# Courses - Feature #9

## Fully integrate SCORM 1.2

16/10/2009 14:32 - Sven Vanpoucke

| | | | | |
|---|---|---|---|---|
| **Status:** | Feature implemented | | **Start date:** | 16/10/2009 |
| **Priority:** | Normal | | **Due date:** | |
| **Assignee:** | | | **% Done:** | 0% |
| **Category:** | | | **Estimated time:** | 0.00 hour |
| **Target version:** | Backlog (default) | | **Spent time:** | 0.00 hour |
| **Complexity:** | Normal | | | |

| Description |
|---|
| SCORM 1.2 is integrated as far as the obligated functionality goes. Most of 1.8.6 database fields are created so this shouldn't be an issue. The import, without metadata is finished and the obligated functions of the datamodel are implemented. The things that are still open for now are things like objectives, interactions, ... |

## History

#### #1 - 18/10/2009 22:45 - Yannick Warnier

Are interactions and objectives tables defined? An important link (even if not completely up to date) is
http://www.dokeos.com/wiki/index.php/SCORM_tool_redesign which describes the implementation of SCORM in 1.8 (although it doesn't describe the objectives implementation). A more recent database schema (1.8.5) is available here:
http://www.dokeos.com/wiki/index.php/Database_schema_1.8.5

The objectives and interactions part is particularly sensitive because it requires a very specific set of JavaScript functions that avoid specific problems ocurring only with a very small subset of SCORM providers.

#### #2 - 04/12/2009 08:56 - Stefaan Vanbillemont

*- Target version changed from 2 to 3*

#### #3 - 26/02/2010 15:46 - Stefaan Vanbillemont

Postponed to 2.1

#### #4 - 26/02/2010 15:49 - Stefaan Vanbillemont

*- Target version changed from 3 to 7*

Basic 2.1 SCORM packages work. More support is postponed to 2.1 release

#### #5 - 14/03/2011 17:26 - Goulwen Reboux

I've found one of the problem which was simply that the package is in SCORM1.1 while Chamilo expect SCORM 1.2

This allows to open the 4 first SCOs of the package (until "Conduct of Vessels in Restricted Visibility").

To be able to view "Lights & Shapes", I need to solve the following problems :

- The table repository_complex_content_object_item stores the prerequisites (including the prerequisite for the folder)
- The table repository_scorm_item stores only leaf nodes (and so, not the folder) As a consequence the array $content_objects in the class PrerequisitesTranslator located in repository/content_object/learning_path/php/display/prerequisites_translator.class.php doesn't contains folders of the package. So the folder "Steering & Sailing Rules" can't be completed.

When digging in this problem, I've found that the following tables are involved in the storage of the SCORM packages :

# Table `repository_content_object`

The base table is `repository_content_object`, that stores obviously all content
objects. For SCORM, we can have 3 types :

- Folders have the `learning_path` type.
- Leafs have each 2 rows, one with the `learning_path_item` type and another
  with the `scorm_item` type. The 2 rows are linked using the
  `repository_learning_path_item`

# Table `repository_learning_path`

This table stores only informations about folders of a package, got from the manifest (path, version and control mode.)

# Table `repository_learning_path_item`

This table stores informations like the max attempts and the mastery score. This table also contains the link between `repository_content_object.type="learning_path_item"` rows and `repository_content_object.type="scorm_item"` rows.

# Table `repository_scorm_item`

This table stores informations about leafs like the location on disk of the content and some parameters got from the manifest (identifier, time limit, objectives, visible or not).

Notably the identifier used by `repository_complex_learning_path_item.prerequisites` is here but, as it stores only leaf, there's no way to store completion of the folders.

Maybe `repository_learning_path_item` and `repository_scorm_item` can be merged?

# Table `repository_complex_content_object_item`

This table store the list of prerequisites, using the identifier of a leaf. This is a problem as folders can't be completed.

### #6 - 14/03/2011 17:58 - Sven Vanpoucke

This was the same issue i had in the system, after that we got a bit stuck because of our architectural choices and we stopped with scorm integration.

The idea behind the scorm implementation was that it was just like another learning path in the system. Since in chamilo 2.0 everything is reusable, this meant that learning paths are built with sublearning paths (so folders are now translated as sub learning paths). For general learning paths this is the best way to go. However when using the scorm package we noticed that there are also some scorm properties on that folders and because we were using learning paths for folders we were not able to store these properties.

The problem is that, if you want to do it the right way, you need to generate the scorm package as some kind of scorm learning path instead of using the regular learning path content objects. Through this way all the scorm items and folders would become a scorm_item in the system.

The reason why we use a learning_path_item is because this is some kind of wrapper for other content objects (like document, link etc). We currently needs these wrappers because depending on the parent object, the content objects can have different information in the relation with the parent. (for example: a document has prerequisites in a learning path, but a document has no additional information in a portfolio), so we need a wrapper item for documents in the learning path so we can add this additional information. Thats why it was also used for scorm items, although they are only used in one learning path, it was easier to just reuse the code and use the same structure for both chamilo and scorm learning paths.

### #7 - 15/03/2011 12:22 - Goulwen Reboux

Even I've made the same assumption about reusability while working on another LMS, I'm finally not sure that it works with SCORM packages, especially with SCORM2004. Because a package includes some logic by itself (read sequencing) it's not easy to take part of a package and use it in a new context. You need to do some work on the content, which will probably done outside of Chamilo.

This is somewhat related to the feature asked on the following [issue 1297](#) and to the discussion in [issue 491](#)

So we need to change the way the SCORM Package is imported. At this step, we need to evaluate what changes have to be made and I've created a new Wiki [Scorm_support](#) page to collect all informations about those changes.

### #8 - 01/04/2011 14:36 - Stefaan Vanbillemont

*- Project changed from Chamilo LCMS Connect to Courses*

*- Category deleted (2)*

### #9 - 14/04/2011 14:26 - Stefaan Vanbillemont

*- Target version changed from 7 to 2.1.0*

### #10 - 16/06/2011 14:32 - Sven Vanpoucke

*- Target version changed from 2.1.0 to 88*

**#11 - 17/06/2011 13:21 - Stefaan Vanbillemont**

*- Target version changed from 88 to Backlog (default)*

**#12 - 22/06/2012 13:48 - Stefaan Vanbillemont**

*- Status changed from New to Feature implemented*