ELECTRONIC STAFF RECORD

ESR-NHS0167 – ESR E-LEARNING CONTENT GUIDANCE AND STANDARDS

Information Classification: Public

Owner: ESR Development and Operations Team
Author: Matthew Lye
Creation Date: 10 March 2008
Last Updated: 24 December 2025
Next Update: 24 December 2027
Version: 13

Approvals:

Name	Chris Moorley
Title	Head of Operations and Development

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 content in 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
04/06/2024	Matt Lye	11.1	General review
24/06/2024	Chris Moorley	12	Approved version
23/09/2025	Matt Lye	12.1	General review
24/12/2025	Chris Moorley	13	Approved version

1.2. REVIEWERS

Name	Position
Pushpa Mistry	NHS ESR Development Team
Sam Wright	NHS ESR Development Team
Anna Phillips	NHS ESR Development Team

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 England 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

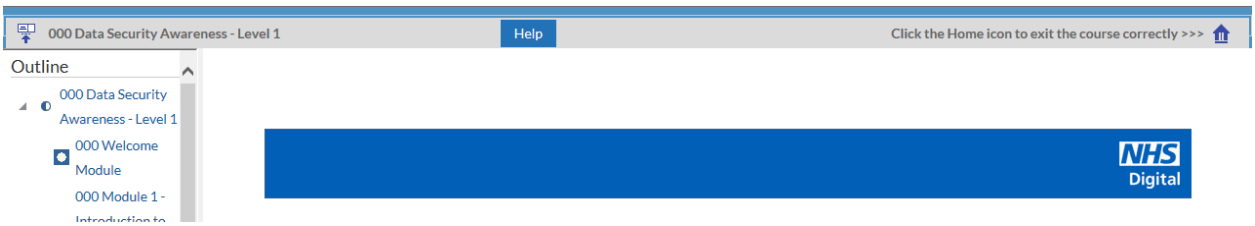
3.3. FURTHER INFORMATION

Setting up e-Learning content in ESR.

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.



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, the 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>
```

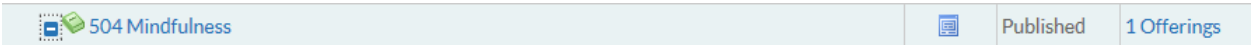
If the content has an exit button, it should not attempt to close the OLM Player window. Instead the exit button should make the relevant calls, and then instruct the user to close the course manually

5. CONTENT DEPLOYMENT GUIDELINES

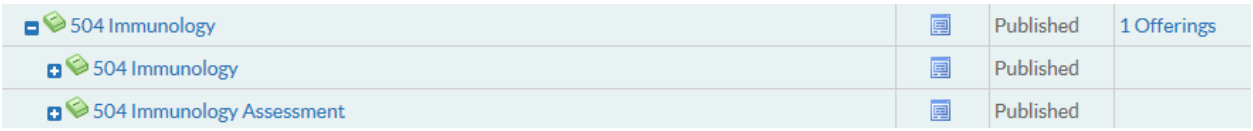
5.1. ENSURING ACCURATE TRACKING

Content in ESR can be set up in two ways.

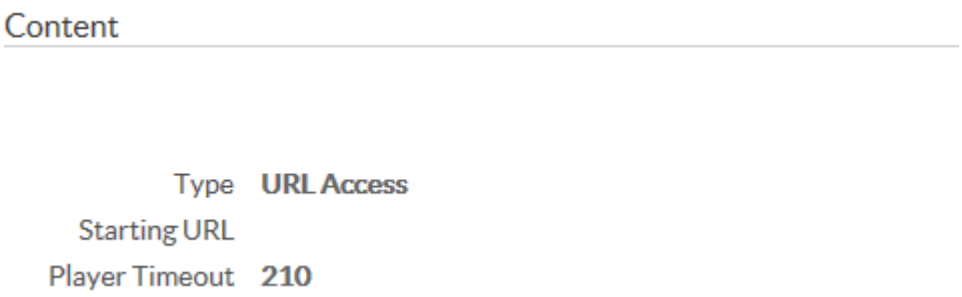
1. Course linked to a single learning object



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



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

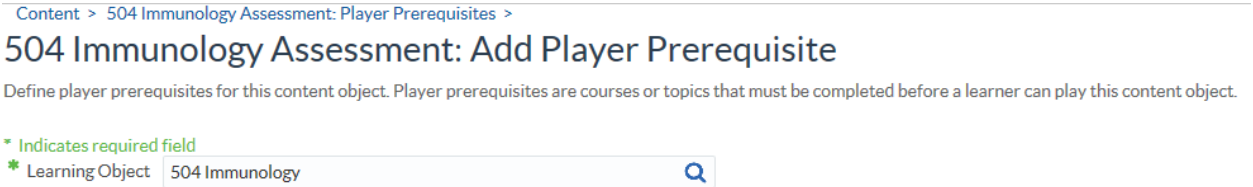


The completion status and time played for the parent learning object is calculated by aggregating the completion status and time spent from the child learning objects.

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.



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 Microsoft Edge, the Development Console 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

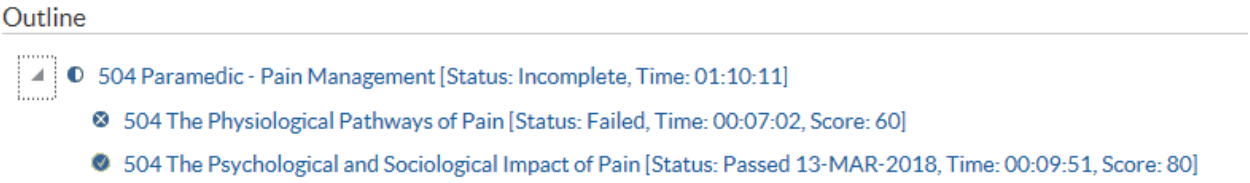
Tests enable you to create and manage tests either as learning objects within a course, or as courses in their own right. Tests can be scored, or set up as surveys for which there is no score. Tests are made up of questions stored in a question bank created in ESR. ESR handles presentation of the test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

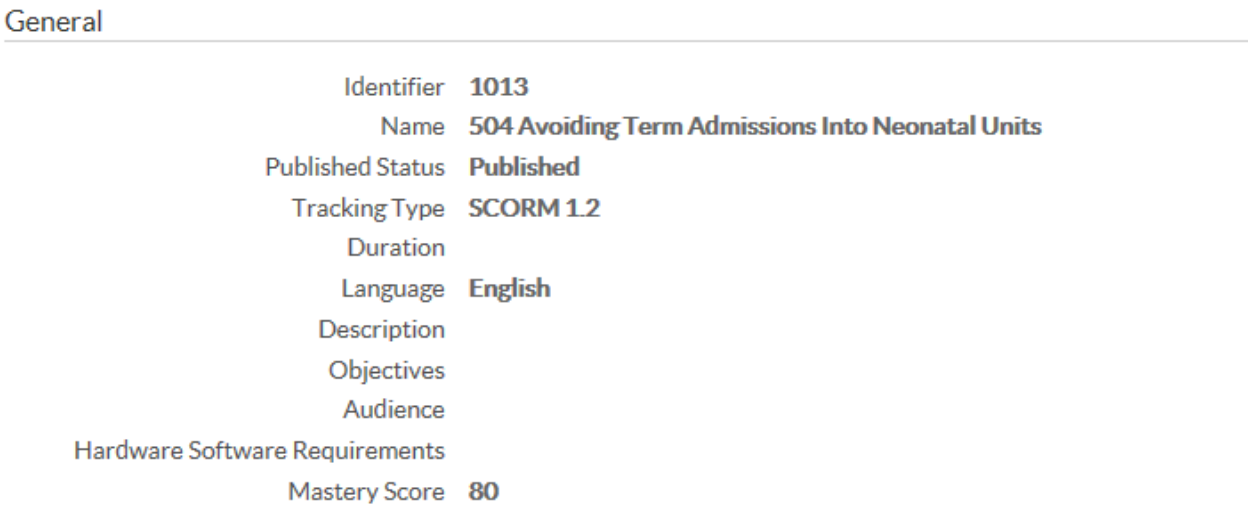
The tracking of a learner's progress is directly tied to the learning objects within a course. When a learner plays a course for the first time, an attempt record for the learner is created in ESR. This stores information such as the time spent, completion status, and score for that learning object. 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.



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.



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.

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:

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).

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.

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 EPRO, the familiarisation environment available to organisations.

8. COMMON ISSUES FOR E-LEARNING DEVELOPERS

8.1. CONTENT DOES NOT DISPLAY CORRECTLY IN A WEB BROWSER

There may be certain content that does not display correctly or certain functionality does not work as expected. 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 some courses, 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 1 - 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.
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