

SCORM WBT Creation Guidelines

imc Learning Suite

SCORM WBT Creation Guidelines

imc Learning Suite

Author(s): Daniel Dahrendorf

Date: 2021-09-27

Document	Description
Version	ILS 14.10
Status (Draft / Review / Finalisation)	Review
Contact Person(s)	Daniel Dahrendorf

History	Status	Who
2016-11-16	Draft	Daniel Dahrendorf
2020-11-23	Review	Daniel Dahrendorf
2021-09-27	Review	Daniel Dahrendorf
2021-09-27	Finalisation	Dr. Peter Zönnchen

Content

1	Certification	4
2	Adding SCORM content packages in the Learning Suite	5
2.1	Importing Files	5
2.2	PENS Standard	6
3	Configuration of SCORM content packages	8
3.1	Using meta-data	8
3.2	Using Single SCO's and complete SCORM packages in the learning logic of the imc Learning Suite	9
4	The run-time environment	10
4.1	Data model	10
4.2	Finishing a SCO	11
5	Enabling SCORM content for Learning Portal and mobile Apps	12
6	Display of SCORM content packages	13
6.1	Displaying SCORM content in the Learning Suite	13
6.2	Displaying SCORM content in the imc Learning Portal	15

1 Certification



The imc Learning Suite is officially certified by ADL on the level LMS-RTE3. Thus, the technical aspects of the SCORM interface are sufficiently described, and content producers are referred to the respective ADL documents (<http://www.adlnet.org>) that accurately define the standard. Therefore, the specification itself won't be repeated in the following chapters and a thorough knowledge of the standard is taken for granted.

2 Adding SCORM content packages in the Learning Suite

2.1 Importing Files

The import of SCORM content into the imc Learning Suite needs to be done via uploading a ZIP file, containing the content package, as it is specified in the standard. Please do note that there must be a manifest file named `imsmanifest.xml` on top-level of the ZIP file. This file contains the relative references to included files or content pages to the package root. When uploading SCORM content, the administrator executing the upload needs to decide which version of SCORM will be imported (1.2 or 2004). The version is not detected automatically.

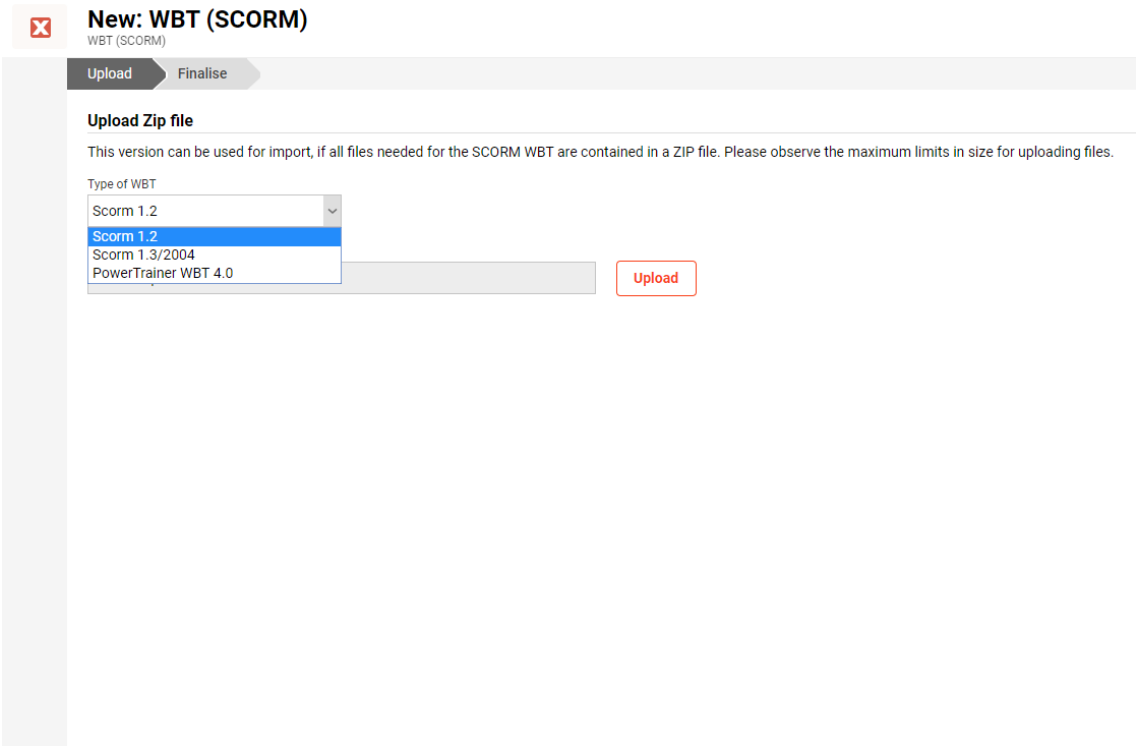


Fig. 2.1: Uploading a new SCORM media into the Learning Suite

As the imc Learning Suite operates with various languages, the platform must be able to work with different character sets. Therefore the manifest files should be saved in UTF-8. Otherwise, umlauts and other special characters of descriptive data in the imc Learning Suite may be displayed incorrectly. The names of all files within the ZIP file must not contain any umlauts or special characters.

A SCORM WBT can be updated by reloading single parts of it. This should only serve to fix bugs, since no structural information is checked. Only SCORM learning objects which are already

created in Learning Suite can be replaced this way. Please do note that the ZIP file needs to contain the respective content files in the same directory structure as the one which was initially created. Please make sure, that the SCORM communication is not changing when updating a SCORM object, specially that existing SCORM runtime data - created from the old version of the files - are compatible with the new version of the files.

imc recommends the usage of SCORM 1.2.

2.2 PENS Standard

The imc Learning Suite also supports the PENS standard. By using a PENS compatible authoring tool (e.g. the imc Content Studio) content can be directly uploaded into the imc Learning Suite. HTTP(S) and (S)FTP are supported. The configuration of the imc Content Studio is quite easy and intuitive:

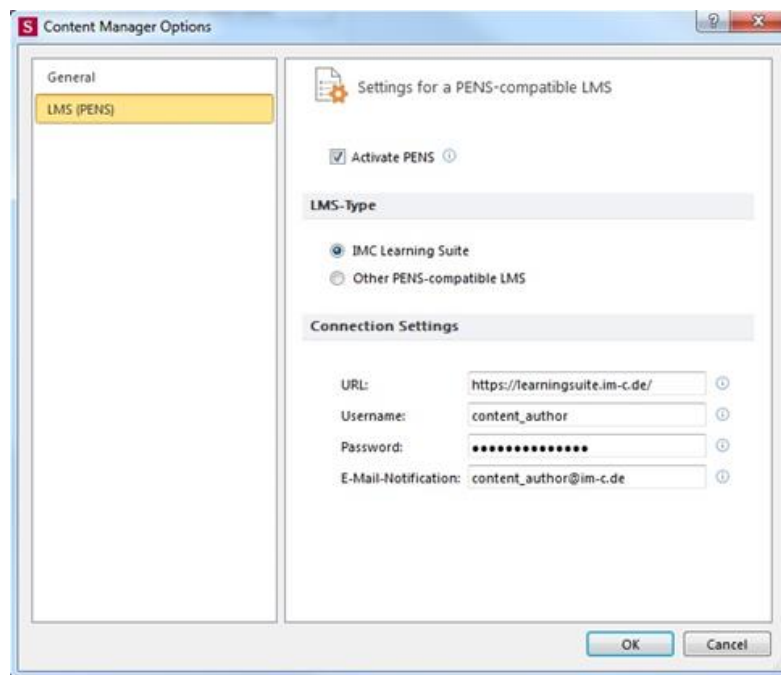


Fig. 2.2: PENS configuration within the imc Content Studio

The user also gets a notification whether the upload was successful:

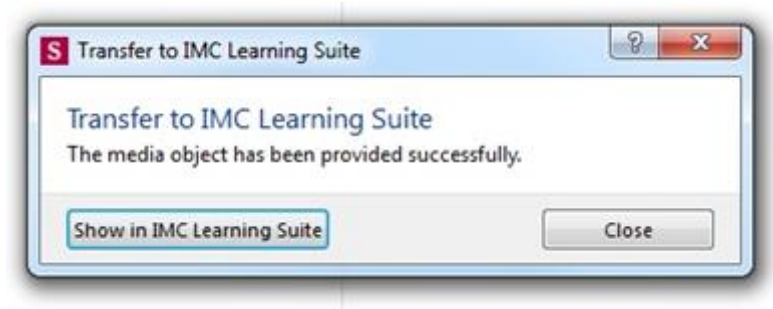


Fig. 2.3: Upload feedback of PENS within the imc Content Studio

By using the PENS option with imc Content Studio an auto detection of the content is done concerning the following scenarios:

Has content been published to the imc Learning Suite before?

Has the SCORM WBT structure changed?

The user can also decide whether the content should be added as a new media object or whether an existing content should be updated (only available if the structure is unchanged). This prevents content authors breaking running course scenarios in the imc Learning Suite by using the content update function.

3 Configuration of SCORM content packages

3.1 Using meta-data

The imc Learning Suite basically accepts every valid form of LOM meta-data. But imported meta-data only result in an increase in value, if they also are displayed on screen or available for search. In standard installations of the imc Learning Suite only those LOM parameters listed in the table below are carried over for each learning object and displayed on the respective info pages. Since it's possible to specify LOM meta-data in a SCORM manifest file for any element on any hierarchy level, it's crucial that the LOM annotation takes place on the right level. Meta-data on top-level of the manifest are especially important as they describe the imported SCORM WBT in Learning Suite as a whole. At that location, especially the parameters title and description, as well as keywords, ought to be set, as those can be searched for in the imc Learning Suite. On all other levels, the meta-data listed in the table below can at least be displayed. It is possible to embed separate SCO's as objects into courses of the the imc Learning Suite (described in more detail below). Particularly in case this is desired by the platform operator, the appropriate item has to be filled with respective meta-data. Regarding meta-data it's recommended to swap those into separate files referenced in the manifest file in order to avoid possible name space problems.

Table 1: LOM parameters imported in the Learning Suite

Meta data element	LOM parameter 1.2	LOM parameter 2004
Title	general.title	general.title
Description	general.description	general.description
Keywords	general.keyword	general.keyword
Structure	general.structure	general.structure
Language	general.language	general.language
Granularity Level	general.aggregationlevel	general.aggregationLevel
Version	lifecycle.version	lifeCycle.version
Status	lifecycle.status	lifeCycle.status
Interactivity Type	educational.interactivitytype	educational.interactivityType
Learning Resource Type	educational.learningresourcetype	educational.learningResourceType
Difficulty	educational.difficulty	educational.difficulty
Learning Time	educational.typicallearningtime	educational.typicalLearningTime

3.2 Using Single SCO's and complete SCORM packages in the learning logic of the imc Learning Suite

The imc Learning Suite supports a learning logic within the syllabus of a course. A learning logic describes a set of rules which individually determine the learning path within a course. Apart from complete SCORM WBTs the imc Learning Suite can also embed separate SCO's into the syllabus. Thus SCOs from different WBTs can be put together within the scope of one course. Furthermore, this offers the possibility to define learning logic rules on SCOs since single SCOs have a defined status as opposed to WBTs. For using this functionality it is mandatory that the SCO behaves in a way that rules can be successfully carried out. The correct handling of the status a learner holds for a SCO is particularly critical. As long as a SCO is running and isn't considered to be completed by the learner, the status should obtain the value incomplete. As soon as the SCO is considered as completed an end status should be set either completed, passed, or failed. Furthermore, the score always must be declared in percent since the learning logic rules are defined in percent too.

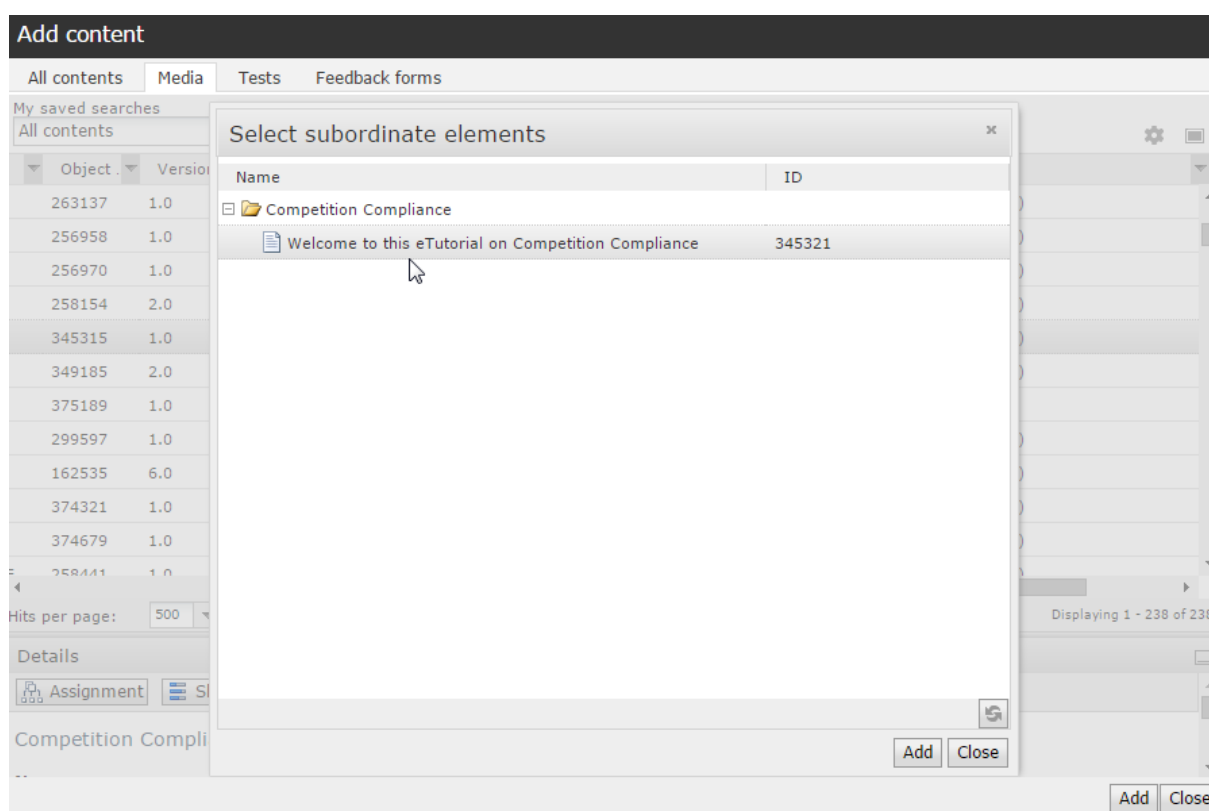


Fig. 3.1: Adding a single SCO to a course

A SCORM WBT is basically a collection of various SCOs. While each SCO has its own status, it makes sense to display an aggregated status for the WBT to learners, if the complete SCORM WBT is added to the course syllabus. This also allows using learning logic depending on the status of a SCORM WBT. For this purpose the Learning Suite sets the status of the WBT to passed, if all SCOs are completed or passed. If all SCOs are completed, passed and at least one of them failed, the status of the WBT is set to failed.

4 The run-time environment

4.1 Data model

The imc Learning Suite completely supports the run-time environment data model, but the following data model elements, which are fundamental in the imc Learning Suite and therefore should be communicated by SCO's at any rate, play an especially important role:

Status (`cmi.core.lesson_status` or `cmi.success_status` + `cmi.completion_status`): The current status is shown to the learner for each SCO. Furthermore, the status plays a very important role in case the SCO is embedded in a Learning Suite learning logic. If the learner has completed an SCO, this should also be represented by the status (completed, passed or failed) right then.

Learning time of the current session (`cmi.core.session_time` or `cmi.session_time`): As the aggregated learning time (`cmi.core.total_time` or `cmi.total_time`) is available to the learner in the overview at all times, the session-related learning time should be set by the SCO. This often counts as the simplest check whether the communication between SCO and Learning Suite was successful.

Score (`cmi.core.score.raw` or `cmi.score.raw`): The accomplished score is displayed in the structure overview page as well. A standardized input in percent is required for the Learning Suite. Besides, the score plays an important role, if the SCO is embedded in a Learning Suite learning logic.

If a SCO is called up from within the media management or the tutorial area, the attribute `cmi.core.lesson_mode`, or `cmi.mode`, is given the value `browse` (default `normal`), the attribute `cmi.core.credit`, or `cmi.credit`, is given the value `no-credit` (default `credit`) in the imc Learning Suite. Reading these values out is an offer by the imc Learning Suite to the learning content, which is thus given the possibility to make its behaviour dependent on the calling context. As administrators and tutors are not learners themselves, it should be actually abstained from reporting "learning run-time data" to the platform and thus having them saved. If data are sent to the imc Learning Suite, they are handled to the full extent irrespectively of the run-time mode.

4.2 Finishing a SCO

Section 2.1.3 of the “SCORM 2004 3rd & 4th Edition Specs” (see <https://www.adlnet.gov/adl-research/scorm/scorm-2004-4th-edition/>) points out:

In some cases, the content author does not want to allow the user to interact with the SCO after it has finished (whatever “finish” means in the context of the SCO). In such a case, the following behaviors are allowed, depending on the type of window in which the SCO was launched (refer to Section 3.2: *LMS Responsibilities*):

1. If the window in which the SCO was launched is a top-level window (i.e., the window has no parent window, but it has an opener) then the SCO may attempt to close the window after calling `Terminate(“”)`. There is no requirement that the SCO behave this way. It is recommended that an LMS monitor the status of the dependent pop-up window in which it launched the SCO to detect when such an event happens. This will allow an LMS to present the appropriate implementation-defined user interface to the learner.
2. If the window is not a top-level window (i.e., the window has a parent window), the SCO may not act on the parent window or any window in the chain of parents. For example, a SCO is not allowed to attempt to close the top window, unless it is its own window.

In such a case, the recommended behavior is for the SCO to display neutral, passive content while waiting to be taken away by the LMS.

As consequence is

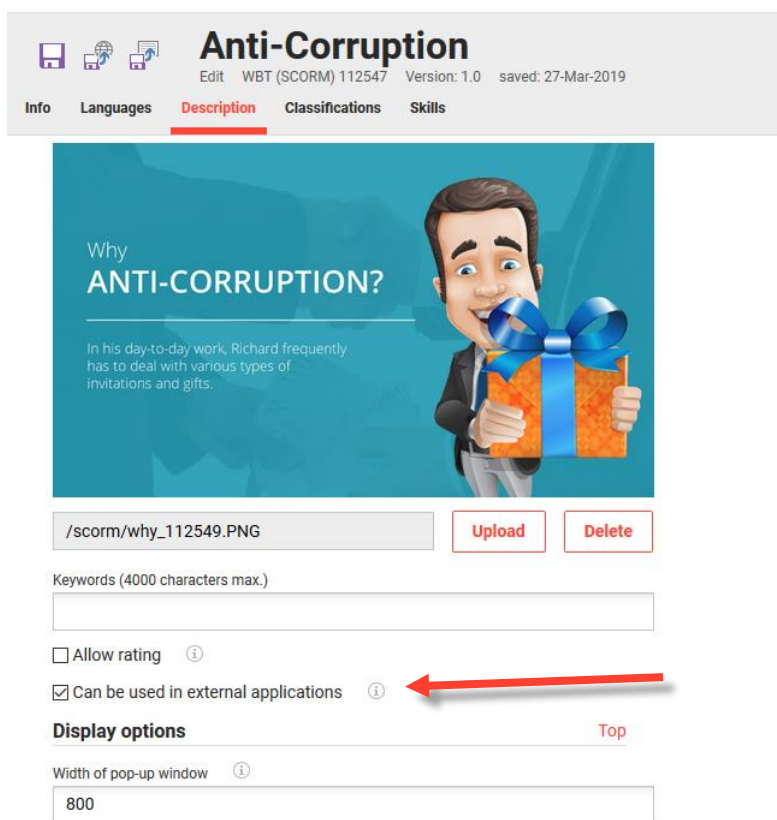
```
top.window.close()
```

not allowed to be executed by the SCO without checking, if it is on top level or not. The recommendation is to not use this command at all.

5 Enabling SCORM content for Learning Portal and mobile Apps

To display SCORM content in the imc Learning Portal and mobile Apps the meta tag "Can be used in external application" must be configured for the media and activated. This feature was introduced to hide non mobile-ready content from mobile apps. Content which does not have this meta tag or the meta tag value is false will not be displayed in the Learning Portal and mobile Apps.

Please note that sequencing and navigation is not enabled in the imc Learning Portal and mobile Apps. For this reason the meta tag "Can be used in external application" is disabled and cannot be activated for SCORM WBTs which have sequencing and navigation enabled. Some WBTs have the feature sequencing and navigation activated in imsmanifest.xml, but do not use it at all. Please clarify that WBT's program code is really not using sequencing and navigation features, then remove the corresponding entries in imsmanifest.xml. Alternatively: If you have the possibility to publish the WBT in SCORM 1.2 instead of 2004, simply do so, because SCORM 1.2 does not have sequencing and navigation at all per definition.



The screenshot shows the configuration interface for a WBT titled "Anti-Corruption". The interface includes tabs for "Info", "Languages", "Description" (selected), "Classifications", and "Skills". The "Description" tab displays a preview of the content, which includes a cartoon character holding a gift box and the text "Why ANTI-CORRUPTION?". Below the preview, there is a file upload section with a text input field containing "/scorm/why_112549.PNG" and buttons for "Upload" and "Delete". A section for "Keywords (4000 characters max.)" has an empty text input field. Below this, there are two checkboxes: "Allow rating" (unchecked) and "Can be used in external applications" (checked). A red arrow points to the "Can be used in external applications" checkbox. At the bottom, there is a "Display options" section with a "Width of pop-up window" input field set to "800".

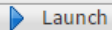
Fig. 5.1: Enabling SCORM content for Learning Portal and mobile Apps with enabled meta tag "Can be used in external applications"..

6 Display of SCORM content packages

6.1 Displaying SCORM content in the Learning Suite

A hierarchical arrangement of the structure is used for SCORM WBTs within the imc Learning Suite. Depending on LMS settings the learner is able to open and close the folder structure and already obtains certain status information about each SCO in this overview. It's recommended to choose an adequate granularity for SCORM WBT's. Extremely fine-grained content - with many SCOs - can lead to the interaction between the learner and the system becoming very intricate and the learner having difficulties regarding the navigation within the large structure. So a structure of eight SCOs is recommended by imc. Extremely fine grained content can also cause bad performance. Often you have WBTs that are monolithic, consisting only of one SCO.

Excel 2013 for Beginners

 Launch			
Contents	Score (%)	Processing sta...	Processing time
 Using the menu		-/-	
 Inserting columns and rows		-/-	
 Using of filters		-/-	
 Configuration of the page layout		-/-	

Details	
	
Using the menu (SCO 552344) (Object ID: 552345)	
Status	Component was not yet processed
Learning time (total)	00:00:00
Name	Using the menu

Fig. 6.1: Display of the structure of SCORM WBT's

Various learning content have problems with the localization of the API adapter, which unfortunately can't be verified through the ADL test suite, even though the implementation in the imc Learning Suite is correct and standard-conform. The API adapter can be found with the standard method for localization (getAPI()). The respective source code is listed in the SCORM

specification. It has to be considered however, that this method first searches for the API adapter towards the root in the frame set hierarchy of the current window. If the API adapter isn't found that way, the search is continued in the opener of the location where the search was started.

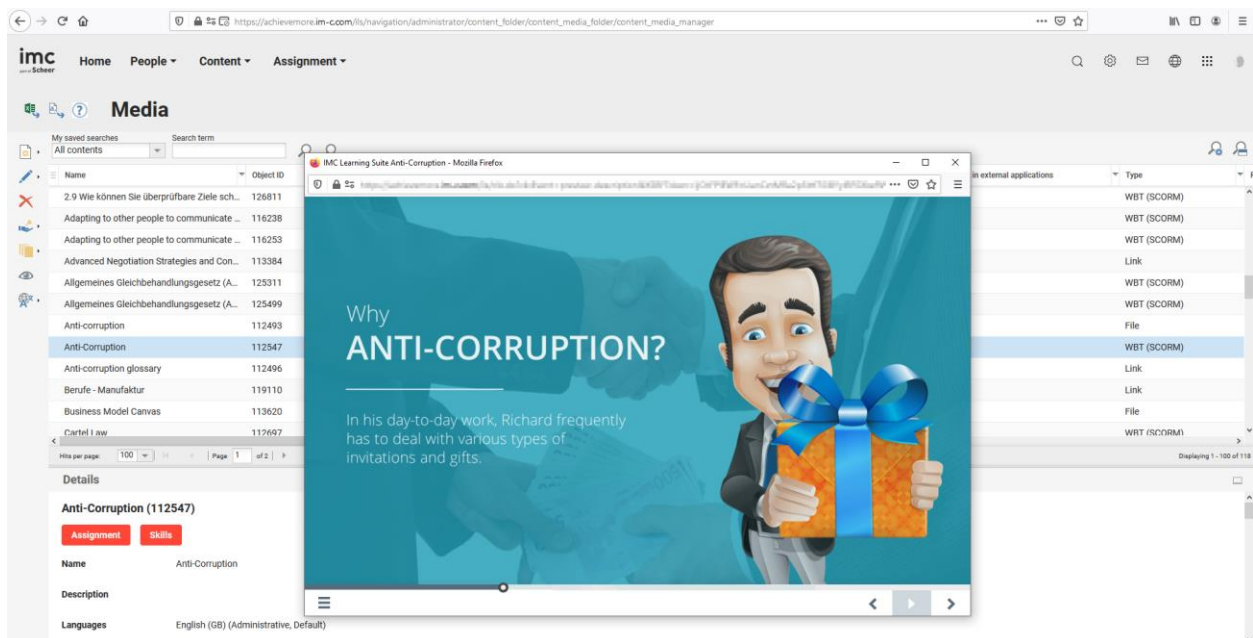


Fig. 6.2: Display of a SCORM in a separate window

It's important to make sure that when the content window is manually closed by the learner, a transfer of the learning run-time data is initiated by the SCO and that the communication was finished properly by the finish method (`LMSFinish("")` or `Terminate("")`). As far as the subject of cross domain scripting is concerned, note that for safety reasons JavaScript doesn't allow communication across domain boundaries. This isn't evaded in the imc Learning Suite either. If a SCORM learning content shall be hosted on an external server, at least the communication interface and the start pages of the respective SCOs to be launched have to be created on the server of the learning platform by checking-in in the imc Learning Suite.

6.2 Displaying SCORM content in the imc Learning Portal

The imc Learning Portal uses iFrames to provide the SCORM API and allows the user to navigate back to the syllabus. This was done in favour for avoiding issues with pop-ups and proving the context and navigation to the user. On Learning Portal there are JavaScript listeners for load / resize, orientation change events that computes new iFrame height and width.



Fig. 6.3: SCORM WBT displayed in the SCORM player of the Learning Portal