toolbar and outline frames from data about the content and the learner stored in the database. Each learning object in the course outline has a content location stored in the database. When a learner selects a learning object in the outline, the corresponding content is loaded from a content server into the content frame on the right.


000 Data Security Awareness - Level 1
Help
Click the Home icon to exit the course correctly >>>
Outline
000 Data Security Awareness - Level 1
000 Welcome Module
000 Module 1 -
Introduction to
NHS
Digital

4.2 Navigational Control

Individual learning objects may have multiple pages and navigation is controlled by content. Navigation between learning objects is handled by the OLM Player from the outline on the left. Content should not contain internal links to other learning objects as the content will not be tracked correctly.

4.3 Exiting the Course

The OLM Player has its own exit button in the shape of a ‘home’ icon, this means that Exit buttons are not required in content.

Once clicked, this home icon will return the learner back to the point where they launched the e-Learning from.,. In order for content to track correctly, a user-initiated event is normally required upon closing the content to record the student’s current location, status and score. This can be achieved by placing the function required in the ‘onunload’ and ‘onbeforeunload’ HTML events. An example of how this could be used is shown below:

```
<body onunload="stopLMS();” onbeforeunload="stopLMS();">

<script language="javascript">
function stopLMS(){
API.LMSCommit("");
API.LMSFinish("");
}
</script>
```

In SCORM 2004

```
<body onunload="stopLMS();” onbeforeunload="stopLMS();">
```

```
<script language="javascript">
function stopLMS(){
API.Commit("");
API.Terminate("");
}
</script>
```

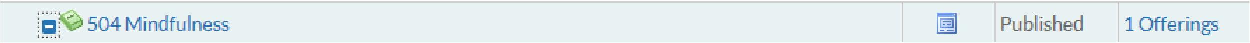
If the content has an exit button, it should not attempt to close the OLM Player window. Instead the exit button should make the relevant calls, and then instruct the user to close the course manually

5 CONTENT DEPLOYMENT GUIDELINES

5.1 Ensuring Accurate Tracking

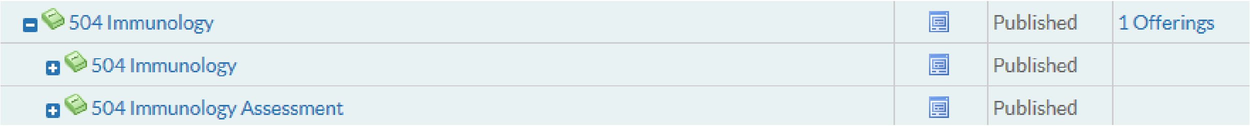
Content in ESR can be set up in two ways.

1. Course linked to a single learning object



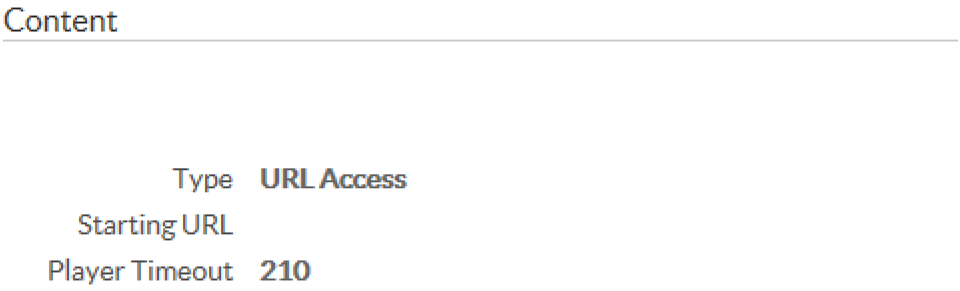
⊟ 504 Mindfulness	▤	Published	1 Offerings

2. Course linked to a parent learning object with multiple children



⊟ 504 Immunology	▤	Published	1 Offerings
⊞ 504 Immunology	▤	Published	
⊞ 504 Immunology Assessment	▤	Published	

Where a course has a parent learning object, it is important to ensure that the parent learning object does not contain a URL.



Content

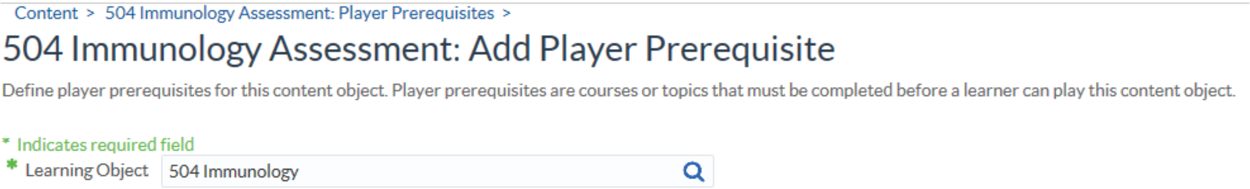
Type URL Access
Starting URL
Player Timeout 210

The completion status and time played for the parent learning object is calculated by aggregating the completion status and time spent from the child learning objects.

When all child learning objects are completed, the parent learning object is marked complete and reflects the total time the course has been played.

5.2 Pre-requisites

Where a course has child learning objects, the learner can play these in any order. If a learner must complete a particular child object before playing others, then this can be set as a prerequisite.



Content > 504 Immunology Assessment: Player Prerequisites >

504 Immunology Assessment: Add Player Prerequisite

Define player prerequisites for this content object. Player prerequisites are courses or topics that must be completed before a learner can play this content object.

* Indicates required field

* Learning Object: 504 Immunology

5.3 Using Existing Content

Organisations have many existing resources that can be used as e-Learning content. A learning object can point to files including Microsoft PowerPoint, HTML pages, Microsoft Word, PDF etc. A course can easily be constructed as a series of these types of resources.

5.4 Building Content

A learning object may have multiple pages of HTML content. ESR holds the location of the first (e.g. index.html) HTML file that is launched for the object.

5.5 Frames

To play correctly in the OLM Player, content must be frame-friendly. The content may contain browser frames of its own and may manage those frames, it cannot reference frames that ESR provides.

In particular, the content should never make reference to the window.top JavaScript object or use _top as the target for a URL as this will cause errors.

If errors occur, the code can be revised by looking for error messages produced in the browser.

In Internet Explorer, a warning icon is shown in the bottom left hand corner of the browser window. Double-clicking on this icon shows the relevant JavaScript error that occurred. This may assist in determining which line of code is causing the error.

5.6 Screen Resolution

Most users in the NHS have a screen resolution of no less than 1024x768 pixels. Content can be designed to resize to the viewable area or written to fit in the space available. If content is too large, users will have to use scroll bars to use content effectively.

The screen real-estate available for users to see will depend on the structure of the content. In ESR, there are two bars that are displayed:

Outline – approximately 210 pixels wide, located on the left of the screen - this is only required where content has multiple learning objects
Toolbar - approximately 45 pixels tall, located at the top of the screen – this contains additional help for learners and the exit button. It is recommended that this is always displayed.
The space available for the content is summarised below:

	Space Available
With outline	810 (w) x 610 (h)
Without outline	1020 (w) x 610 (h)

5.7 Tests

Tests enable you to create and manage tests either as learning objects within a course, or as courses in their own right. Tests can be scored, or set up as surveys for which there is no score. Tests are made up of questions stored in a question bank created in ESR.

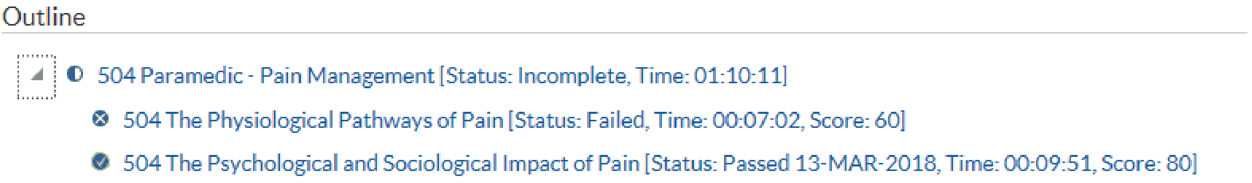
ESR handles presentation of the test questions, scores the test, and records the learner's results.

5.8 UNIX SERVERS AND CASE SENSITIVITY

All URLs for UNIX web servers are case sensitive, but URLs for Microsoft Windows web servers are not. Frequently, content is developed in a Windows environment and may not be tested on UNIX servers prior to its release.

6 TRACKING LEARNER PROGRESS IN CONTENT

The tracking of a learner's progress is directly tied to the learning objects within a course. When a learner plays a course for the first time, an attempt record for the learner is created in ESR. This stores information such as the time spent, completion status, and score for that learning object. When the learner leaves the learning object the attempt record is finished and a performance record for the learning object is created (or updated if the content has been played before). It is the data stored in the performance record that a learner sees in ESR, showing progress with the parent and child learning objects.



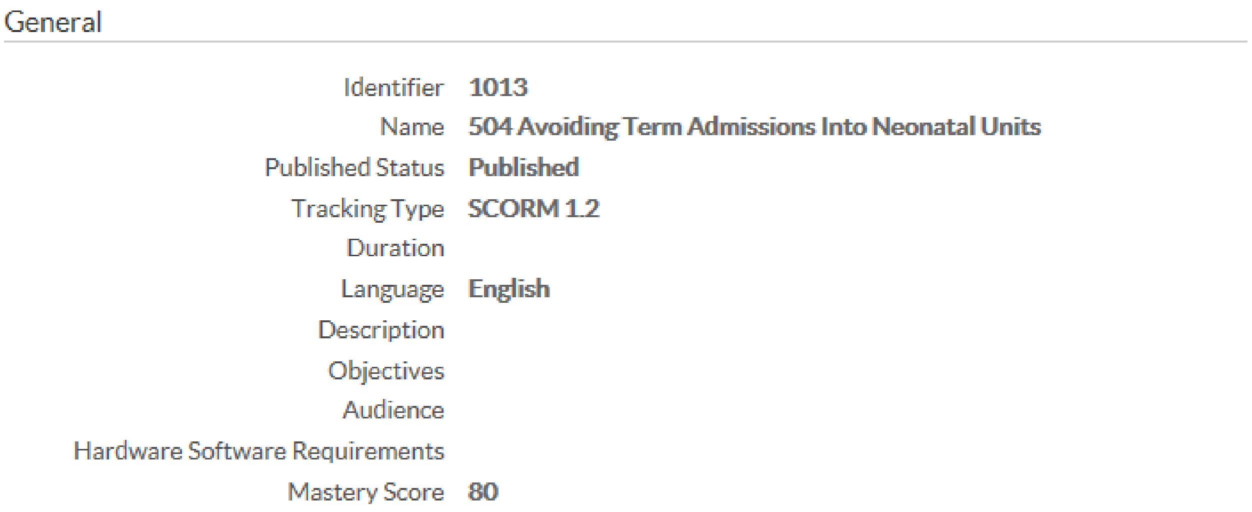
Outline

- ◐ 504 Paramedic - Pain Management [Status: Incomplete, Time: 01:10:11]
  - ⊗ 504 The Physiological Pathways of Pain [Status: Failed, Time: 00:07:02, Score: 60]
  - ✓ 504 The Psychological and Sociological Impact of Pain [Status: Passed 13-MAR-2018, Time: 00:09:51, Score: 80]

The performance record is updated to reflect learner progress up to the point where an attempt is marked as completed or passed. At that point, the performance record completion status is updated with the latest status (and optionally a score) and cannot be modified by subsequent attempts of a learning object. The time spent on the performance record continues to be updated however, based on subsequent learner attempts of the learning object.

ESR has two methods for tracking a learner’s progress through content: Automatic tracking and SCORM tracking. A learning object can track, for each learner, a completion status (not attempted, incomplete, completed, passed, failed, browsed), the time spent in the content, and a score.

ESR can hold a Mastery Score against a Learning Object, where this is entered, it will take precedence over one defined in the content.
A mastery score of 80 has been entered against the learning object.



General

Identifier	1013
Name	504 Avoiding Term Admissions Into Neonatal Units
Published Status	Published
Tracking Type	SCORM 1.2
Duration	
Language	English
Description	
Objectives	
Audience	
Hardware Software Requirements	
Mastery Score	80

A user plays the content and scores 75, the content returns a status of ‘Completed’ and a score of 75. ESR will ignore the status and set the user’s status to ‘Failed’ as they have not reached the mastery score.

6.1 Automatic Tracking

ESR can automatically track the launch of and time spent in any content. When set to automatic tracking, each topic is marked as completed as soon as it is launched. This may be useful for playing and reporting on content that is not SCORM compliant such as a simple web page or a PDF document.

6.2 SCORM Tracking

For more detailed tracking, ESR supports the CMI data model in the SCORM specification published by the ADL. (See https://www.adlnet.gov/scorm/ for more information). SCORM tracking uses JavaScript calls for the content to communicate with the LMS.

6.3 SCORM Adapters

ESR does not store e-Learning content so it must be on a separate web server.

The JavaScript security model does not allow scripts in frames coming from different domains to communicate. To overcome this limitation, ESR uses a SCORM Adapter. The adapter is a set of files that are stored on the remote server with the content files.

Details on the types of SCORM adapter available to organisations and content providers are available here, https://my.esr.nhs.uk/dashboard/web/esrweb/e-learning-technical-information, along with the steps for setup. Wherever possible organisations should use the Non JRE SCORM adapters as these enable e-Learning to be played in internet enabled PCs and mobile devices.

When SCORM compliant content is launched, ESR scans its list of registered SCORM adapters. If the content comes from a location with a registered SCORM adapter, the URL is automatically rewritten to redirect the learner's browser to the SCORM adapter files, which in turn loads the content files.

Installing SCORM 1.2 and SCORM 2004 Adapters

If both SCORM 1.2 and 2004 content are held on servers then both adapters must be installed and registered separately as described above.

SCORM 1.2 and SCORM 2004 Adapters must be placed in separate directories on the server eg JSAdapter12_php/Oracle_SCORM_Adapter_JS.html and https://www.myorganisation.nhs.uk/elearning/JSAdapter12_php/Oracle_SCORM_2004_Adapter_JS.html

The URL for the SCORM 2004 adapter must contain 2004 in the address for it to be recognised as a SCORM 2004 adapter.

Please refer to the next section for ESR's implementation of the SCORM specification.

The CMI data model allows statuses and scores to be set against each content object. The SCORM specification uses the term "SCO" (Shareable Content Object), which is equivalent to a learning object in ESR. The CMI API also allows the content to query ESR for data about the learner and their past interactions with the content.

As specified by the ADL for SCORM tracking, the content must contain code to find the API, initialise the API, get and set values as appropriate, and finalise transaction upon exit. The specific technology used by the content to implement its tracking logic is not important, as long as it can make the correct calls via JavaScript to meet the SCORM specification.

6.4 ESR SCORM IMPLEMENTATION

The SCORM specification has an extensive list of data elements, some mandatory and many optional. ESR is “LMS RTE 2”, this means that it supports the mandatory data elements of the SCORM specification only. The data elements supported in ESR are listed below:

6.4.1 SCORM 1.2 Calls Compatible with ESR

Type	Calls	Comments
Core	exit, score (raw only) ,credit, lesson_location, session_time, total_time, lesson_mode, student_id,student_name, lesson_status, entry	
N/A	suspend_data, launch_data,	
Objectives	score (raw only), status, id	Each objective must be given an id).
LMS	LMSGetLastError, LMSSetValue, LMSGetValue, LMSGetLastError, LMSGetErrorString, LMSGetDiagnostic, LMSCommit, LMSFinish, LMSInitialize	

Content running in ESR should ideally record as a minimum: lesson_status, lesson_location and score (where an assessment is included).

6.4.2 SCORM 1.2 Calls not Compatible with ESR

Type	Calls	Comments
Core	score.min, score.max	
N/A	comments, comments_from_lms	
Objectives	score.max, score.min	
Student_data	mastery_score, max_time_allowed, time_limit_action	
Student_preference	audio, language, speed, text	
Interactions	id, objectives, time, type, correct_response, weighting, student_response, result, latency	

Should any of the above calls be made to ESR, the LMS will respond with:
"LMS Error: 401 - Not implemented error"

This should not prevent the content from functioning. Content should be designed so that if an LMS does not support one of the optional calls above, then it should continue with the calls that are supported.

For example:

Content calls LMSGetValue("cmi.core.score.max")

LMS will experience the following error: "LMS Error: 401 - Not implemented error" but return a null value ("") to the content.

The content should be designed so that if a null value is returned, it should continue without the information (i.e. in this example, not knowing the maximum score should not prevent the content from setting the user's attained score).

Core Element Name	Definition
student_id	The username of the user.
student_name	The student's name.
lesson_location	This corresponds to the lesson's exit point passed to the CMI system the last time the learner experienced the lesson.
Credit	Indicates whether the content gives credit for completion.
lesson_status	The status of the lesson – i.e. Incomplete, Completed.
entry	Indication of whether the student has been in the lesson before.
exit	Indication of how or why the student left the session
score.raw	Numerical representation of student performance in lesson. May be unprocessed raw score.
total_time	Accumulated time of all the student sessions in the lesson.
lesson_mode	Identification of student related information that may be used to change the behaviour of the lesson.
suspend_data	Unique information generated by the lesson during previous uses that is needed for the current use. Please note, as per SCORM 1.2 standards, ESR limits the amount of data stored in this field to 4096 characters. Attempts to commit more than this data may result in error or truncation of the data.
launch_data	Unique information generated at the lesson's creation that is needed for every use.
session_time	Time spent in the lesson during the session that is ending.

Note that the CMI terms "lesson" and "AU" (assignable unit) refer to a learning object in ESR.

Objective Element Name	Definition
id	A unique identifier for each objective. Note: if you are using objectives in your content, each objective MUST have an id, otherwise the communication may fail.
score	Indication of the score obtained by the student after each attempt to master an objective.
raw	This may be an unprocessed or processed indicator of how the student performed with the AU's interactions (related to the objective) experienced.
max	This is the largest score the student could have with the AU's interactions (related to the objective) experienced.

min	This is the smallest score that the student could have achieved with the AU's interactions (related to the objective) experienced.
status	Indication of the status of an objective.

6.4.3 SCORM 2004

ESR is compatible with SCORM 2004 4th edition and all calls are mandatory in this standard.

SCORM 2004 enables a learner to record multiple attempts on an e-learning course. Rather than seeing completed against a course they will have a status of Multiple.

Clicking on the Multiple hyperlink will enable them to view each individual attempt

Type	**Calls**
Core	cmi._version, cmi.completion_status, cmi.completion_threshold, cmi.credit, cmi.entry, cmi.exit, cmi.launch_data, cmi.learner_id, cmi.learner_name, cmi.location, cmi.max_time_allowed, cmi.mode, cmi.progress_measure, cmi.scaled_passing_score, cmi.score.scaled, cmi.score.raw, cmi.score.min, cmi.score.max, cmi.session_time, cmi.success_status, cmi.suspend_data, cmi.time_limit_action, cmi.total_time
Objectives	cmi.objectives._count, cmi.objectives.n.id, cmi.objectives.n.score._children, cmi.objectives.n.score.scaled, cmi.objectives.n.score.raw, cmi.objectives.n.score.min, cmi.objectives.n.score.max, cmi.objectives.n.success_status, cmi.objectives.n.completion_status, cmi.objectives.n.progress_measure, cmi.objectives.n.description
Interactions	cmi.interactions._count, cmi.interactions.n.id, cmi.interactions.n.type, cmi.interactions.n.objectives._count, cmi.interactions.n.objectives.n.id, cmi.interactions.n.timestamp, cmi.interactions.n.correct_responses._count, cmi.interactions.n.correct_responses.n.pattern, cmi.interactions.n.weighting, cmi.interactions.n.learner_response, cmi.interactions.n.result, cmi.interactions.n.latency, cmi.interactions.n.description
Navigation	adl.nav.request, adl.nav.request_valid.continue, adl.nav.request_valid.previous, adl.nav.request_valid.choice.{target=}, adl.nav.request_valid.jump.{target=}
Comments	cmi.comments_from_learner._count, cmi.comments_from_learner.n.comment, cmi.comments_from_learner.n.location, cmi.comments_from_learner.n.timestamp, cmi.comments_from_lms._children, cmi.comments_from_lms._count, cmi.comments_from_lms.n.comment, cmi.comments_from_lms.n.location, cmi.comments_from_lms.n.timestamp
Learner Preferences	cmi.learner_preference.audio_level, cmi.learner_preference.audio_level, cmi.learner_preference.audio_level, cmi.learner_preference.audio_level
LMS	Initialize, Terminate, GetValue, SetValue, Commit, GetLastError, GetErrorString, GetDiagnostic

Element Name	Comments
cmi._version	Represents the version of the data model
cmi.comments_from_learner._children	Listing of supported data model elements
cmi.comments_from_learner._count	Current number of learner comments
cmi.comments_from_learner.n.comment	Textual input
cmi.comments_from_learner.n.location	Point in the SCO to which the comment applies
cmi.comments_from_learner.n.timestamp	Point in time at which the comment was created or most recently changed
cmi.comments_from_lms._children	Listing of supported data model elements
cmi.comments_from_lms._count	Current number of comments from the LMS
cmi.comments_from_lms.n.comment	Comments or annotations associated with a SCO
cmi.comments_from_lms.n.location	Point in the SCO to which the comment applies
cmi.comments_from_lms.n.timestamp	Point in time at which the comment was created or most recently changed
cmi.completion_status	Indicates whether the learner has completed the SCO
cmi.completion_threshold	Used to determine whether the SCO should be considered complete
cmi.credit	Indicates whether the learner will be credited for performance in the SCO
cmi.entry	Asserts whether the learner has previously accessed the SCO
cmi.exit	Indicates how or why the learner left the SCO
cmi.interactions._children	Listing of supported data model elements
cmi.interactions._count	Current number of interactions being stored by the LMS
cmi.interactions.n.id	Unique label for the interaction
cmi.interactions.n.type	Which type of interaction is recorded
cmi.interactions.n.objectives._count	Current number of objectives (i.e., objective identifiers) being stored by the LMS for this interaction
cmi.interactions.n.objectives.n.id	Label for objectives associated with the interaction
cmi.interactions.n.timestamp	Point in time at which the interaction was first made available to the learner for learner interaction and response
cmi.interactions.n.correct_responses._count	Current number of correct responses being stored by the LMS for this interaction

cmi.interactions.n.correct_responses.n.pattern	One correct response pattern for the interaction
cmi.interactions.n.weighting	Weight given to the interaction relative to other interactions
cmi.interactions.n.learner_response	Data generated when a learner responds to an interaction
cmi.interactions.n.result	Judgment of the correctness of the learner response
cmi.interactions.n.latency	Time elapsed between the time the interaction was made available to the learner for response and the time of the first response
cmi.interactions.n.description	Brief informative description of the interaction
cmi.launch_data	Data provided to a SCO after launch, initialized from the data. From LMS manifest element
cmi.learner_id	Identifies the learner on behalf of whom the SCO was launched
cmi.learner_name	Name provided for the learner by the LMS
cmi.learner_preference._children	Listing of supported data model elements
cmi.learner_preference.audio_level	Specifies an intended change in perceived audio level
cmi.learner_preference.language	The learner’s preferred language for SCOs with multilingual capability
cmi.learner_preference.delivery_speed	The learner’s preferred relative speed of content delivery
cmi.learner_preference.audio_captioning	Specifies whether captioning text corresponding to audio is displayed
cmi.location	The learner’s current location in the SCO
cmi.max_time_allowed	Amount of accumulated time the learner is allowed to use a SCO
cmi.mode	Identifies one of three possible modes in which the SCO may be presented to the learner
cmi.objectives._children	Listing of supported data model elements
cmi.objectives._count	Current number of objectives being stored by the LMS
cmi.objectives.n.id	Unique label for the objective
cmi.objectives.n.score._children	Listing of supported data model elements
cmi.objectives.n.score.scaled	Number that reflects the performance of the learner for the objective
cmi.objectives.n.score.raw	Number that reflects the performance of the learner, for the objective, relative to the range bounded by the values of min and max

cmi.objectives.n.score.min	Minimum value, for the objective, in the range for the raw score
cmi.objectives.n.score.max	Maximum value, for the objective, in the range for the raw score
cmi.objectives.n.success_status	Indicates whether the learner has mastered the objective
cmi.objectives.n.completion_status	Indicates whether the learner has completed the associated objective
cmi.objectives.n.progress_measure	Measure of the progress the learner has made toward completing the objective
cmi.objectives.n.description	Provides a brief informative description of the objective
cmi.progress_measure	Measure of the progress the learner has made toward completing the SCO
cmi.scaled_passing_score	Scaled passing score required to master the SCO
cmi.score._children	Listing of supported data model elements
cmi.score.scaled	Number that reflects the performance of the learner
cmi.score.raw	Number that reflects the performance of the learner relative to the range bounded by the values of min and max
cmi.score.min	Minimum value in the range for the raw score
cmi.score.max	Maximum value in the range for the raw score
cmi.session_time	Amount of time that the learner has spent in the current learner session for this SCO
cmi.success_status	Indicates whether the learner has mastered the SCO
cmi.suspend_data	Provides space to store and retrieve data between learner sessions
cmi.time_limit_action	Indicates what the SCO should do when cmi.max_time_allowed is exceeded
cmi.total_time	Sum of all of the learner’s session times accumulated in the current learner attempt
adl.nav.request	Navigation request to be processed immediately following Terminate()
adl.nav.request_valid.continue	Used by a SCO to determine if a Continue navigation request will succeed.
adl.nav.request_valid.previous	Used by a SCO to determine if a Previous navigation request will succeed.
adl.nav.request_valid.choice.{target=}	Used by a SCO to determine if a Choice navigation request for the target activity will succeed.

adl.nav.request_valid.jump.{target=}	Used by a SCO to determine if a Jump navigation request for the target activity will succeed.

7 INSTALLING E-LEARNING CONTENT ON HOSTS

On ESR, the content must not be in zipped format, the files will need to be loaded into a separate folder that can be referenced with a unique starting URL.

All course content must be loaded at the same level or below that of the SCORM adapters in the hierarchy on the content server as in the example below:

https://www.ourserver.com/ourcourses/JSAdapter12_php/Oracle_SCORM_Adapter_JS.html

https://www.ourserver.com/ourcourses/course1/index.html
https://www.ourserver.com/ourcourses/course2/index.html
https://www.ourserver.com/ourcourses/course3/index.html

Courses must also be set up within the ESR course catalogue. Where possible it is recommended to keep the number of learning objects in a course to a minimum. For full instructions on how to do this refer to Quick Guide to Setting up Local e-Learning.

NHS organisations are advised, where possible to test e-Learning in the TPLY environment before release into the production environment commences.

8 COMMON ISSUES FOR E-LEARNING DEVELOPERS

8.1 CONTENT DOES NOT DISPLAY CORRECTLY IN INTERNET EXPLORER

There may be certain content that does not display correctly or certain functionality does not work as expected, for example, drag and drop functionality. Where this is the case, it is recommended that the content is tested in a different browser. If the content can play and track correctly then it is recommended that a Service Request is raised for investigation. For Internet Explorer users, ticking the Open in New Window flag on the Learning Object form should enable the content to work as expected

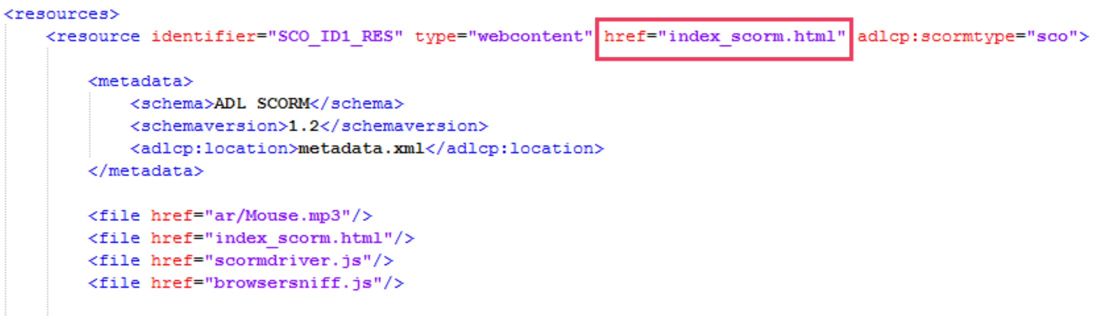
Note - Changing this setting will mean that content will not open in the main player window for all users.

8.2 COURSE NOT TRACKING CORRECTLY

Where a course has been set to track as SCORM 1.2 but does not appear to be tracking correctly, this may be because the correct URL is not being used.

The Starting URL to use in ESR can be identified from the imsmanifest.xml file.

The manifest file can be found in the folder which contains the course files which will be uploaded to a server. In the resources section of the manifest, find the name of the file for the href attribute. This will be the starting URL you must use.



```
<resources>
    <resource identifier="SCO_ID1_RES" type="webcontent" href="index_scorm.html" adlcp:scormtype="sco">

        <metadata>
            <schema>ADL SCORM</schema>
            <schemaversion>1.2</schemaversion>
            <adlcp:location>metadata.xml</adlcp:location>
        </metadata>

        <file href="ar/Mouse.mp3"/>
        <file href="index_scorm.html"/>
        <file href="scormdriver.js"/>
        <file href="browsersniff.js"/>
```

So if the file is uploaded to https://www.contentserver.com/content/course1/:

The starting URL in this example will be https://www.contentserver.com/content/course1/index_scorm.html

9 APPENDIX 2 - CHECKLIST

This checklist is for e-Learning developers to ensure content will perform correctly in ESR.

Item	Reason
Content does not reference OLM Player window (using top.window or parent.window)	Course will reside on a different domain to LMS. JavaScript is not allowed to cross-domains. ‘Permission Denied’ error will occur. This may occur with content produced with Lectora, see guidance in Appendix 5.
Content does not have an ‘exit’ button	The LMS has its own exit button (a ‘home’ icon at the top right). Appropriate calls should be made on the HTML ‘onunload’ and ‘onbeforeunload’ events to record status, bookmarking etc. See section 4.3 for further detail / example.
Ideally content should record as a minimum lesson_status, lesson_location and score (where an assessment is included).	For reporting purposes it must be clear whether a user has completed a piece of e-Learning and that any assessment score is recorded. Bookmarking should also be included so that a user can recommence a course on the page they left.
Content does not make any unsupported calls	ESR is an RTE 2 LMS. ESR will only support the mandatory SCORM 1.2 calls. Should the content make any of these calls (for support in other LMS), you should ensure the content does not display any JavaScript alerts if they error in the OLM player. (Otherwise a ‘Not Implemented’ error will be displayed).
Content is confirmed as SCORM 1.2 or 2004 compliant using the Content Package test in the free Advanced Distributed Learning (ADL) Conformance Test Suite or by using SCORM cloud.	ESR should not be used in the first instance to check if content is SCORM compliant
Content keeps to a minimum number of Learning Objects arranged in a simple hierarchy where possible	Learning objects and the parent child hierarchy have to be set up manually on ESR since there is no facility to import the course manifest metadata.
Content does not attempt to resize itself	Content is displayed in the same window as the OLM Player. Should the content try to modify the OLM Player window in any way, a ‘permission denied’ error will occur.
Content resolution is set within the recommended dimensions	Users will get frustrated if they have to use the scroll bars to view and operate content

10 APPENDIX 3 - SETTING UP A LOCAL SERVER FOR TESTING CONTENT

One of the requirements for using the non-JRE adapters is that the servers must be web facing. TPLY is not web facing so cannot be used for testing with the non-JRE adapters in a conventional way.

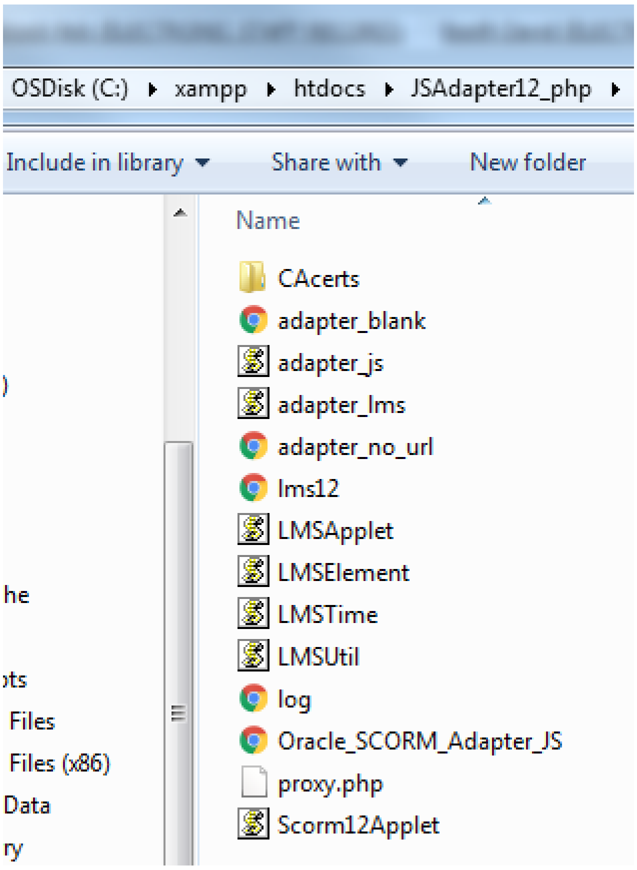
Following the instructions below, it is possible to set up a local server and use this for testing content on TPLY. To assist organisations there is a PHP version of the SCORM adapter registered on TPLY which can be used for this purpose.

The AMP stack used to set up the local server in this document was XAMPP https://www.apachefriends.org/index.html , but other stacks are available and there should not be much variation in set up.

10.1 Install Local Server and PHP SCORM Adapter

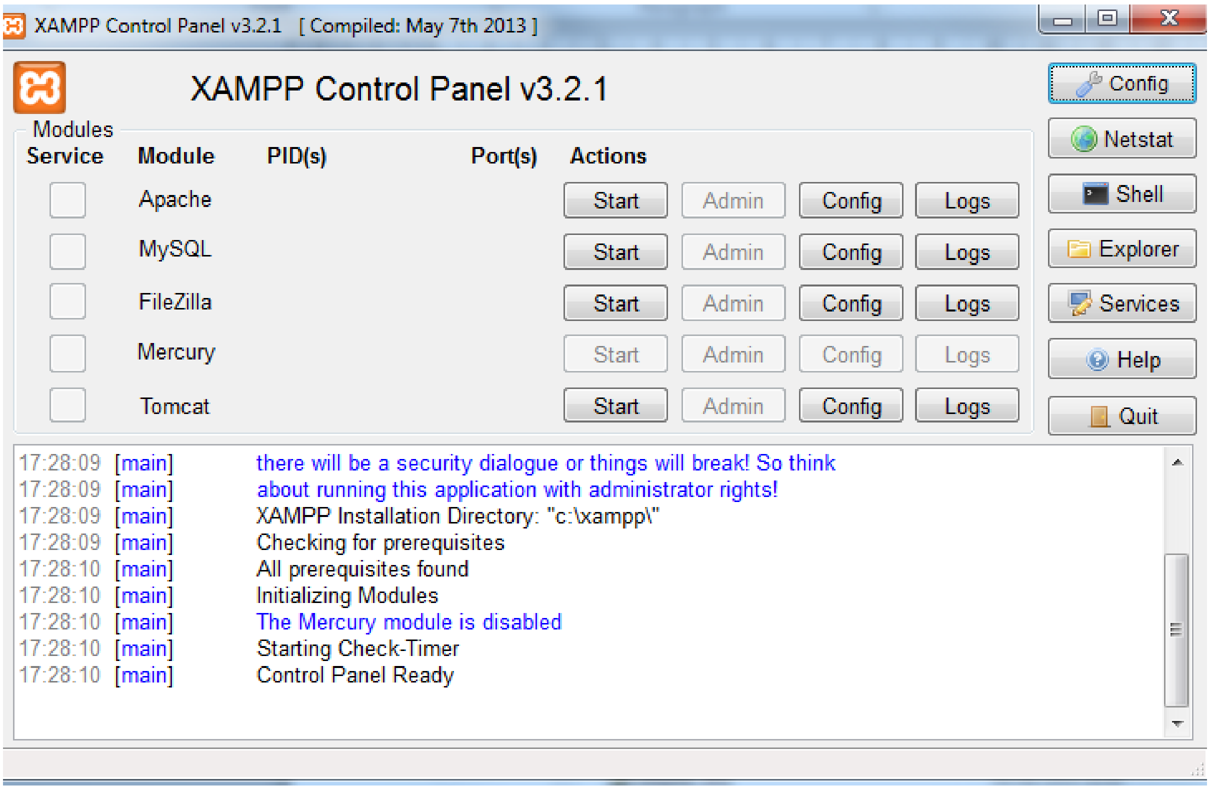
1. Download and install an AMP stack, using the default settings.
2. Download the PHP SCORM adapter.
3. Unzip the SCORM adapter.
4. Copy and paste the SCORM adapter folder to the following location C:\xampp\htdocs.

It should look similar to the screenshot below.



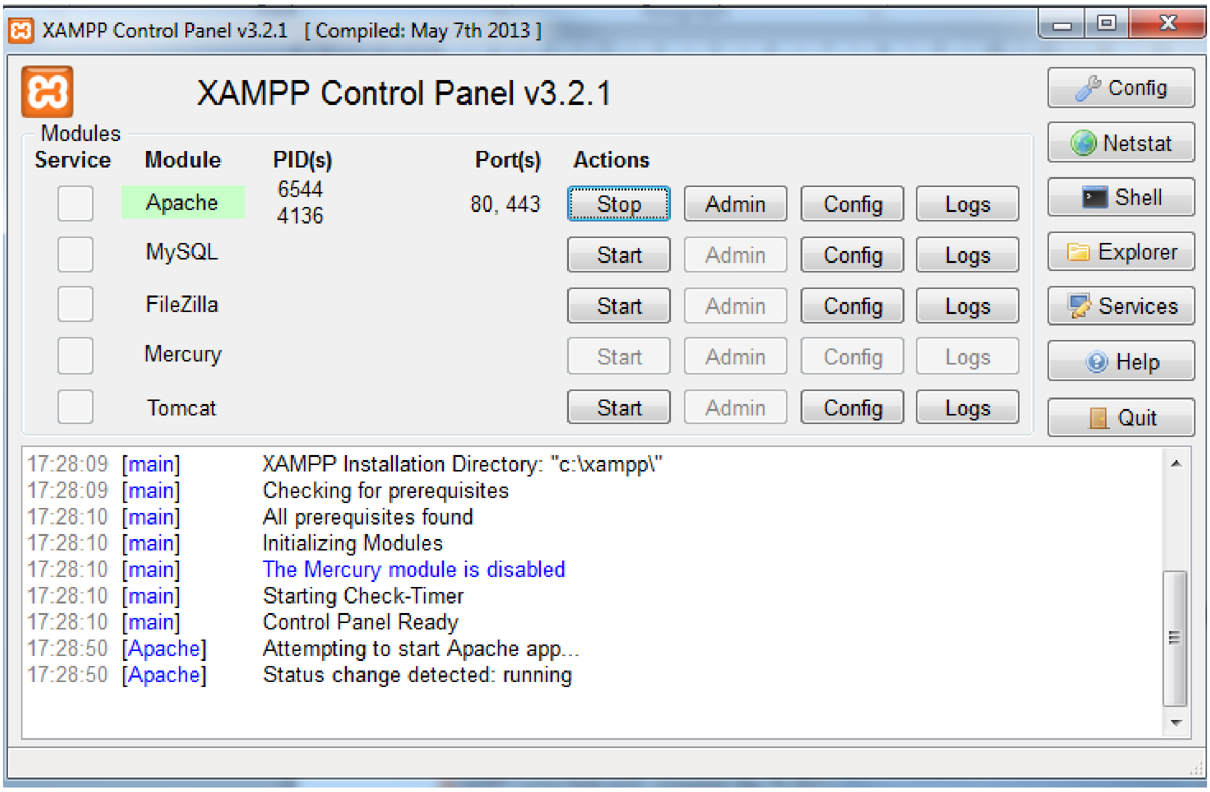
CAcerts
adapter_blank
adapter_js
adapter_lms
adapter_no_url
lms12
LMSApplet
LMSElement
LMSTime
LMSUtil
log
Oracle_SCORM_Adapter_JS
proxy.php
Scorm12Applet

5. Start the XAMPP Control Panel



```
17:28:09  [main]     there will be a security dialogue or things will break! So think
17:28:09  [main]     about running this application with administrator rights!
17:28:09  [main]     XAMPP Installation Directory: "c:\xampp\"
17:28:09  [main]     Checking for prerequisites
17:28:10  [main]     All prerequisites found
17:28:10  [main]     Initializing Modules
17:28:10  [main]     The Mercury module is disabled
17:28:10  [main]     Starting Check-Timer
17:28:10  [main]     Control Panel Ready
```

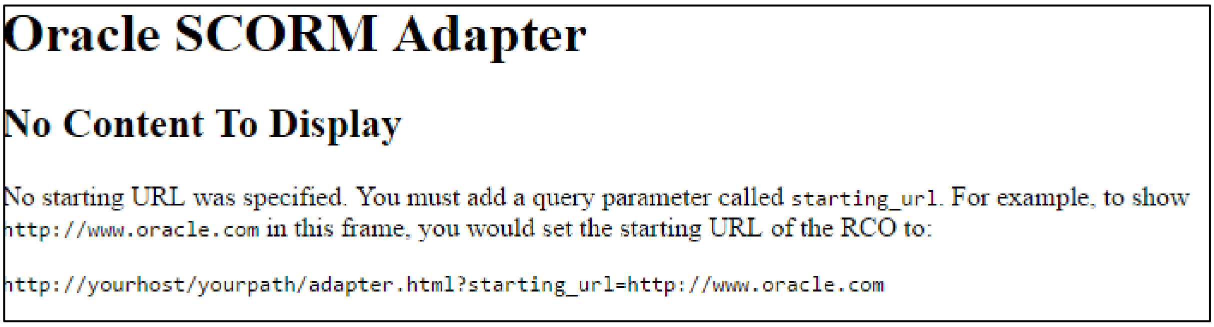
6. Start Apache



```
17:28:09  [main]     XAMPP Installation Directory: "c:\xampp\"
17:28:09  [main]     Checking for prerequisites
17:28:10  [main]     All prerequisites found
17:28:10  [main]     Initializing Modules
17:28:10  [main]     The Mercury module is disabled
17:28:10  [main]     Starting Check-Timer
17:28:10  [main]     Control Panel Ready
17:28:50  [Apache]   Attempting to start Apache app...
17:28:50  [Apache]   Status change detected: running
```

7. Confirm that you can load the SCORM adapter by entering the following URL into a browser

http://127.0.0.1/JSAdapter12_php/Oracle_SCORM_Adapter_JS.html
You should see the following in the browser window



Oracle SCORM Adapter

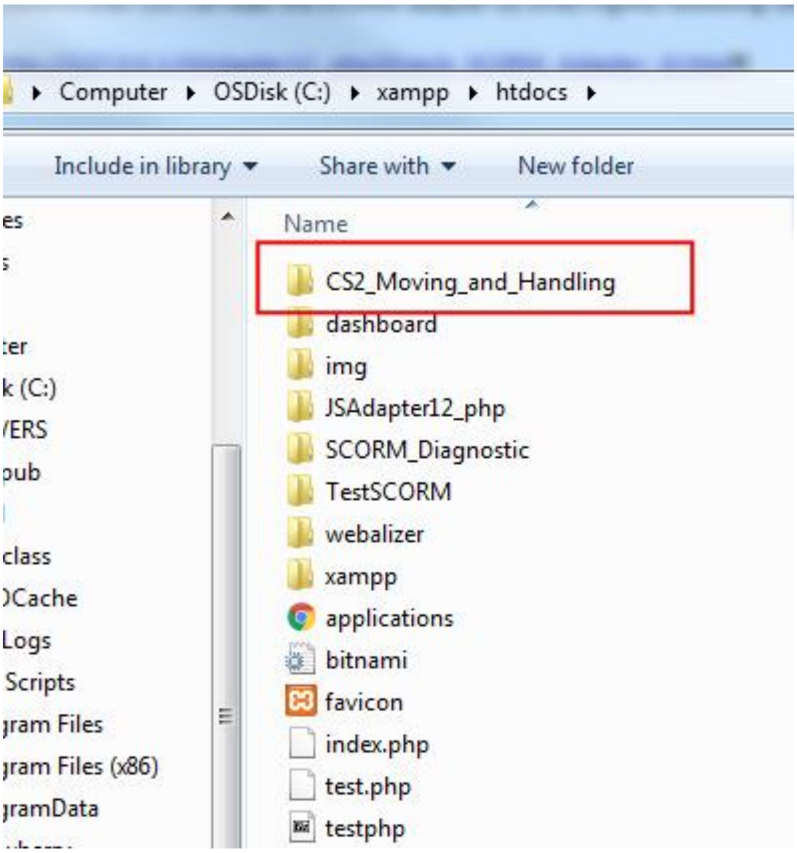
No Content To Display

No starting URL was specified. You must add a query parameter called `starting_url`. For example, to show `http://www.oracle.com` in this frame, you would set the starting URL of the RCO to:

```
http://yourhost/yourpath/adapter.html?starting_url=http://www.oracle.com
```

10.2 Upload Content to Local Server

1. Publish your content as you would if uploading to a web server.
2. Copy the content folder to the C:\xampp\htdocs folder as demonstrated in below screenshot



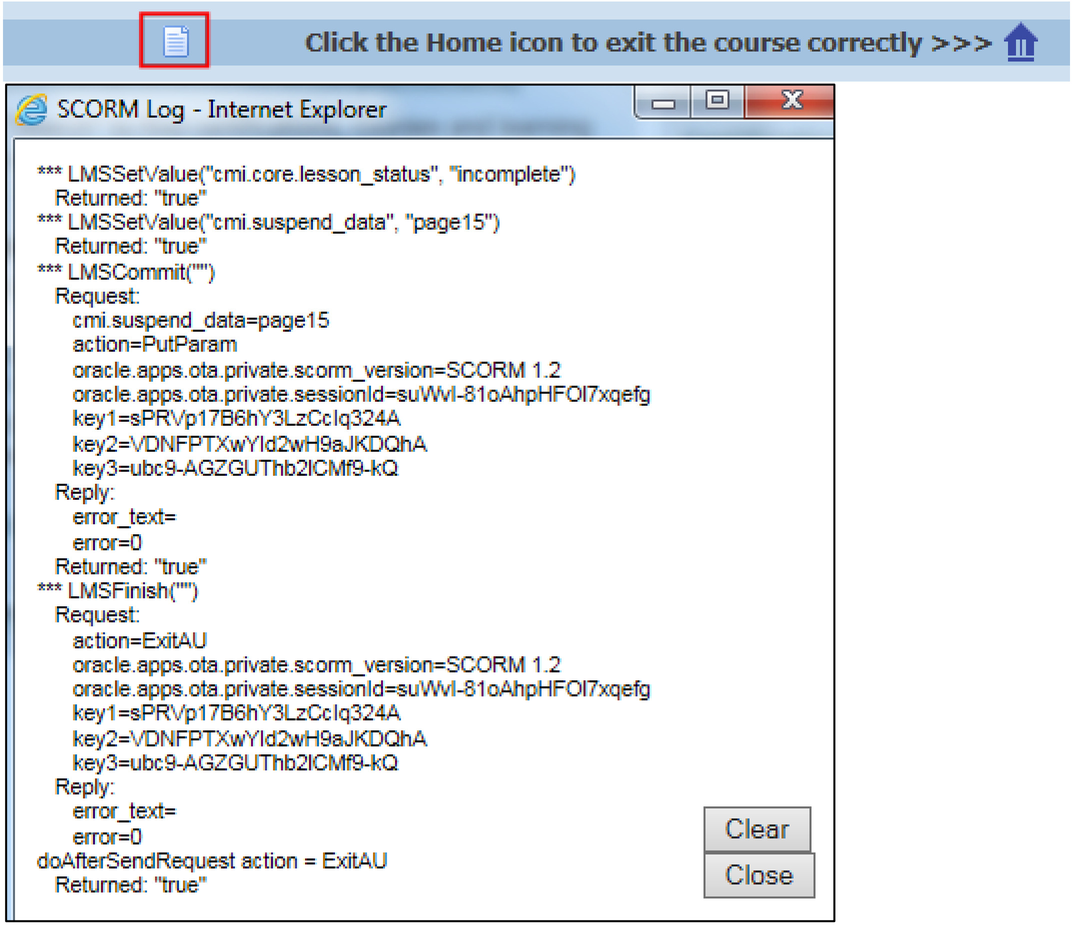
3. Confirm that the starting URL for the content opens in a browser. It should look similar to https://127.0.0.1/coursecontentfolder/course.html

Opening the file in a browser may generate some SCORM errors as you are not viewing the course within an LMS.

10.3 Set up the content in TPLY

1. Create a course in TPLY in the usual way.
2. When creating the Learning Object ensure that you use the local URL e.g.https://127.0.0.1/CS2_Moving_and_Handling/a001index.html
3. Test that the content plays as expected.

SCORM calls can be viewed by clicking on the Log button.


Click the Home icon to exit the course correctly >>>
SCORM Log - Internet Explorer
Clear
Close

```
*** LMSSetValue("cmi.core.lesson_status", "incomplete")
  Returned: "true"
*** LMSSetValue("cmi.suspend_data", "page15")
  Returned: "true"
*** LMSCommit("")
  Request:
    cmi.suspend_data=page15
    action=PutParam
    oracle.apps.ota.private.scorm_version=SCORM 1.2
    oracle.apps.ota.private.sessionId=suWvl-81oAhpHFOl7xqefg
    key1=sPRVp17B6hY3LzCclq324A
    key2=VDNFPTXwYld2wH9aJKDQhA
    key3=ubc9-AGZGUThb2lCMf9-kQ
  Reply:
    error_text=
    error=0
  Returned: "true"
*** LMSFinish("")
  Request:
    action=ExitAU
    oracle.apps.ota.private.scorm_version=SCORM 1.2
    oracle.apps.ota.private.sessionId=suWvl-81oAhpHFOl7xqefg
    key1=sPRVp17B6hY3LzCclq324A
    key2=VDNFPTXwYld2wH9aJKDQhA
    key3=ubc9-AGZGUThb2lCMf9-kQ
  Reply:
    error_text=
    error=0
doAfterSendRequest action = ExitAU
  Returned: "true"
```

11 APPENDIX 4 - USING ARTICULATE PRODUCTS WITH ESR

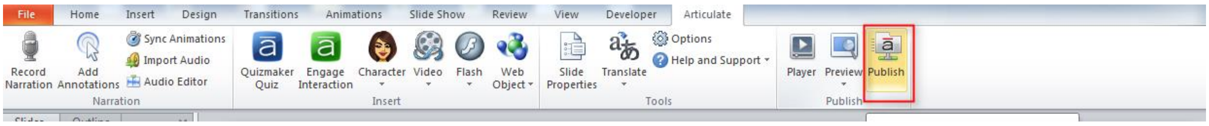
11.1 Overview

This section details the settings in Articulate Presenter that have been found to work correctly when content is played through ESR. This document refers to Articulate Presenter ‘13, however similar settings may work correctly in other versions of Articulate Presenter or Articulate Storyline.

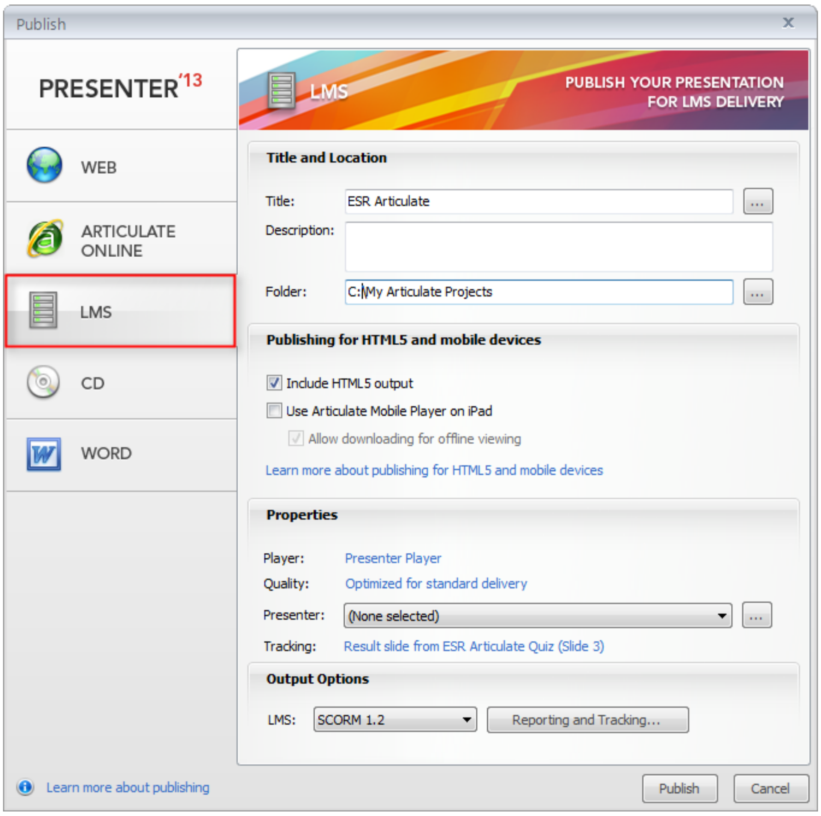
11.2 Publish Settings

When your course design is complete:

1. Select Publish from the ribbon



2. Select LMS


Publish
PRESENTER'13
WEB
ARTICULATE ONLINE
LMS
CD
WORD
LMS
PUBLISH YOUR PRESENTATION FOR LMS DELIVERY
Learn more about publishing
Publish
Cancel
Reporting and Tracking...

Title and Location

Title: ESR Articulate

Description:

Folder: C:\My Articulate Projects

Publishing for HTML5 and mobile devices

☑ Include HTML5 output

☐ Use Articulate Mobile Player on iPad

☑ Allow downloading for offline viewing

Learn more about publishing for HTML5 and mobile devices

Properties

Player: Presenter Player

Quality: Optimized for standard delivery

Presenter: (None selected)

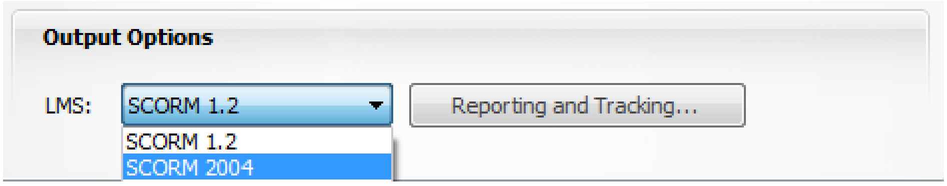
Tracking: Result slide from ESR Articulate Quiz (Slide 3)

Output Options

LMS: SCORM 1.2

Output Options for publication are available at the bottom of the form

3. For ESR you can select SCORM 1.2



Output Options

LMS: SCORM 1.2

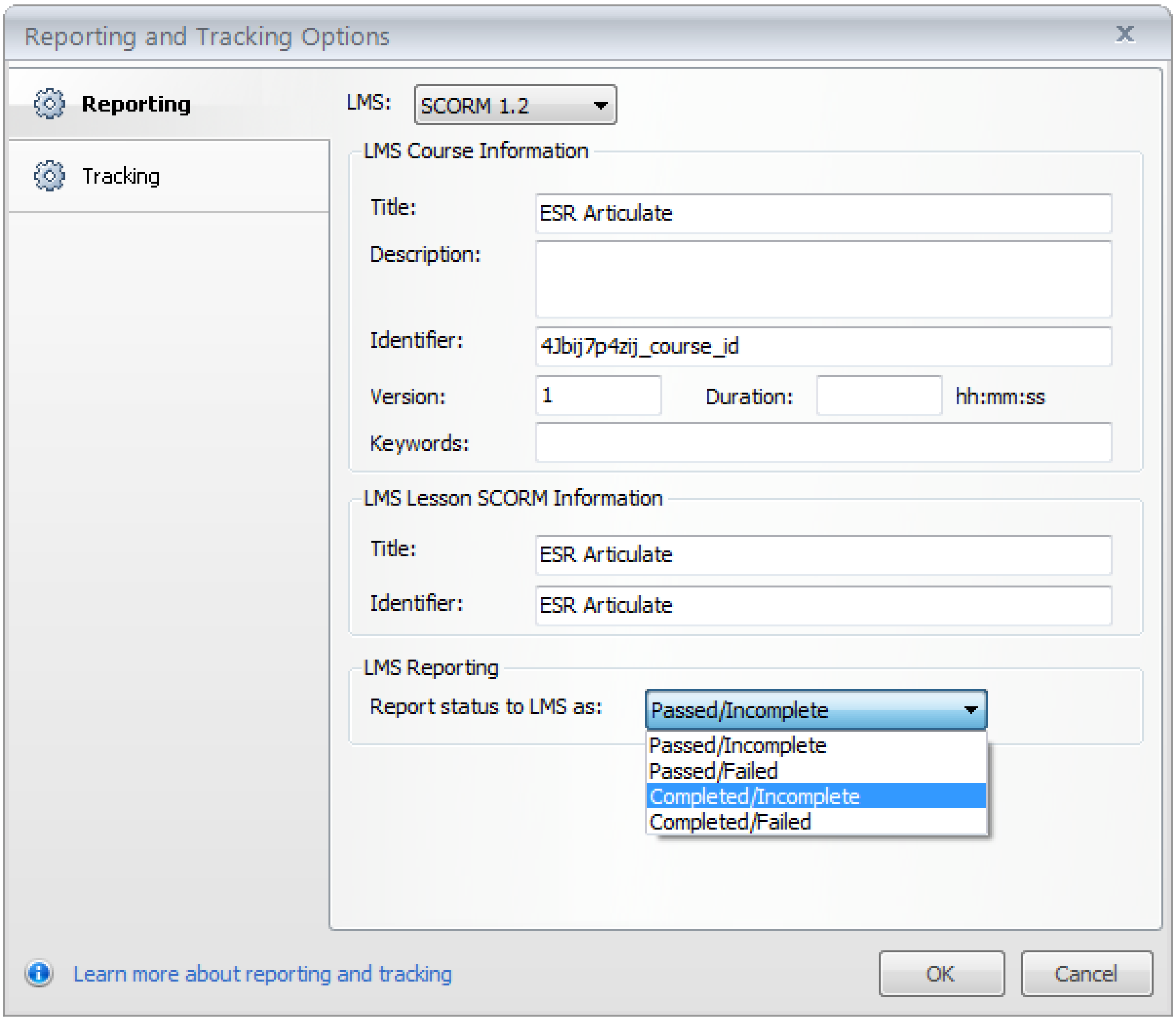
4. Select Reporting and Tracking to fine tune how the course communicates with the LMS.

11.3 Reporting

LMS Reporting will give be the status that a learner receives on their record.

- Passed/Incomplete
- Passed/Failed
- Completed/Incomplete
- Completed/Failed

  Use Completed/Incomplete if your course does not have a quiz.
  Use Passed/Failed if your course has a graded assessment.



Reporting and Tracking Options

LMS: SCORM 1.2

LMS Course Information

Title: ESR Articulate

Description:

Identifier: 4Jbij7p4zij_course_id

Version: 1 Duration: hh:mm:ss

Keywords:

LMS Lesson SCORM Information

Title: ESR Articulate

Identifier: ESR Articulate

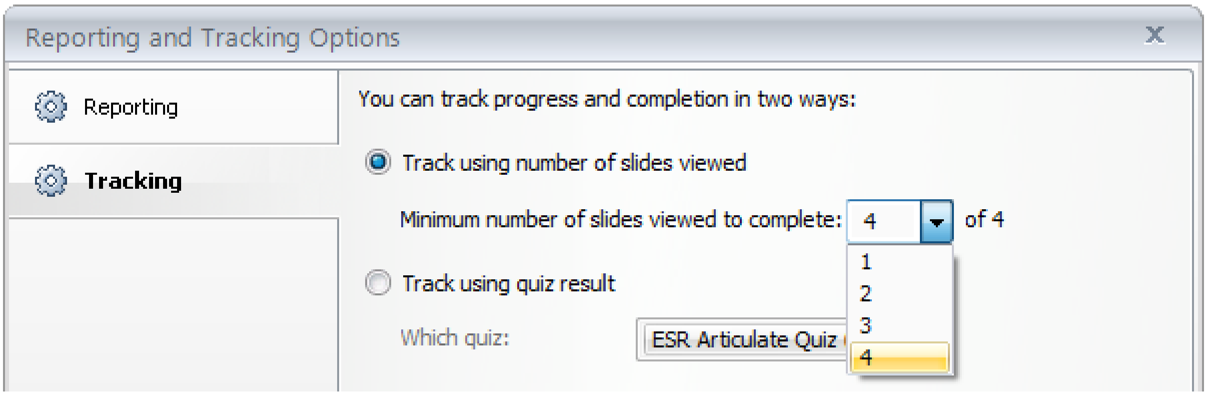
LMS Reporting

Report status to LMS as: Passed/Incomplete

11.4 TRACKING

Tracking is used for changing the settings on how you wish to track progress and completion.

This can be based on the number of slides viewed or based on a quiz result within the course.



Reporting and Tracking Options

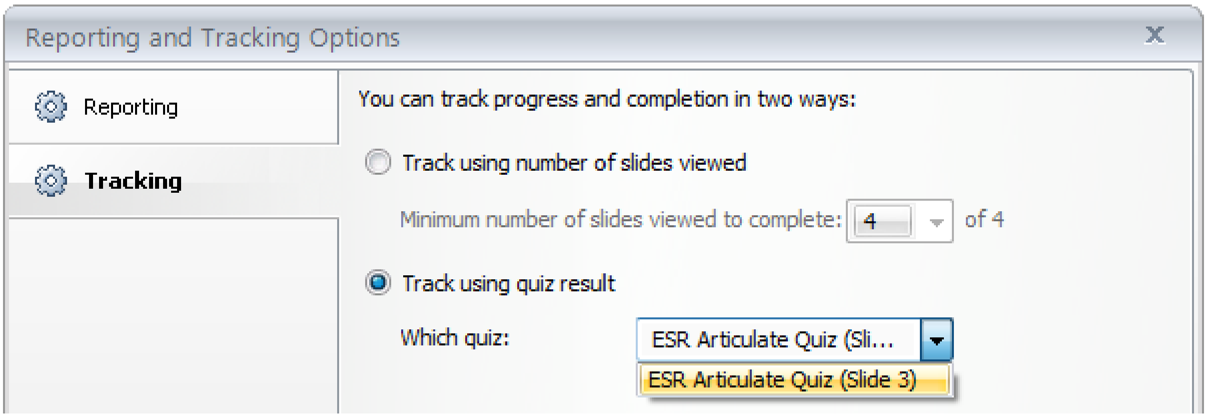
You can track progress and completion in two ways:

● Track using number of slides viewed

Minimum number of slides viewed to complete: 4 of 4

○ Track using quiz result

Which quiz: ESR Articulate Quiz



Reporting and Tracking Options

You can track progress and completion in two ways:

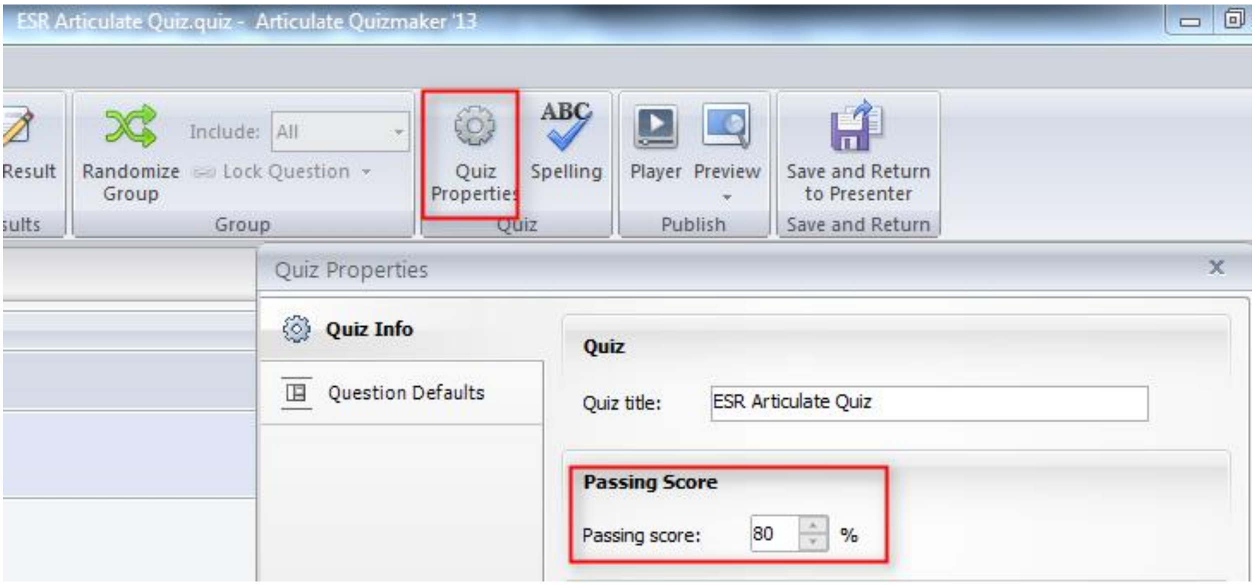
○ Track using number of slides viewed

Minimum number of slides viewed to complete: 4 of 4

● Track using quiz result

Which quiz: ESR Articulate Quiz (Sli...

If you are using a quiz then the settings are set up in the Quiz properties.



ESR Articulate Quiz.quiz - Articulate Quizmaker '13

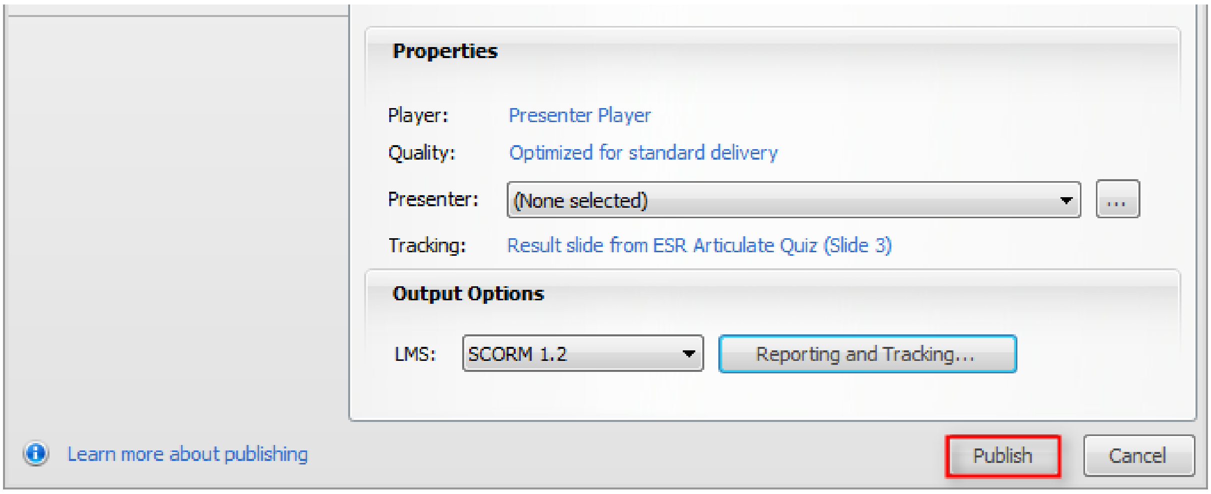
Quiz

Quiz title: ESR Articulate Quiz

Passing Score

Passing score: 80 %

- When you are happy with the settings, click OK
- Click Publish



Properties

Player: Presenter Player

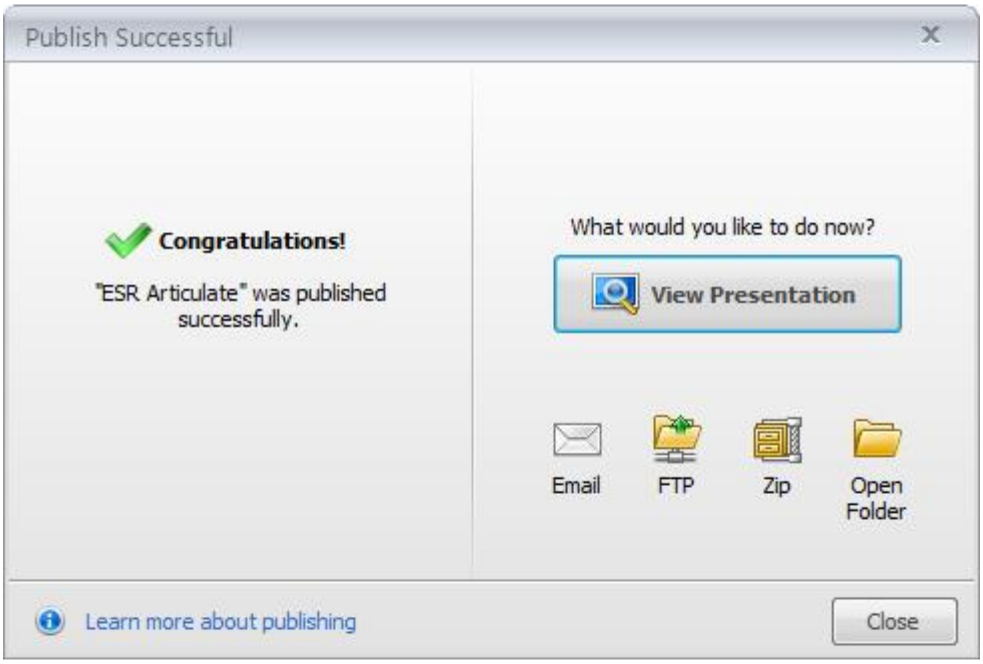
Quality: Optimized for standard delivery

Presenter: (None selected)

Tracking: Result slide from ESR Articulate Quiz (Slide 3)

Output Options

LMS: SCORM 1.2



Publish Successful

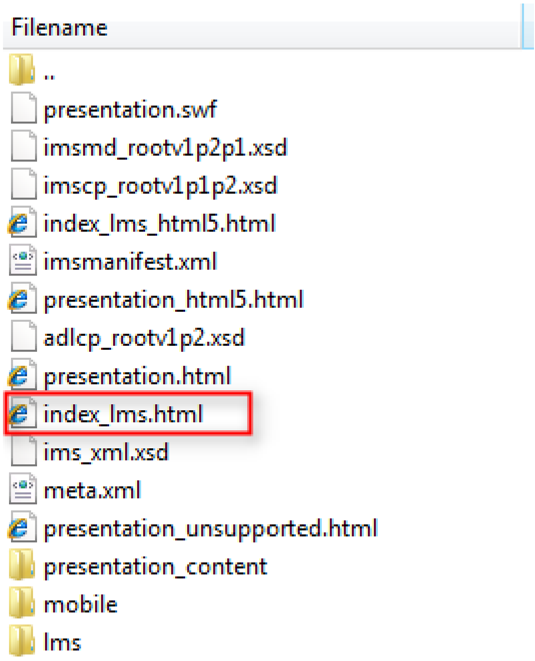
Congratulations!

"ESR Articulate" was published successfully.

What would you like to do now?

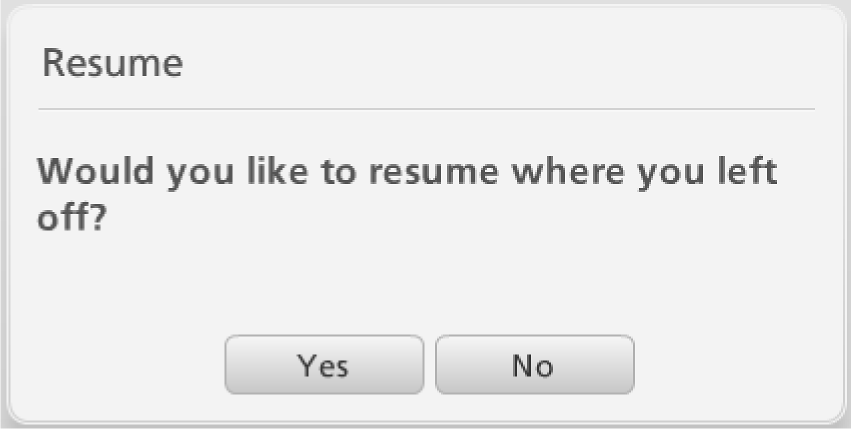
- Copy the unzipped files to your web server.

The file index_lms.html is the starting file for you course.
E.g. in the Learning Object URL field in ESR you would enter:
https://www.myserver.com/elearning/course/index_lms.html



Filename
..
presentation.swf
imsmd_rootv1p2p1.xsd
imscp_rootv1p1p2.xsd
index_lms_html5.html
imsmanifest.xml
presentation_html5.html
adlcp_rootv1p2.xsd
presentation.html
index_lms.html
ims_xml.xsd
meta.xml
presentation_unsupported.html
presentation_content
mobile
lms

11.5 Bookmarking - Prompt to Resume

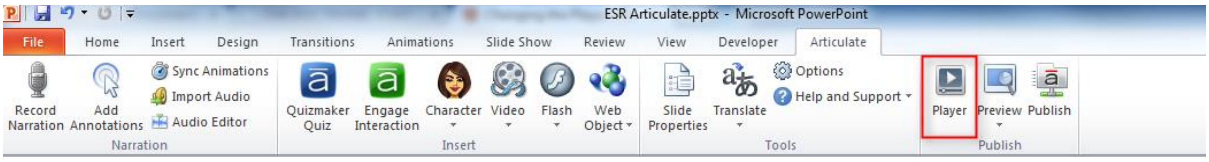


Resume

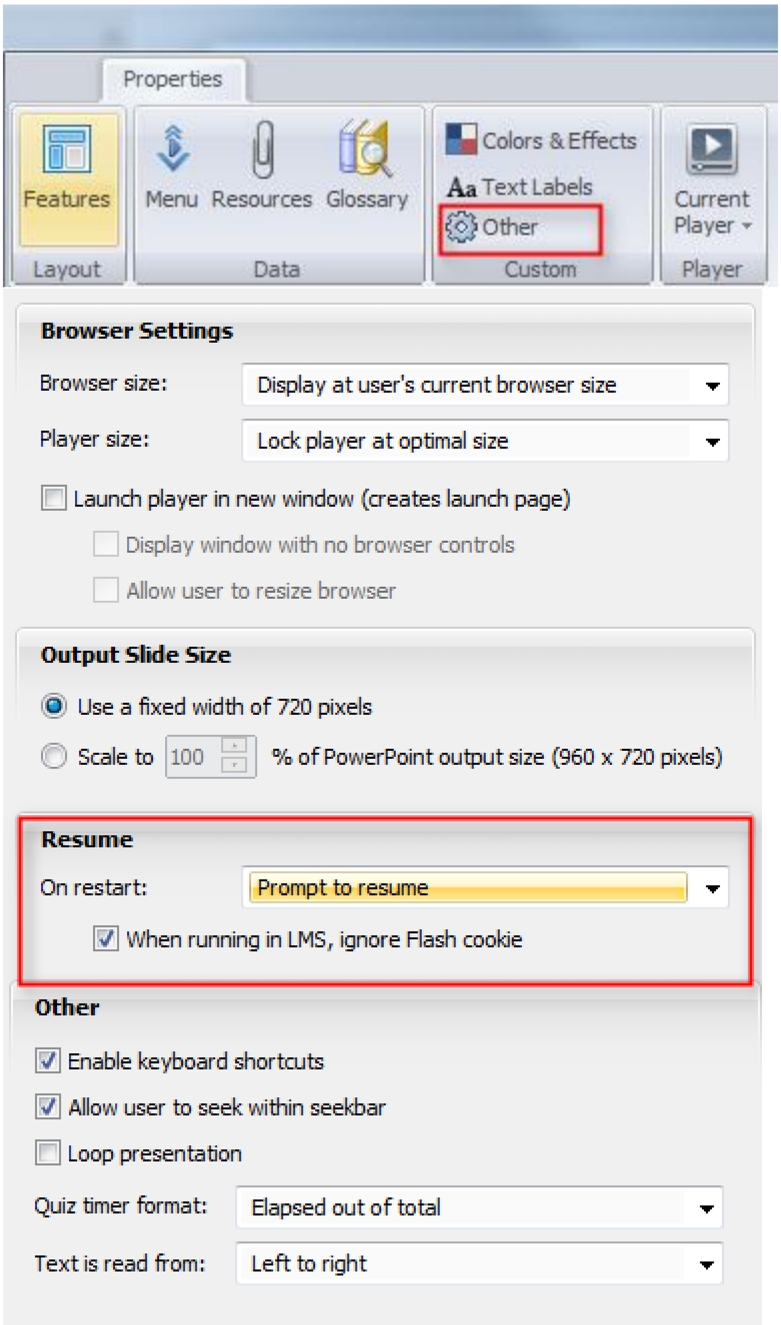
Would you like to resume where you left off?

You can change the behaviour for how a course starts up if a learner exits the course and returns to it at a later time.
This is set in the Player options (note - this option is also available in the Player options for Engage Interactions and Quizmaker Quizzes).

1. Select Player



2. Select Other



Browser Settings

Browser size: Display at user's current browser size

Player size: Lock player at optimal size

□ Launch player in new window (creates launch page)

□ Display window with no browser controls

□ Allow user to resize browser

Output Slide Size

● Use a fixed width of 720 pixels

○ Scale to 100 % of PowerPoint output size (960 x 720 pixels)

Resume

On restart: Prompt to resume

☑ When running in LMS, ignore Flash cookie

Other

☑ Enable keyboard shortcuts

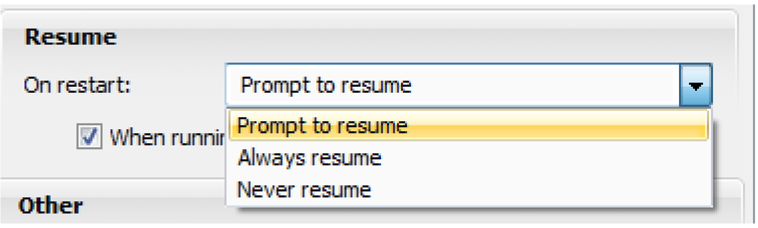
☑ Allow user to seek within seekbar

□ Loop presentation

Quiz timer format: Elapsed out of total

Text is read from: Left to right

You can select from three options



Resume

On restart: Prompt to resume

☑ When runnir

Other

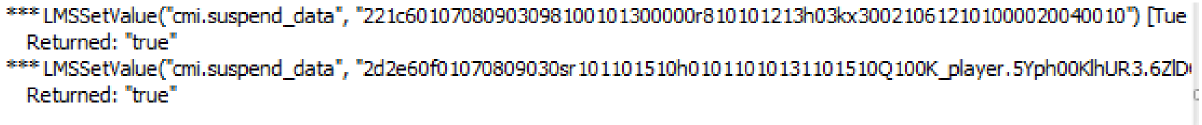
1. Prompt to resume - the course will display the following prompt whenever a learner returns

Note - if the Learner selects No then it will reset all previous interactions with the course.

2. Always resume - the course will always resume from the last place that the learner reached in a course
3. Never resume - the course will always start from the beginning.

11.5.1 Exceeding SCORM Suspend Data Limits

If bookmarking is enabled (Option 1 or 2) then the ‘suspend data’ element stores the learner’s location in the course.



```
*** LMSSetValue("cmi.suspend_data", "221c60107080903098100101300000r810101213h03kx30021061210100002004001 0") [Tue
   Returned: "true"
*** LMSSetValue("cmi.suspend_data", "2d2e60f01070809030sr101101510h01011010131101510Q100K_player.5Yph00KlhUR3.6ZlD
   Returned: "true"
```

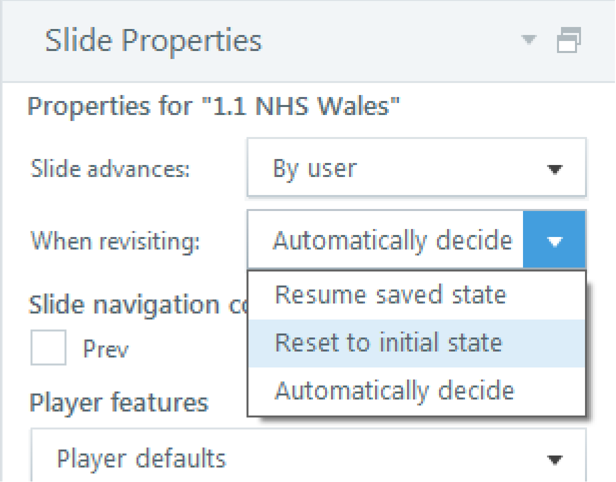
If courses are very large, or if Engage and Quiz interactions also create suspend data, then the Limit can be exceeded.
If this limit is exceeded then the course will always resume at the point where the limit was reached.
You will see a message in the console log similar to the following
LMSSetValue (“ cmi.suspend_data…)
LMS Error: 405 - Incorrect data type

11.5.2 Solutions

If you have a large course that does exceed suspend data limits, here are some suggestions for correcting it:

- Disable the resume feature in your content: Presenter, Quizmaker, Engage.
- Reduce the number of slides until the resume feature works as expected. The limit will vary, depending on a variety of factors. You'll need to test your content in your TPLY to verify.

In Articulate Storyline under Slide Properties, change the ‘When revisiting’ value from ‘Automatically decide’ to ‘Reset to initial state’.



Slide Properties

Properties for "1.1 NHS Wales"

Slide advances: By user

When revisiting: Automatically decide

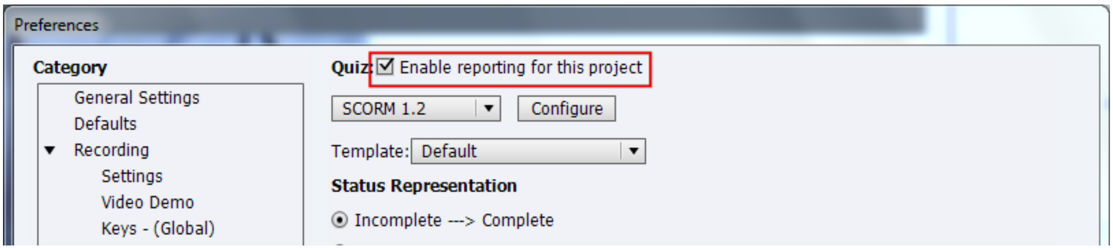
12 APPENDIX 5 - USING ADOBE CAPTIVATE WITH ESR

12.1 OVERVIEW

This document details the settings in Adobe Captivate that have been found to work correctly when content is played through ESR. This document refers to Adobe Captivate 6, however similar settings may work correctly in newer versions although the menu path may be slightly different.

12.2 PUBLISH SETTINGS

To enable SCORM reporting in a Captivate session, the ‘Enable reporting in this project’ option must be checked as detailed in the below example.



Quiz: ☑ Enable reporting for this project

SCORM 1.2

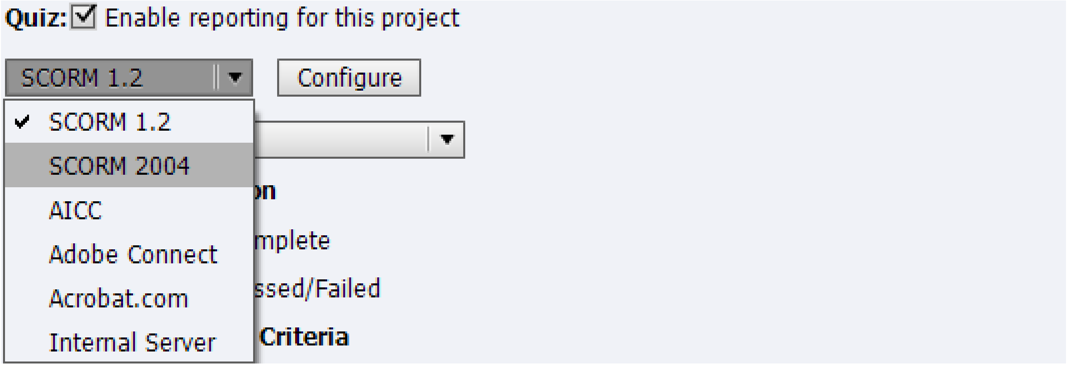
Template: Default

Status Representation

● Incomplete ---> Complete

12.3 SCORM VERSION

Select SCORM 1.2



Quiz: ☑ Enable reporting for this project

SCORM 1.2

12.4 MANIFEST

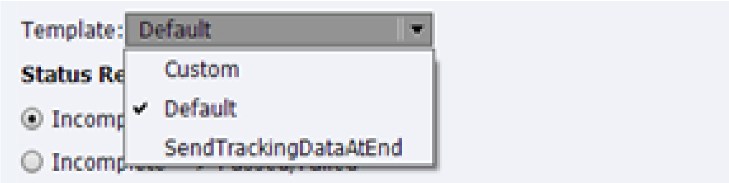
There is no need to select Configure Manifest files as these are not used in ESR. (These are normally used in systems to create learning objects, however this is done by an administrator in ESR).

12.5 TEMPLATE

The template defines how the content will send SCORM updates to the LMS.
The Default template sends interaction details about the content to the LMS at every point during the course.

The SendTrackingDataAtEnd template only sends interaction details when the user exits a course.

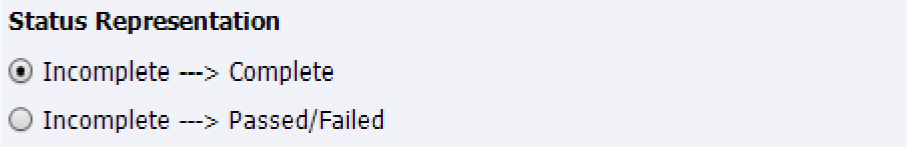
Both templates will work correctly with ESR, however if the content is generating LMS errors in your testing (visible in the log file) then it is recommended that you use the SendTrackingDataAtEnd template.



Template: Default

12.6 STATUS

This setting controls the statuses that are sent back to the LMS. This will usually be Incomplete/Complete if a course does not have an assessment, or Passed/Failed when it does.



Status Representation

● Incomplete ---> Complete

○ Incomplete ---> Passed/Failed

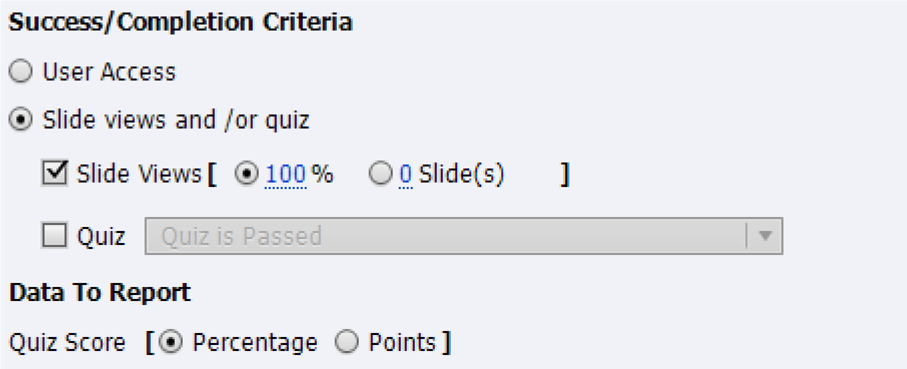
12.7 SUCCESS/COMPLETION CRITERIA

The settings here specify when the learner Status will be updated to Complete or Passed/Failed
User Access - Sets Status to completed/passed as soon as the learner views the content.
This setting would not usually be used with SCORM content in ESR as the same outcome could be achieved in ESR by setting the Learning Object tracking type to Automatic.

Slide views and / or quiz

The learner Status will be updated based on either Slide Views achieved or a Quiz passed or a combination of both.

In most cases, if your course does not have a quiz then you would update the learner status based on slide views. If your course has a quiz then the learner status would be updated based on the quiz result.



Success/Completion Criteria

○ User Access

● Slide views and /or quiz

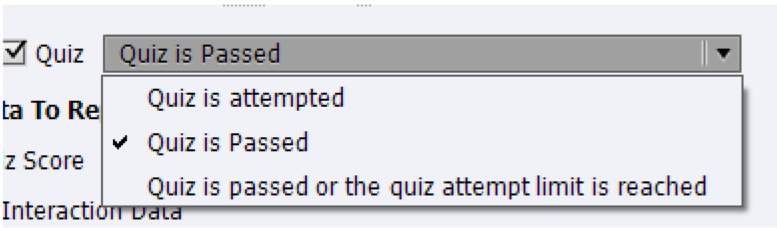
☑ Slide Views [● 100 % ○ 0 Slide(s)]

□ Quiz Quiz is Passed

Data To Report

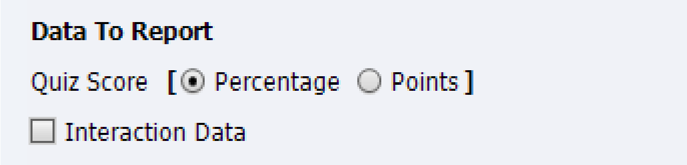
Quiz Score [● Percentage ○ Points]

Slide Views - Updates Status once user has completed enough slides (either a specified percentage or a set number of slides). If you require that the learner must view all slides before their status is updated to completed then set this to 100%.
Note - care must be used if you use this setting and your course uses branching. This is covered in further detail in the section Custom Completion Criteria.
Quiz - If you are giving the learner unlimited attempts to pass the quiz then select 'Quiz is Passed'.
If there is a limit to the number of attempts then select 'Quiz is passed or the quiz attempt limit is reached'.
The settings for your quiz are set in the 'Pass or Fail' section



☑ Quiz Quiz is Passed

Data to Report - Percentage or Points (raw score), Percentage is recommended
Interaction Data should be unticked as this is not recorded in ESR.

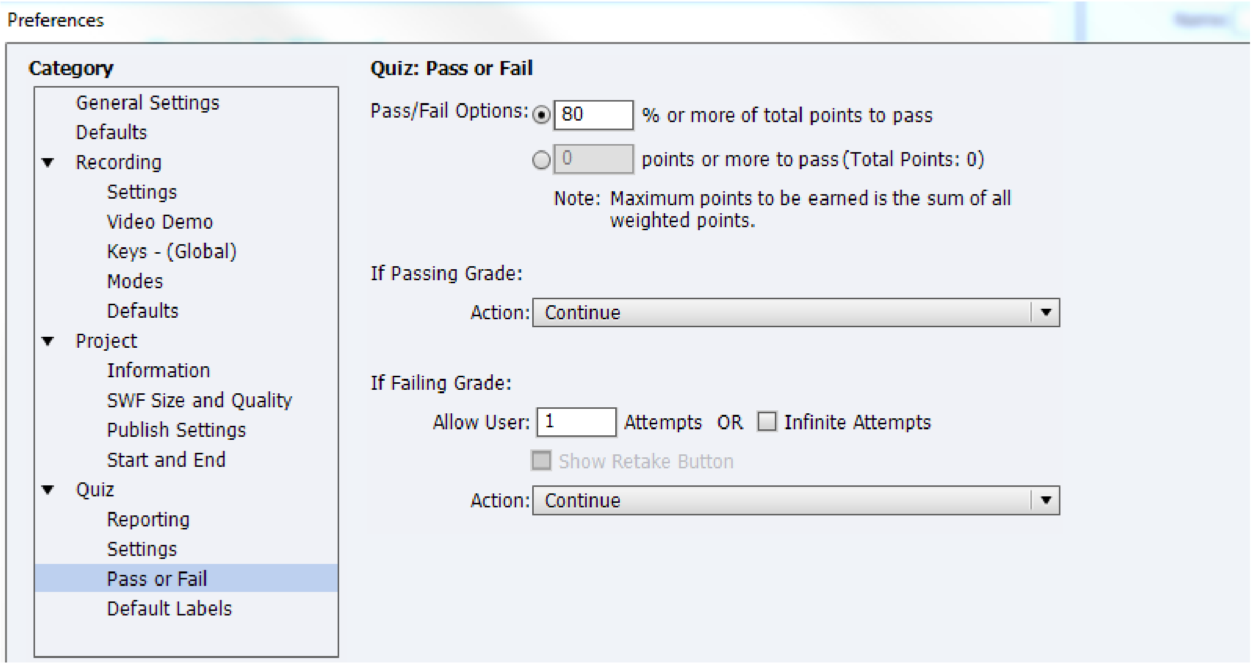


Data To Report

Quiz Score [● Percentage ○ Points]

□ Interaction Data

Pass orFail options.
Either set the passing score in Captivate or on the Learning Object in ESR. Do not use both/
Pass/Fail Options - It is recommended that you use a percentage score for your pass mark.



Quiz: Pass or Fail

Pass/Fail Options: ● 80 % or more of total points to pass

○ 0 points or more to pass (Total Points: 0)

Note: Maximum points to be earned is the sum of all weighted points.

If Passing Grade:

Action: Continue

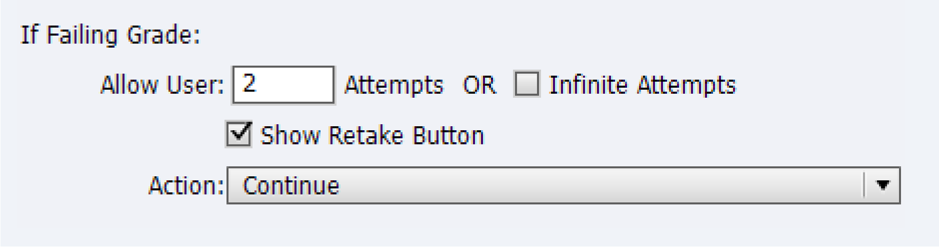
If Failing Grade:

Allow User: 1 Attempts OR □ Infinite Attempts

□ Show Retake Button

Action: Continue

If the number of attempts at the quiz is greater than 1 or the Infinite Attempts check box is ticked then it is recommended that the Show Retake Button check box is ticked. This button will take the learner to the beginning of the quiz.



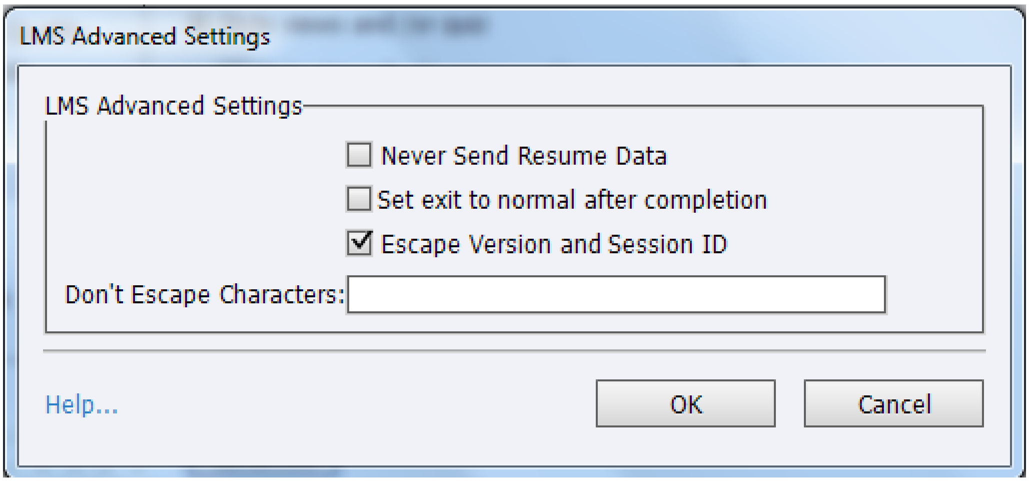
If Failing Grade:

Allow User: 2 Attempts OR □ Infinite Attempts

☑ Show Retake Button

Action: Continue

12.8 LMS Advanced Settings



LMS Advanced Settings

LMS Advanced Settings

□ Never Send Resume Data

□ Set exit to normal after completion

☑ Escape Version and Session ID

Don't Escape Characters:

Never Send Resume Data
Check this box to turn off bookmarking i.e. the content will not remember where the user last played up to

Set exit to normal after completion

Tick this box if you want learners to be able to view a course from the beginning once it has been successfully completed. This is not necessary if your course is to be used as part of a certification.

Ticking this box does not mean that a learner will update their course completion date if they view the course at a later date.

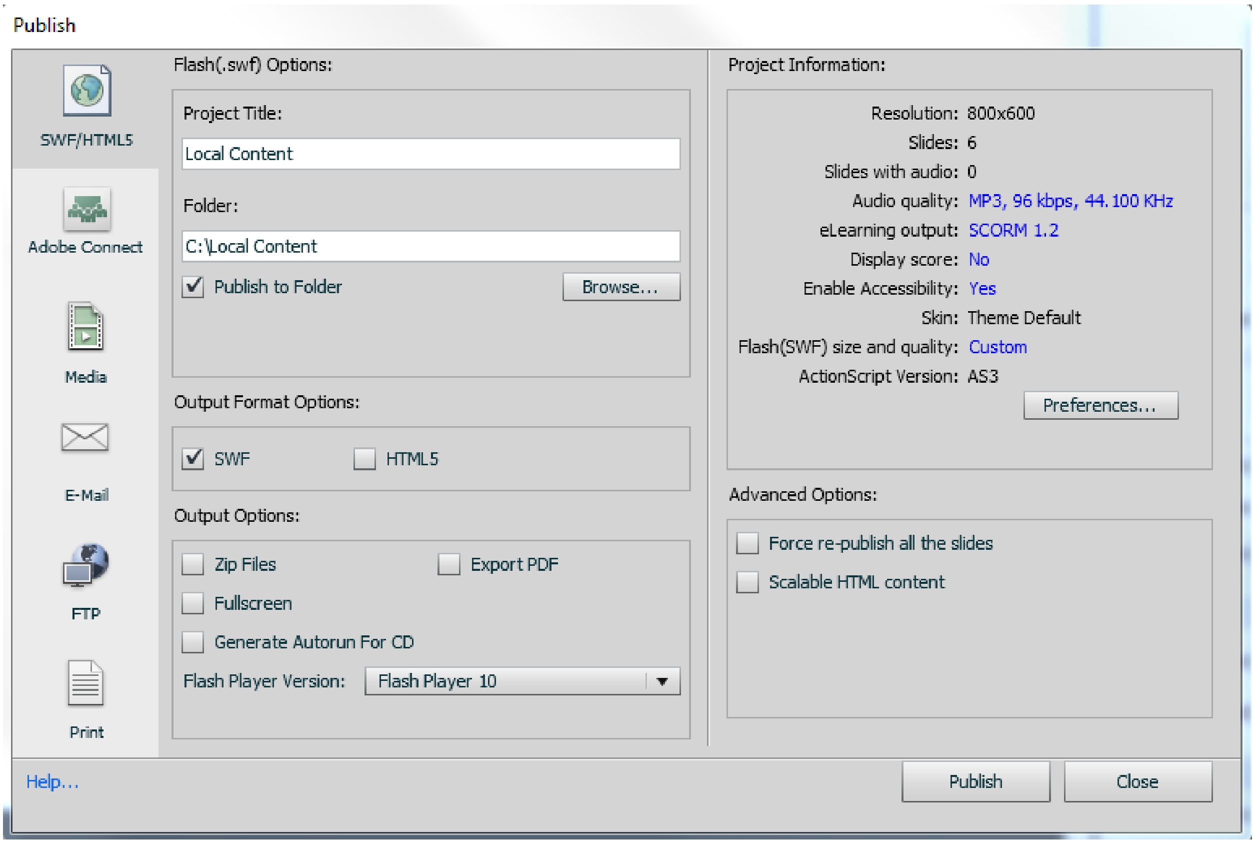
Escape Version and Session ID

This setting is not used in ESR.

12.9 Publish Course

Once you have set up your Publish settings

- Select File - Publish
- Ensure that the eLearning output in the Project Information states SCORM 1.2.
- Enter your Project Title and the location where you wish to Publish.
- Select Publish to Folder and ensure that Zip Files is unchecked
- You may choose HTML 5 as well as SWF (Flash) in the Output Format Options, this will ensure that you content can be played on devices other than PCs.
- If you check HTML 5, you can check Scalable HTML content in the Advanced Options so that the content will scale to the size of the screen that is being used to view the content.



Flash(.swf) Options:

Project Title: Local Content

Folder: C:\Local Content

☑ Publish to Folder

Output Format Options:

☑ SWF ☐ HTML5

Output Options:

☐ Zip Files ☐ Export PDF

☐ Fullscreen

☐ Generate Autorun For CD

Flash Player Version: Flash Player 10

Project Information:

Resolution: 800x600
Slides: 6
Slides with audio: 0
Audio quality: MP3, 96 kbps, 44.100 KHz
eLearning output: SCORM 1.2
Display score: No
Enable Accessibility: Yes
Skin: Theme Default
Flash(SWF) size and quality: Custom
ActionScript Version: AS3

Advanced Options:

☐ Force re-publish all the slides

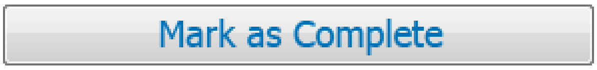
☐ Scalable HTML content

12.10 Custom Completion Criteria

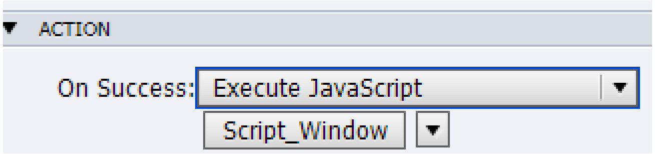
If you are using Slide Views as Completion Criteria and you set Slides viewed to 100%, if your course has any branching then a user will not be able to complete the course as they will not view all the slides.

You can overcome this by adding a button to a course and adding code to the button to mark the course as complete.

1. Add a button to a page in your course (usually the last slide, but it can be added anywhere)



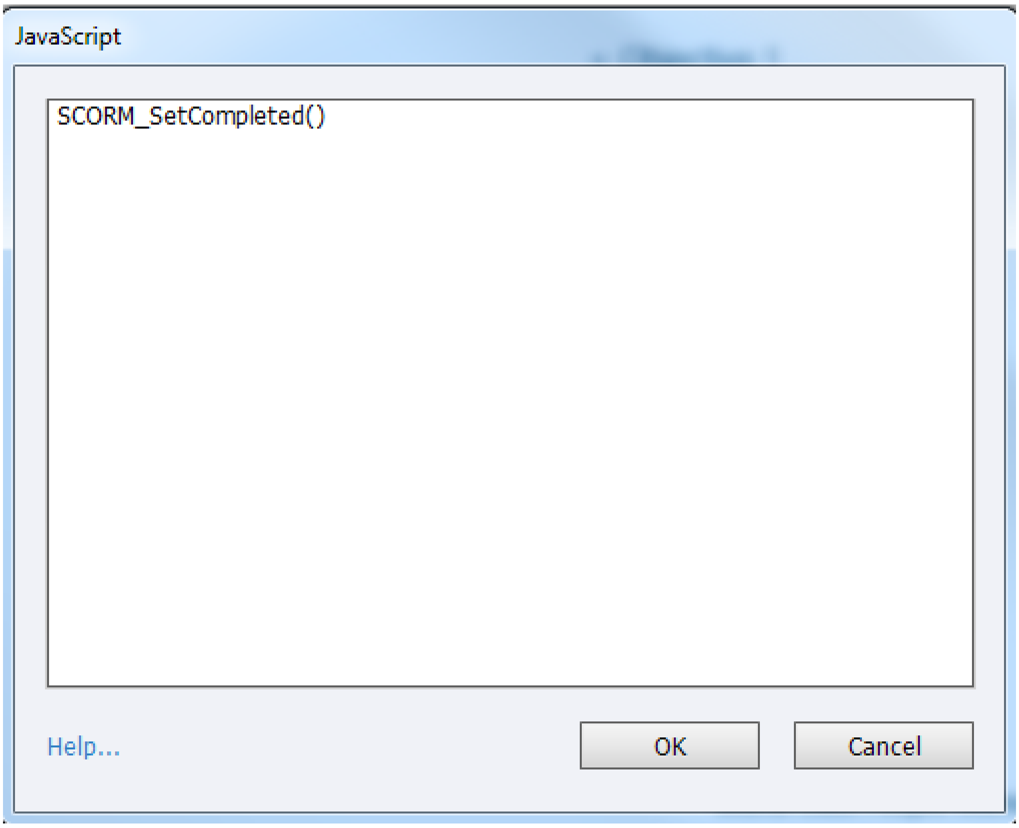
2. In the Action section - select Execute JavaScript from the On Success drop-down



On Success: Execute JavaScript

3. Click Script Window
4. Enter SCORM_SetCompleted() in the JavaScript window

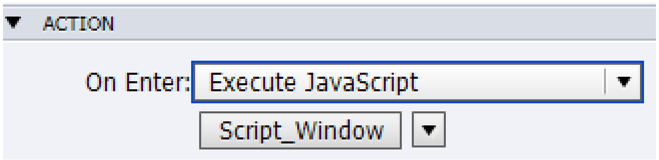
This is the function that is already built into Captivate to change the course status to complete and would usually be triggered by the Success/Completion Criteria.



```
SCORM_SetCompleted()
```

5. Click OK.
6. Save changes and publish your course.

You can also achieve the same result by adding to the JavaScript to the Action Properties for a Slide.
If the same code is added to the On Enter action, when a user navigates to the slide, the course will automatically be marked as complete.



On Enter: Execute JavaScript

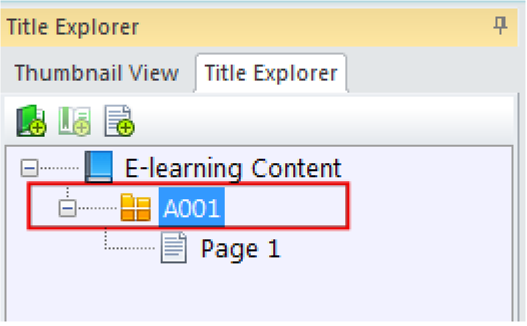
13 APPENDIX 6 - USING LECTORA WITH ESR

13.1 Overview

This section details the settings in Lectora Publisher that have been found to work correctly when content is played through ESR. Normally SCORM standards mean that content can be played in any LMS however Some versions of Lectora publish content in a way that does not work correctly in ESR due to security rules present in all modern browsers. This document refers to Lectora Publisher version 11; however similar settings may work correctly in other versions of Lectora Publisher or Lectora Inspire.

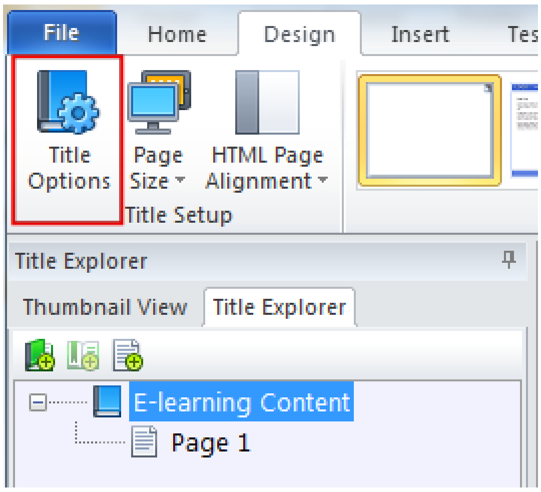
13.2 Initial Set up

A course created in Lectora must contain an Assignable Unit, similar to that shown in the screenshot below, in order to be SCORM compliant.

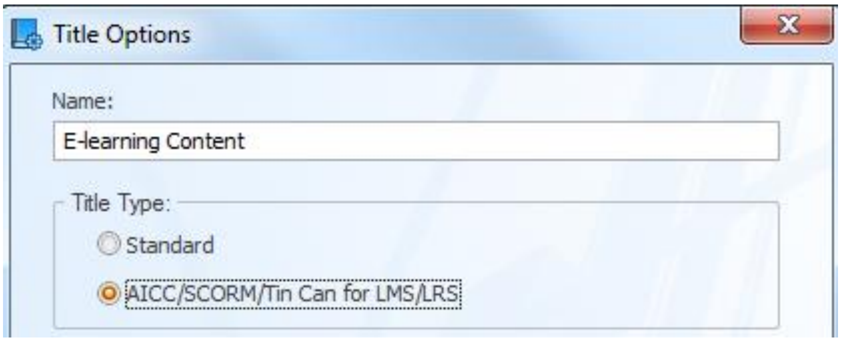


If your course does not contain an Assignable Unit then follow the steps below, if it does then move to the next section - Add Required Actions.

1) Select Design - Title Options



2) Change the Title Type to AICC/SCORM/Tin Can for LMS/LRS

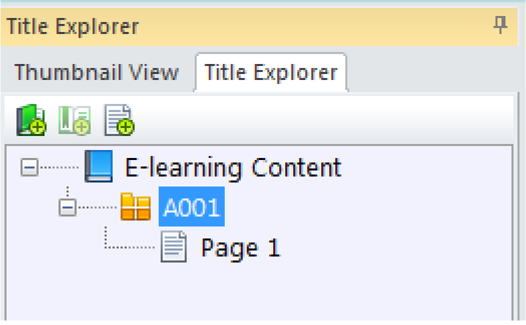

Title Options

Name:
E-learning Content

Title Type:
○ Standard
● AICC/SCORM/Tin Can for LMS/LRS

3) Click OK

Your Title Explorer will now look similar to the screen shot below.



13.3 Add Required Actions

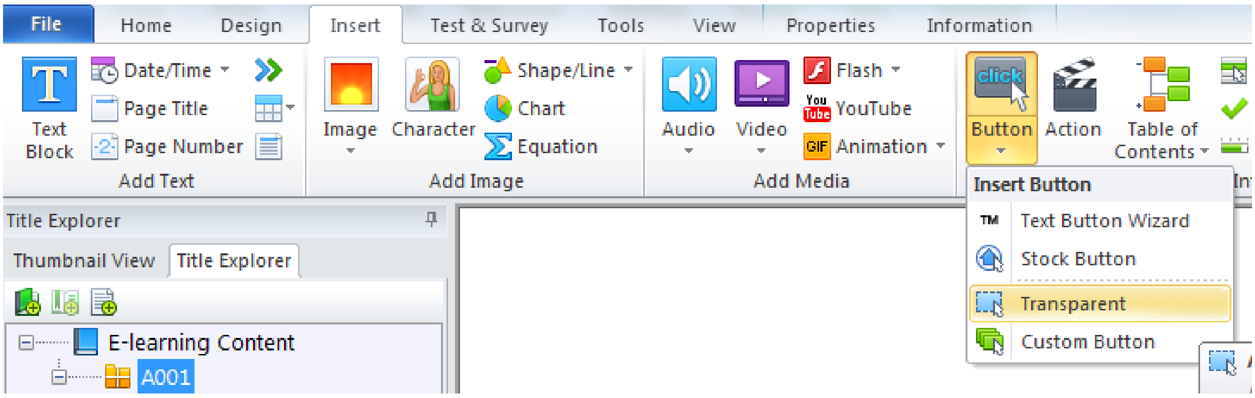
In Lectora, a SCORM compliant course must have an ‘Exit Title’ action and an action that modifies either/or both of the variables AICC_Lesson_Status or AICC_Score.

Add an exit button

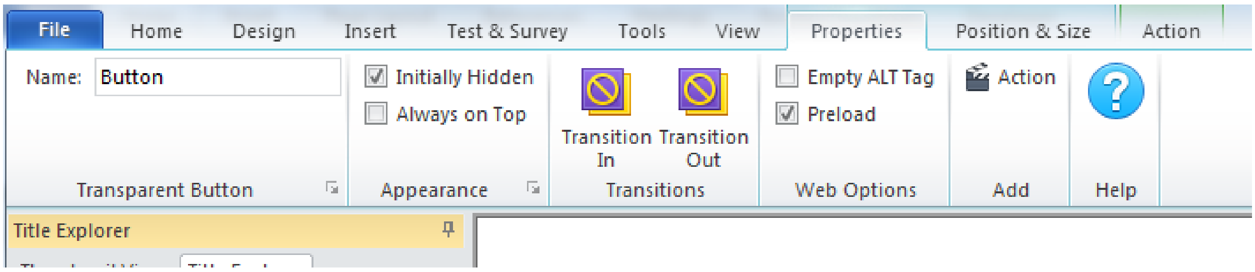
ESR already has an exit button to leave a course, so although we must add the action to the Lectora course we do not want the learner to click it.

This can be achieved by adding the Exit action to an invisible button.

1. Add a Transparent button to A001



2. Ensure the button is initially hidden


Name: Button

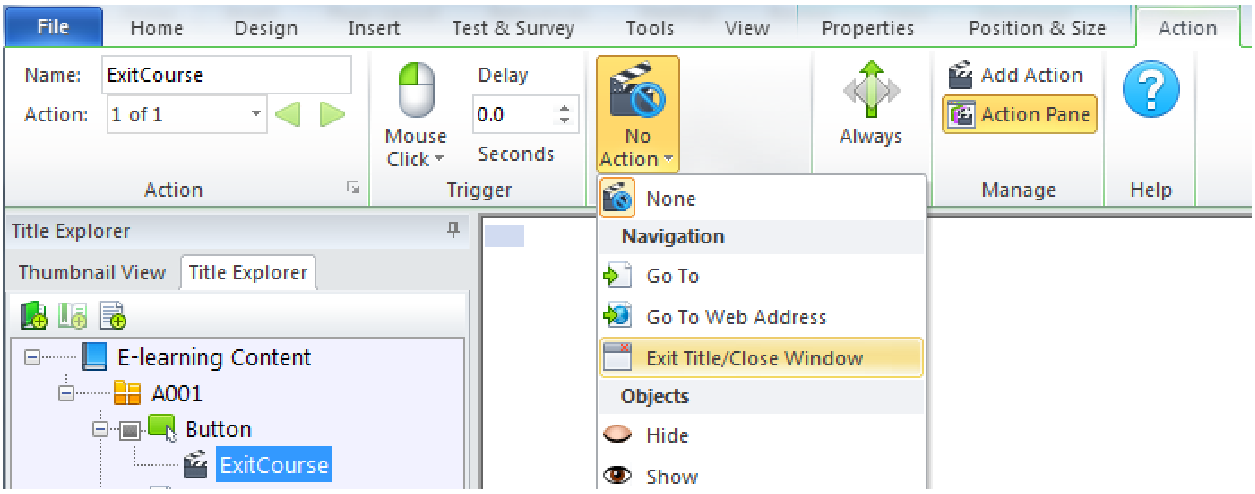
☑ Initially Hidden

☐ Always on Top

☐ Empty ALT Tag

☑ Preload

3. Add the Action Exit Title/Close Window to the button.


Name: ExitCourse

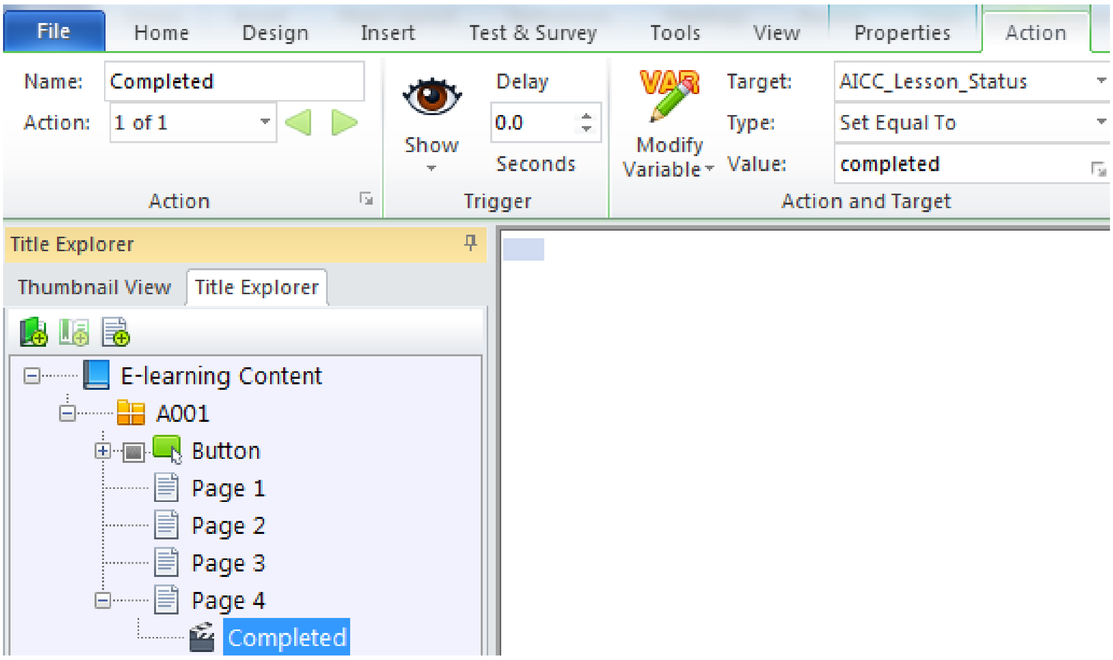
Action: 1 of 1

Delay: 0.0 Seconds

13.4 SETTINGS WITH NO ASSESSMENT

If your course does not have an assessment then you can add an action to your course that sets the course to completed on the last page of the course.

1. Select Modify Variable for action type.
2. Select the target variable of AICC_Lesson_Status
3. Set Equal To ‘completed’ (all lower case)


Name: Completed

Action: 1 of 1

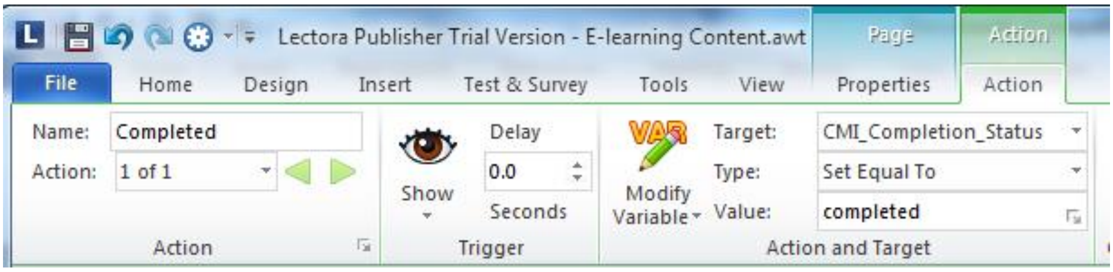
Delay: 0.0 Seconds

Target: AICC_Lesson_Status

Type: Set Equal To

Value: completed

If your course is SCORM 2004 compliant then you will also have to set a value for CMI_Completion_Status.



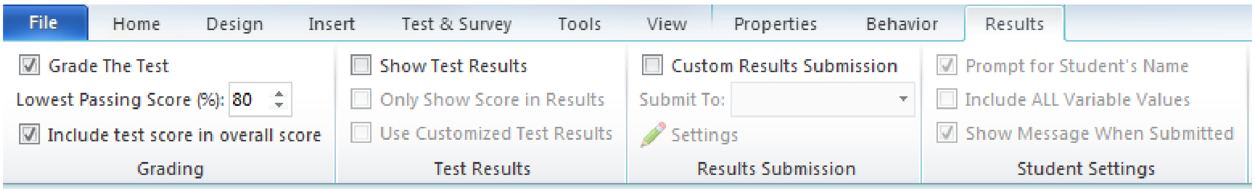
Name: Completed
Action: 1 of 1
Delay 0.0 Seconds
Target: CMI_Completion_Status
Type: Set Equal To
Value: completed

13.5 SETTINGS WITH AN ASSESSMENT

To start, your course must have at least one test chapter. This will ensure that the AICC_Score variable is given a value on completion of the test.

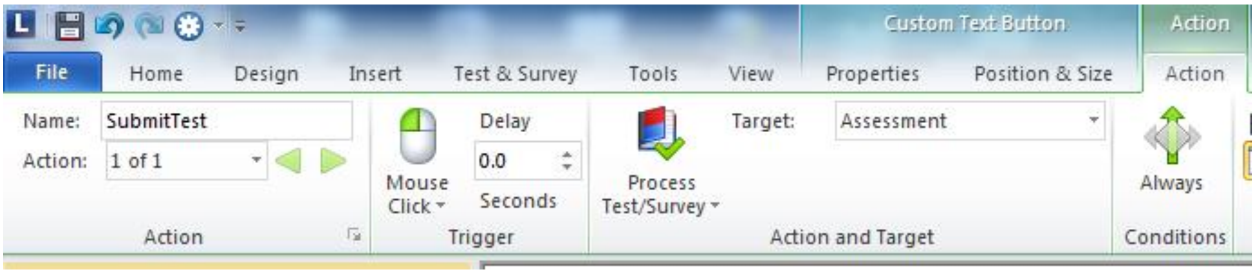
1. Uncheck ‘Include test score in overall score’ if do not want the test to contribute towards the final score.

In this example - we only have one test in the course, the result of this test will be the value assigned to the variable, AICC_Score.
If we had more than one test then the value assigned to AICC_Score would be the average of all the test scores included in the overall score.



☑ Grade The Test
Lowest Passing Score (%): 80
☑ Include test score in overall score
□ Show Test Results
□ Only Show Score in Results
□ Use Customized Test Results
□ Custom Results Submission
Submit To:
Settings
☑ Prompt for Student's Name
□ Include ALL Variable Values
☑ Show Message When Submitted

2. Create an action to Submit/Process the test (usually on the last page of the test).

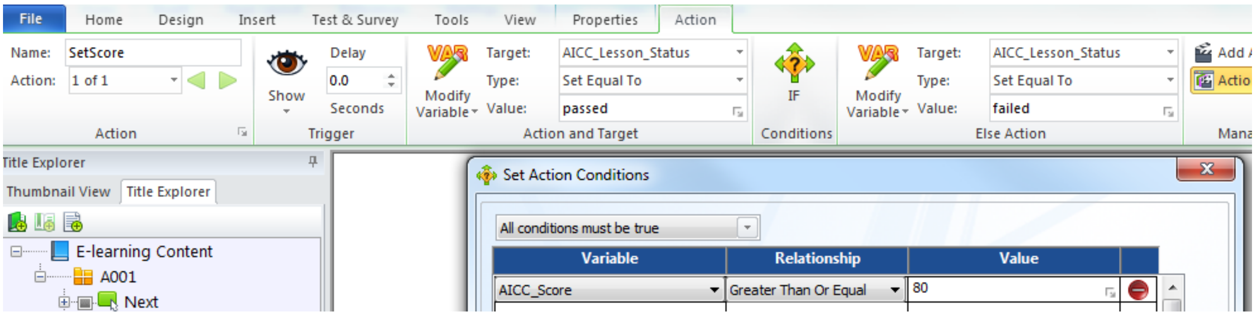


Name: SubmitTest
Action: 1 of 1
Delay 0.0 Seconds
Target: Assessment

Update the Lesson Status

3. Create an action to update the lesson status based on the result of the test, usually on the last page in your course.

Here we are changing the course status to ‘Passed’ if the AICC_Score is greater than 80.
If the score is less than 80 then the lesson status will be set to ‘Failed’



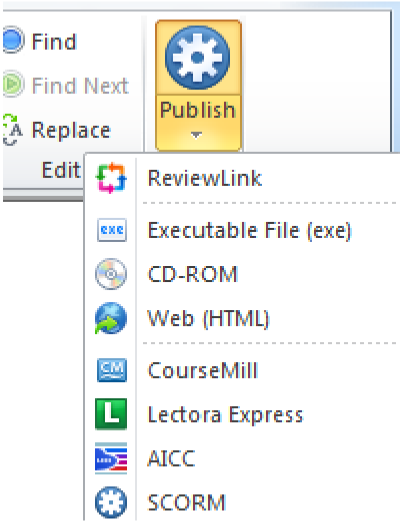
Name: SetScore
Action: 1 of 1
Delay 0.0 Seconds
Target: AICC_Lesson_Status
Type: Set Equal To
Value: passed
Target: AICC_Lesson_Status
Type: Set Equal To
Value: failed

All conditions must be true

Variable	Relationship	Value
AICC_Score	Greater Than Or Equal	80

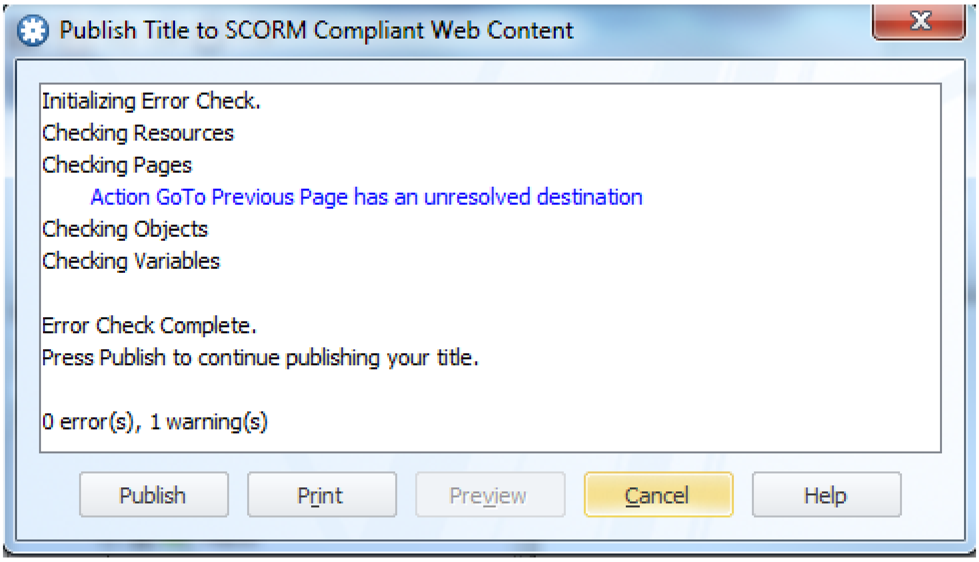
13.6 PUBLISH YOUR COURSE

1. Select Publish -SCORM



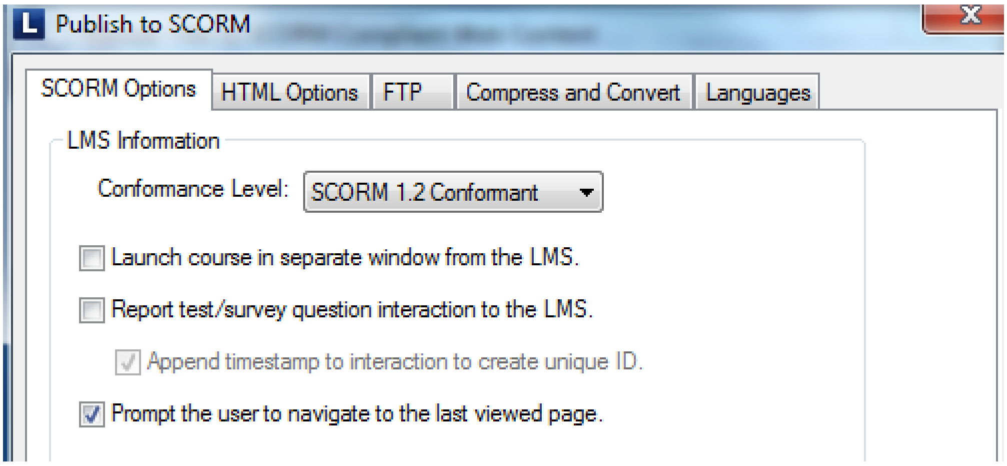
A dialogue box will be displayed.
Errors will be shown in red and must be corrected.
Warnings will be shown in blue and can be corrected or ignored.

2. Click Publish



```
Initializing Error Check.
Checking Resources
Checking Pages
    Action GoTo Previous Page has an unresolved destination
Checking Objects
Checking Variables

Error Check Complete.
Press Publish to continue publishing your title.

0 error(s), 1 warning(s)
```

3. Select SCORM 1.2 Conformant from the Conformance Level drop-down.



Conformance Level: SCORM 1.2 Conformant

☐ Launch course in separate window from the LMS.

☐ Report test/survey question interaction to the LMS.

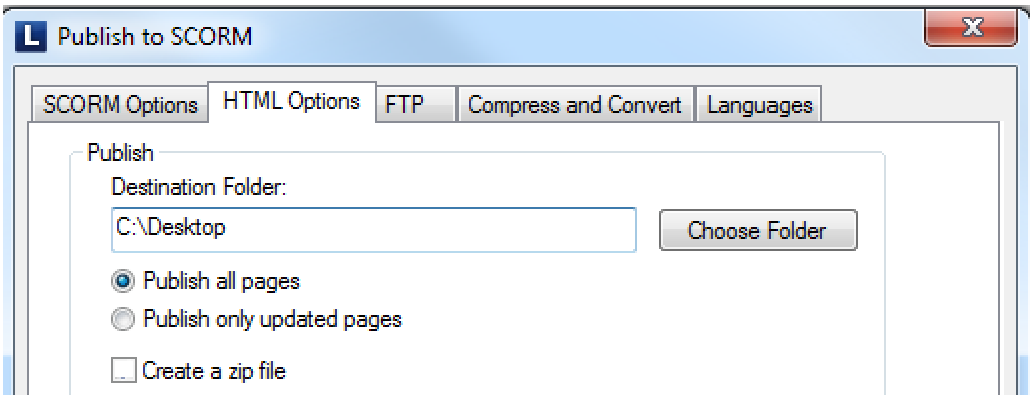
☑ Append timestamp to interaction to create unique ID.

☑ Prompt the user to navigate to the last viewed page.

‘Launch course in separate window from LMS’ should never be ticked.

‘Report test/survey question interaction to the LMS’ should not be ticked as these calls are not supported by SCORM 1.2 in ESR. (These calls are supported by SCORM 2004 but would not be stored in ESR).
Uncheck the ‘Prompt the user to navigate .to the last viewed page’ check box if you do not wish to use bookmarking. This option is checked by default.

4. Select HTML Options to choose the Destination Folder
5. Untick the create zip file as e-Learning content used with ESR is copied directly to a server.



Publish

Destination Folder: C:\Desktop

● Publish all pages

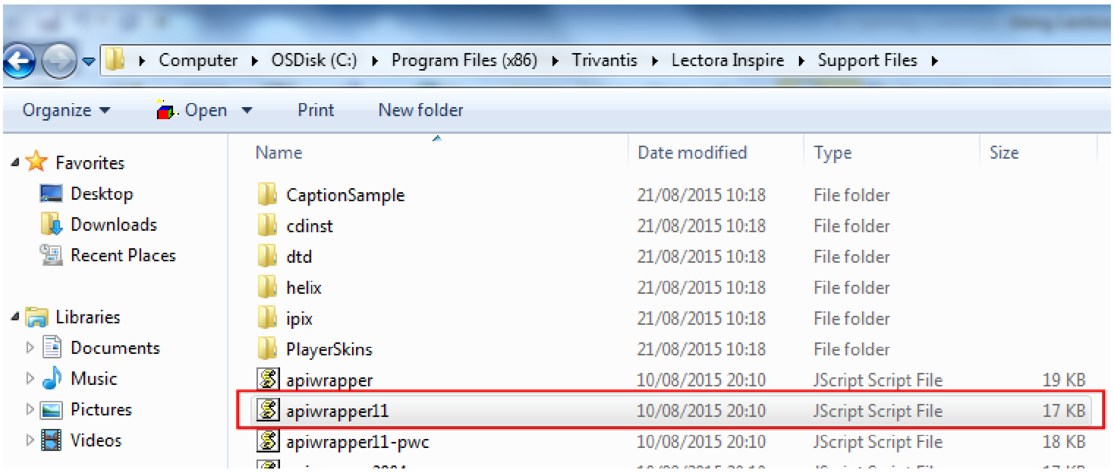
○ Publish only updated pages

□ Create a zip file

6. Click OK

13.7 Updates to Lectora Files

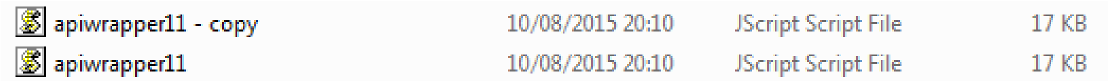
There are configurations in apiwrapper11.js file that mean that courses may not load correctly in ESR when using Internet Explorer 11, Firefox or Chrome.
Below is a workaround that will enable the courses to play correctly. This is only required where there are references to the top window.
The changes require editing of the above file, the screenshots below were produced using Notepad++, but can be made in any text editing software such as Notepad.
Navigate to the following directory on the computer where Lectora is installed.
C:\Program Files\Trivantis\Lectora Inspire\Support Files
Find the file apiwrapper11.js



Computer ▸ OSDisk (C:) ▸ Program Files (x86) ▸ Trivantis ▸ Lectora Inspire ▸ Support Files ▸

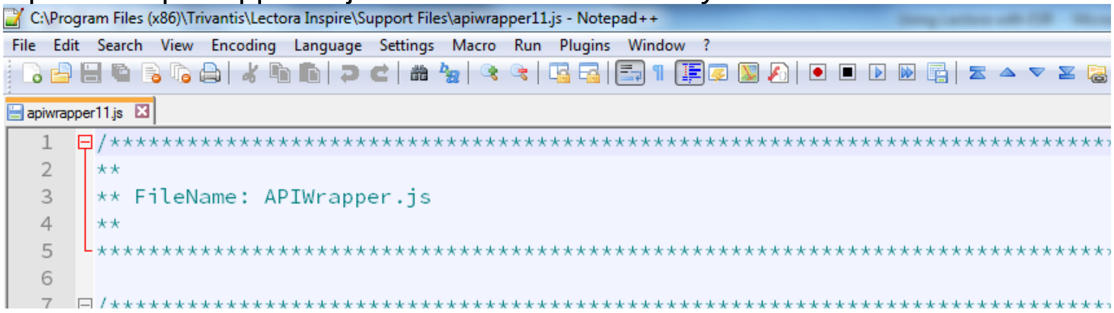
Name	Date modified	Type	Size
CaptionSample	21/08/2015 10:18	File folder	
cdinst	21/08/2015 10:18	File folder	
dtd	21/08/2015 10:18	File folder	
helix	21/08/2015 10:18	File folder	
ipix	21/08/2015 10:18	File folder	
PlayerSkins	21/08/2015 10:18	File folder	
apiwrapper	10/08/2015 20:10	JScript Script File	19 KB
apiwrapper11	10/08/2015 20:10	JScript Script File	17 KB
apiwrapper11-pwc	10/08/2015 20:10	JScript Script File	18 KB

Make a copy of the file and rename, it is good practice to make copies of any source program file that you make changes to.



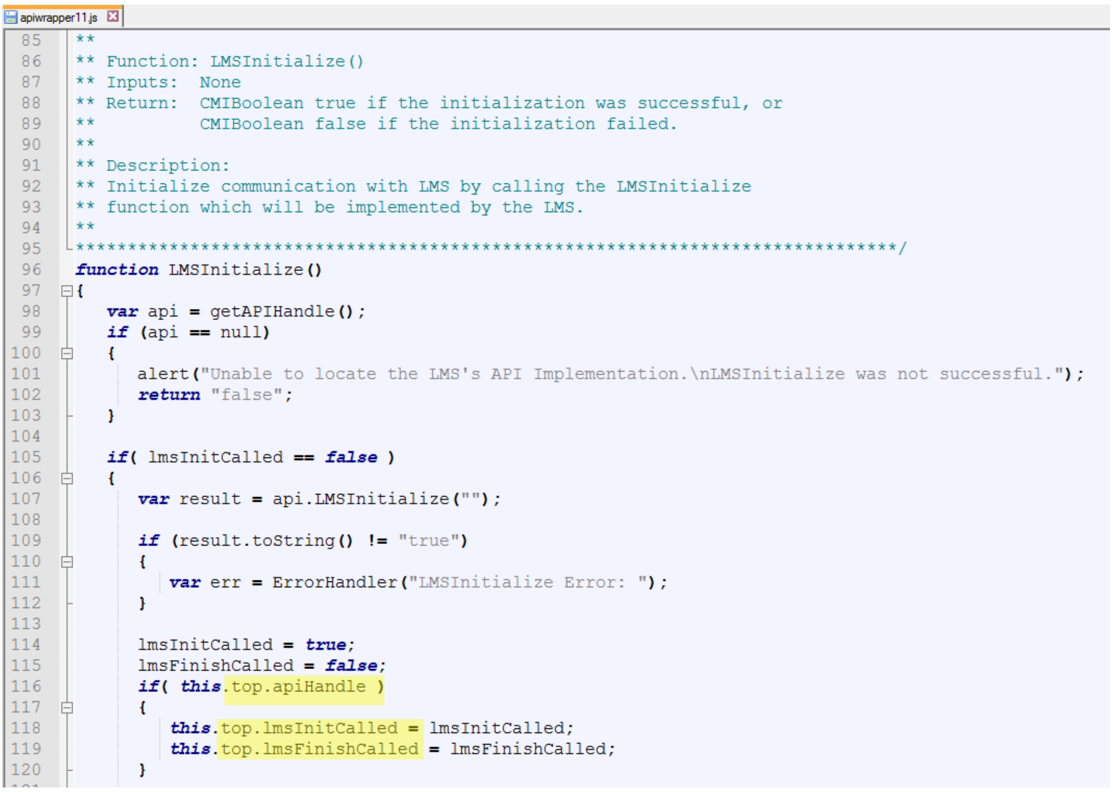
apiwrapper11 - copy	10/08/2015 20:10	JScript Script File	17 KB
apiwrapper11	10/08/2015 20:10	JScript Script File	17 KB

Open the apiwrapper11.js file in the text editor of your choice.


C:\Program Files (x86)\Trivantis\Lectora Inspire\Support Files\apiwrapper11.js - Notepad++
File
Edit
Search
View
Encoding
Language
Settings
Macro
Run
Plugins
Window
?
apiwrapper11.js

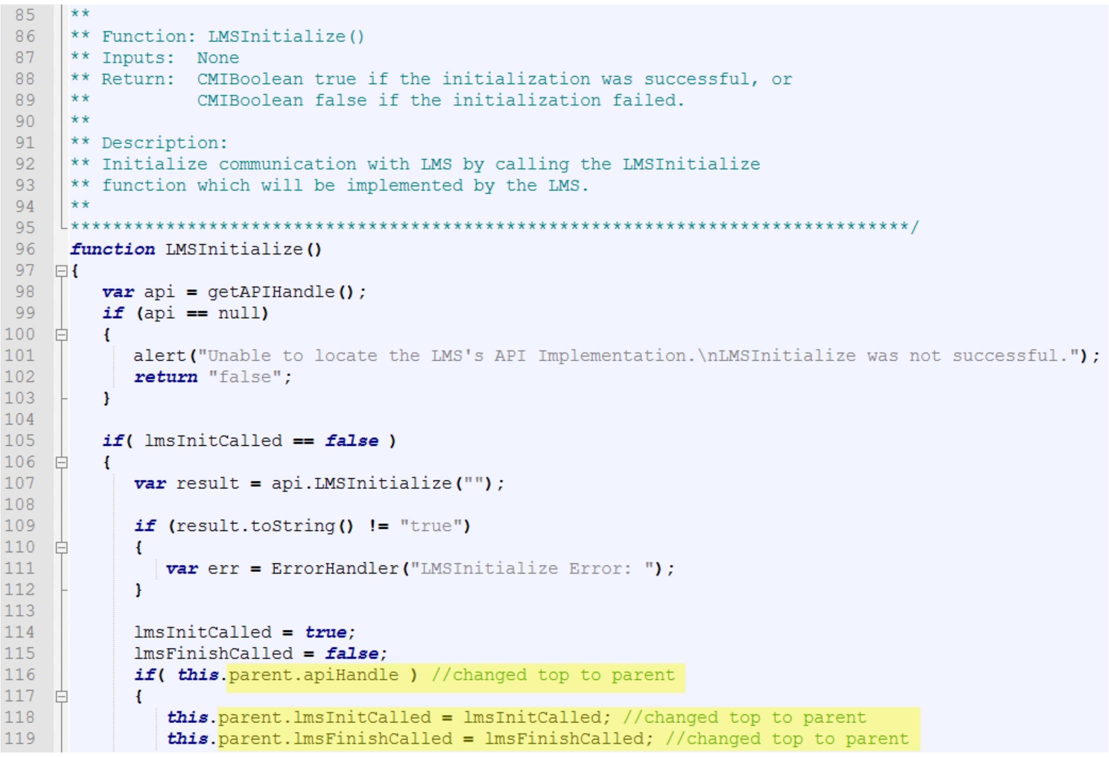
```
1  /*************************************************************************************
2  **
3  ** FileName: APIWrapper.js
4  **
5  *************************************************************************************
6
7  /*************************************************************************************
```

Navigate to the LMSInitialize() section (Starts at line 84)
On lines 116, 118 and 119 change the replace “top” with “parent”



```
85   **
86   ** Function: LMSInitialize()
87   ** Inputs:  None
88   ** Return:  CMIBoolean true if the initialization was successful, or
89   **          CMIBoolean false if the initialization failed.
90   **
91   ** Description:
92   ** Initialize communication with LMS by calling the LMSInitialize
93   ** function which will be implemented by the LMS.
94   **
95   ******************************************************************************/
96   function LMSInitialize()
97   {
98      var api = getAPIHandle();
99      if (api == null)
100     {
101        alert("Unable to locate the LMS's API Implementation.\nLMSInitialize was not successful.");
102        return "false";
103     }
104
105     if( lmsInitCalled == false )
106     {
107        var result = api.LMSInitialize("");
108
109        if (result.toString() != "true")
110        {
111           var err = ErrorHandler("LMSInitialize Error: ");
112        }
113
114        lmsInitCalled = true;
115        lmsFinishCalled = false;
116        if( this.top.apiHandle )
117        {
118           this.top.lmsInitCalled = lmsInitCalled;
119           this.top.lmsFinishCalled = lmsFinishCalled;
120        }
```

Becomes

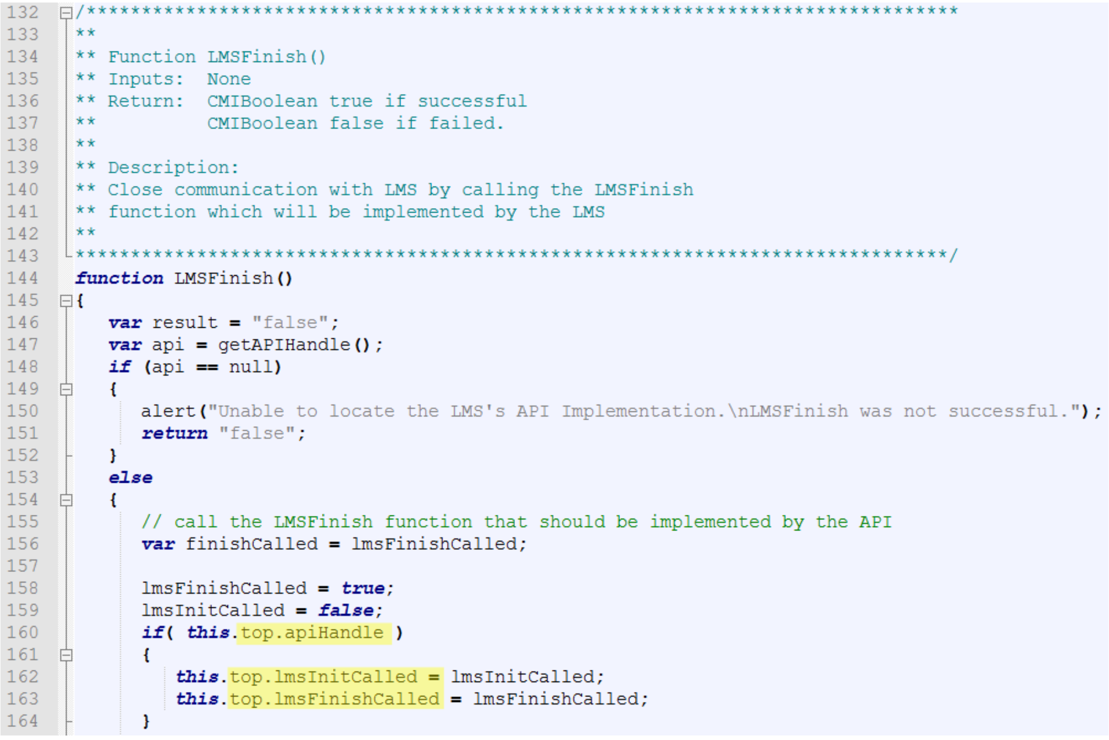


```
**
** Function: LMSInitialize()
** Inputs:  None
** Return:  CMIBoolean true if the initialization was successful, or
**          CMIBoolean false if the initialization failed.
**
** Description:
** Initialize communication with LMS by calling the LMSInitialize
** function which will be implemented by the LMS.
**
*******************************************************************************/
function LMSInitialize()
{
    var api = getAPIHandle();
    if (api == null)
    {
        alert("Unable to locate the LMS's API Implementation.\nLMSInitialize was not successful.");
        return "false";
    }

    if( lmsInitCalled == false )
    {
        var result = api.LMSInitialize("");

        if (result.toString() != "true")
        {
            var err = ErrorHandler("LMSInitialize Error: ");
        }

        lmsInitCalled = true;
        lmsFinishCalled = false;
        if( this.parent.apiHandle ) //changed top to parent
        {
            this.parent.lmsInitCalled = lmsInitCalled; //changed top to parent
            this.parent.lmsFinishCalled = lmsFinishCalled; //changed top to parent
```

Note: A comment has been added to the code so that we can identify where the change has been made.

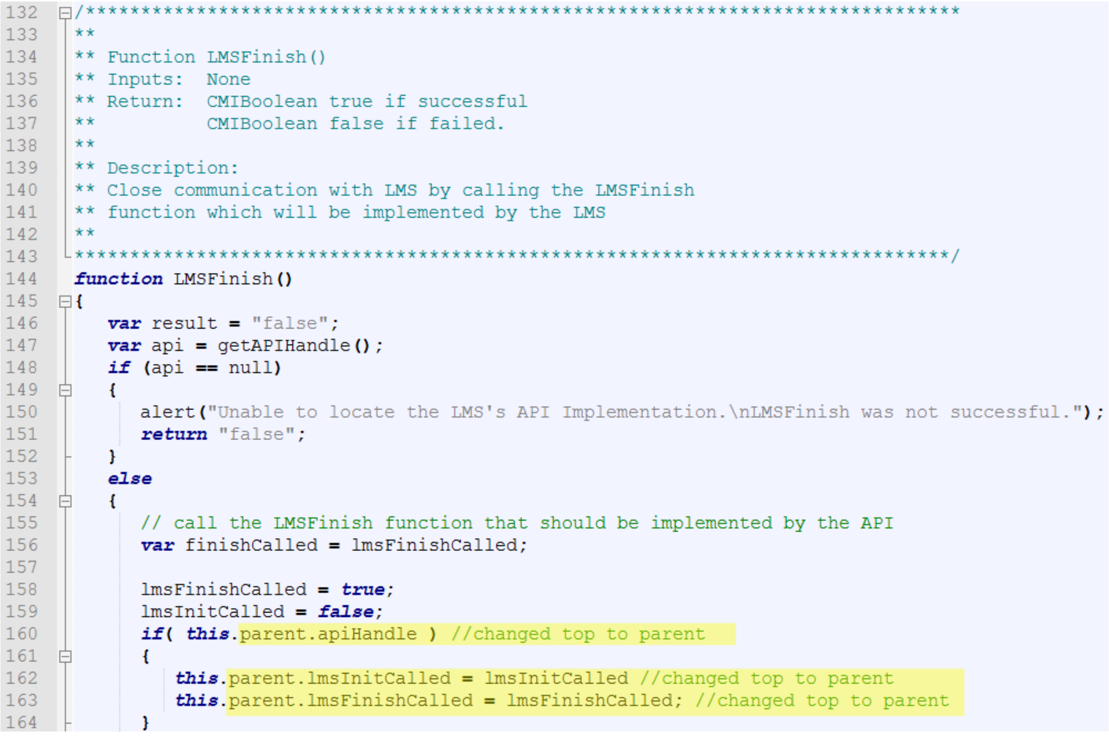
Navigate to the LMSFinish() section (Starts at line 132)

On lines 160, 162 and 163 replace “top” with “parent”



```
132 /*******************************************************************************
133 **
134 ** Function LMSFinish()
135 ** Inputs:  None
136 ** Return:  CMIBoolean true if successful
137 **          CMIBoolean false if failed.
138 **
139 ** Description:
140 ** Close communication with LMS by calling the LMSFinish
141 ** function which will be implemented by the LMS
142 **
143 *******************************************************************************/
144 function LMSFinish()
145 {
146     var result = "false";
147     var api = getAPIHandle();
148     if (api == null)
149     {
150         alert("Unable to locate the LMS's API Implementation.\nLMSFinish was not successful.");
151         return "false";
152     }
153     else
154     {
155         // call the LMSFinish function that should be implemented by the API
156         var finishCalled = lmsFinishCalled;
157
158         lmsFinishCalled = true;
159         lmsInitCalled = false;
160         if( this.top.apiHandle )
161         {
162             this.top.lmsInitCalled = lmsInitCalled;
163             this.top.lmsFinishCalled = lmsFinishCalled;
164         }
```

Becomes



```
/*****************************************************************************
**
** Function LMSFinish()
** Inputs:  None
** Return:  CMIBoolean true if successful
**          CMIBoolean false if failed.
**
** Description:
** Close communication with LMS by calling the LMSFinish
** function which will be implemented by the LMS
**
*****************************************************************************/
function LMSFinish()
{
   var result = "false";
   var api = getAPIHandle();
   if (api == null)
   {
      alert("Unable to locate the LMS's API Implementation.\nLMSFinish was not successful.");
      return "false";
   }
   else
   {
      // call the LMSFinish function that should be implemented by the API
      var finishCalled = lmsFinishCalled;

      lmsFinishCalled = true;
      lmsInitCalled = false;
      if( this.parent.apiHandle ) //changed top to parent
      {
         this.parent.lmsInitCalled = lmsInitCalled //changed top to parent
         this.parent.lmsFinishCalled = lmsFinishCalled; //changed top to parent
      }
```

Save changes.
Any new e-Learning packages you now publish will use the updated file.

13.7.1 Existing e-Learning Packages

It is not necessary to republish existing e-Learning packages.
All that is required is to replace the existing apiwrapper11.js file with the updated file in each e-Learning package folder location on the content server.