imc
part of Scheer

SCORM Creator's Guide

imc Learning Suite

Creation of Web Based Trainings

im-c.com

SCORM Creator's Guide

imc Learning Suite

Creation of Web Based Trainings

Author(s): Andreas Pohl
Date: 2020-11-23

Document	Description
Version	ILS 14.8
Status (Draft / Review / Finalisation)	Finalisation
Contact Person(s)	Andreas Pohl

History	Status	Who
2018-02-08	Draft	Andreas Pohl
2020-11-23	Finalisation	Dr. Peter Zönnchen

imc
information multimedia communication AG
Headquarter Saarbrücken
Scheer Tower, Uni-Campus Nord
D-66123 Saarbrücken
T. +49 681 9476-0 | Fax -530
info@im-c.com
im-c.com

Content

1 Introduction

This documentation aims at providing basic information, guidelines and recommendations for creating SCORM content to be used in imc Learning Suite (ILS). This should help third-party content creators as well as our customers to prepare their SCORM content for optimal usage in the learning management system (LMS).

The following sections clarify on the supported SCORM versions, provide recommendations for content creation and guidelines for integration into ILS.

2 SCORM Versions

imc Learning Suite (ILS) supports the following versions of the SCORM standard: 1.2 and 2004:

- SCORM 1.2 is fully supported in ILS
- SCORM 2004, also referred to as version 1.3, is supported by ILS up to the 3rd Edition excluding sequencing & navigation, multiple organizations.

Version 1.2 already support all the functions that most of our customers are using concerning web based trainings and is at the same time less complex. Therefore, there is still a larger part of SCORM Web Based Trainings (WBTs) based on version 1.2.

3 Recommendations

This section provides recommendations on how to create SCORM packages in general and for mobile usage.

3.1 General

3.1.1 Structure

Concerning the structure, we recommend using only SCOs and no assets. The majority of SCORM WBTs used in ILS are single SCO WBTs.

3.1.2 File names

File names should not include umlauts or special characters. This is running into error situations. Additionally, path names including file name should not be longer than 255 characters as this might lead to issues on certain system combinations.

3.1.3 Encoding

All HTML pages and attribute sets inside the SCORM package need to be encoded with UTF-8.

3.1.4 HTML

We strongly recommend ensuring HTML 5 compatibility for all web based training content.

3.1.5 Browser Compatibility

We recommend going without special handling for Internet Explorer (IE). The default settings should be "standard mode" and "IE = Edge".

3.1.6 Status transfer frequency

Status transfer, i.e. sending of the user's status on the WBT (or in fact the respective SCO) to the LMS, should not happen too often and not only once. From our experiences with many customers and successful SCORM usages, we recommend transferring the status every 5 minutes and on change. I.e. the status should be sent regularly, but not too often to prevent unnecessary high loads.
Additionally, the status should be transferred whenever the completion status changes, e.g. from "incomplete" to "completed".
You should also send the status more often than only once at the end, to prevent that users close the WBT before that point and then do not have a status at all.

3.1.7 Final status transfer

The final status transfer, i.e. the status change from "incomplete" to "completed" (or also "passed" or "failed") always should be in line with the display to the user. This means that when the user is shown a page saying, "you successfully completed the object", then the status should already be sent. We have seen a couple of acceptance issues where there was such a page, but the status was only sent on the next page change. This resulted in users leaving the WBT, not getting a final status, and later complaining that the LMS is not recording their status. This situation is solely an issue with the WBT setup.

3.1.8 Session time tracking

Your web based training should always record and transfer the session time, i.e. the time the user spent on the learning content. This is not technically needed, but our experience shows that users will complain. So, we recommend to always enable the tracking and transfer for the session time.

3.1.9 Language handling

We recommend to only use single language web based trainings. The imc Learning Suite provides a multi-language concept that is based on a preferred content language which can also handle SCORM WBTs that are equal in content, but different in language and is available the same for nearly all other content types in the LMS.

3.1.10 Score

Whenever a score is used, we recommend setting the maximum score to 100 (score_max).

3.1.11 Finishing the SCORM WBT for a user

The SCORM content itself should always call the "finish" method to properly close the WBT session and transfer the data to the LMS. The "finish" should also be triggered on unloading the WBT, to prevent that users close the window and will lose their status.

3.1.12 Exit and Close Buttons in the SCORM content

We do not recommend using individual exit or close buttons inside the SCORM package. Especially, they should never trigger a "top.close()" function.

3.1.13 Tracking of Answers to Quiz Questions

When tracking answers to quiz questions (which is not standard SCORM and is up to the individual implementation or authoring tool), we strongly recommend to only keep the latest answers.
Otherwise, tracking each and every execution of the quiz, can easily lead to unnecessary high database storage volumes in the LMS database – on high number of users paired with a high rate of multiple quiz execution.

3.2 Mobile

3.2.1 Usage of Media Types and Sizes / Resolutions

When creating your SCORM content to be also used for mobile devices, please especially check for used media types (only those that are supported on all target devices and platforms, e.g. no Flash) and media sizes and resolutions to be optimized for use on tablets or smartphones.

3.2.2 Memory handling

We recommend testing memory usage of SCORM content especially on mobile devices as they might have memory restrictions. Be aware that in the scenario of using the native app (in contrast to accessing ILS on the browser on a mobile device), the app itself is already using a part of the available memory.

3.2.3 Content in an iFrame jumping back to Top (iOS)

There is a bug in several iOS versions where changing some CSS properties (like box-shadow, font-family, opacity, probably much more) or changing the text content of an element can bring Safari to jump back to the top of the iFrame document. This can happen too when selecting / setting the focus on a form element or modifying the height of the content. The workaround is to not use iframes, instead load all external HTML content into scrollable DIV elements using Ajax. Recommendation: JQuery's .load() method does a perfect job on this, because any inline JavaScript code that is part of the external document will successfully execute.

3.2.4 CSS and HTML issues

Some CSS properties still need the webkit vendor prefix (e.g. transform), i.e. a check is needed for when using CSS3 tags (see iOS Developer Library).
Further preloading for HTML5 videos needs to be set explicit (see iOS Developer Library).

3.2.5 Non-supported JavaScript events

Safari on iPad does not support the (window.)onbeforeunload and (window.)onunload events (see W3Schools documentation). Apple is marking this as deprecated. This function is used to check, if the user will leave the page and to display an warning message that all changes will be discarded. This is used for e.g. for tests in the Learning Portal, to avoid, that a user loses all input by leaving the page. This results that if the user leaves the screen at iOS devices he will not get a warning message, Instead the event "pagehide" can be used.
Further user interactions should listen for the "touchend" event instead of "touchstart" or 'click, as "a sequence of touch events is delivered to the element that received the original "touchstart" event regardless of the current location of the touches" (see iOS Developer Library).

3.2.6 View port Scaling issue

On old versions of iOS (not reproducible any more on iOS 8.x), on orientation change, the page content is zoomed. For SCORM player, iFrame size gets a wrong value, because of this zooming. A solution for this case will be to set this tag:

```
<meta name="viewport" content="width=device-width; initial-scale=1;
maximum-scale=1">
```

Be aware that this completely disables zooming on all mobile devices which can be bad for the user experience!

3.2.7 Frequent page reloads triggered by Safari

Since iOS7 and more noticeable on iOS8 there seem to be frequent page reloads when a page freezes or crashes. There's no error in the browser console, only the message "A problem occurred with this web page so it was reloaded" appears when the page was reloaded. The reloading behavior is enabled by default and can't be deactivated. This might be RAM-related since most iOS devices have a maximum of 1 GB RAM.
The reasons for the behavior can be:

- one pager with too much content
- content which needs to initialize (e.g. carousels / image sliders) will amplify the problem
- too many JavaScript libraries / frameworks / scripts

When this issue is experienced the following action can be possible solutions:

- always minify and combine CSS + JavaScript
- reduce HTML mark-up and JavaScript to a bare minimum
- dynamically load and unload content (or portions of content)

3.2.8 Different jQuery Behavior

The scrollTop() function of the jQuery library needs the context parameter "parent.document", which then can't be used in any other browser. Therefore, you need to check the user agent and implement the jQuery selector accordingly. Same goes for the .animate() function with the scrollTop property:

- iOS: $('body, html', parent.document).animate({ scrollTop: 0 }, 1500);
- every other browser: $('body, html').animate({ scrollTop: 0 }, 1500);

4 Creation / Usage in ILS

The upload process and the available properties and functions, when importing a SCORM learning content in the imc Learning Suite are comprehensively explained in our standard documentation << SCORM WBT creation guidelines >> ("ILS_Documentation_SCORM_WBT_creation_guidelines_EN.pdf"). Here, we only want to highlight two options that should be considered in combination with the SCORM package setup.

4.1 Enabling for Learner Frontend and Mobile Apps

For the SCORM web based training to be used in the learner's frontend (ILP) and the mobile apps, the content object needs to have the setting "Can be used in external applications" enabled.

4.2 Opening Modes

The imc Learning Suite allows for different opening modes in the learner's frontend: same window, new tab, and pop-up.
We recommend the "same window" setting from a user experience point of view. But based on the setup of the SCORM package, one of the other might be needed. For example, there are "older" WBTs that close the surrounding window when the users close them. They can still be used in the LMS with "new tab" or "pop-up" settings.

4.3 Status Calculation

The LMS provides an additional function for web based trainings that works on top of the SCORM standard. Normally, only a SCO, i.e. a part of the WBT, returns a status to the LMS. Thus, if a complete WBT is included into a syllabus, this object can return a status based on the SCORM standard.
Therefore, imc extended the WBT handling to allow a status calculation for complete WBTs – setting their status to completed or passed if all included SCOs are completed or passed.
In summary, ILS provides two possibilities to integrate SCORM content into your syllabus, the complete WBT or only individual SCOs - and you can already consider these possibilities when creating your content.