

Moodle Tool Adapter



Implementation Completed!

See the [Animations](#) on this implementation. After seeing the animations, here's a [demo server](#) you can play with.

Intro

The Moodle Tool Adapter is a [tool adapter](#) that permits Moodle tools to be used natively (as [LAMS Tools](#)) within a [LAMS-Moodle integrated environment](#).

This work requires two things:

- A LAMS Tool Adapter that wraps the Moodle tool
- Minimal modifications in the Moodle tool

The LAMS Tool Adapter is basically a LAMS Tool but with almost no "business logic" within it. It provides a set of hooks for the Moodle tools to behave as if it would be a [LAMS Tool](#) following the [LAMS Tool Contract](#).

To talk to this LAMS Tool Adapter, the Moodle tool has to have a few minimal modifications to be incorporated in the context of LAMS.

Animations

As we have now completed this implementation, we have created animations on how this works (we will soon have this available for you to play with):

1. [Authoring and Preview](#)
2. [Enactment/Runtime and Monitoring](#)
3. [Using Moodle Tool outputs for Branching](#)
4. [Achieving interoperability using Export and Import](#)

Demo

You can use the [demo server](#) to try this implementation of the Moodle Forum in LAMS. We are looking forward for comments and suggestions in the LAMS Community [Tech Forums](#). Please let us know what you think!

LAMS Tool Adapter

See the [LAMS Tool Adapter](#) page for further information on its implementation.

Modifications to Moodle and Moodle Tool to work in LAMS

In this page we will focus on the necessary extensions we do in the Moodle Forum Tool to add it as a native LAMS tool. There are [significant benefits](#) on extending a Moodle tool to work on the context of LAMS.

Implementation

The main challenge we faced with Moodle Tools is the same as we face with [other LMSs](#): Moodle tools instances are heavily tied to a course. Therefore you can't quite create Moodle tool instances that are not within a context of a course.


LAMS (and learning design for that matter) has a design stage (authoring), where the teacher sets up the sequence of activities, and a runtime or enactment stage, which is when given the design that the teacher has created, the students go thru this design completing and doing the activities.

Moodle and other LMSs and their tools don't really have this concept. If they would, a teacher should be able to create a Forum outside the course and then instantiate one or more instances of this forum in one (or more) of her courses. Currently, LMS' tools allow teacher to create activities/tools in a course. So teacher goes into the course page, selects a tool to add, completes its content and bang! Now this tool is in the course page.

Trying to do a work around this can be difficult as it requires to "hack" the LMS tool, meaning that not only you need a tool adapter to "bring" this tool into LAMS, but also modify the code (though slightly) in the LMS tool to overcome this limitation.

Here's the list of changes and tasks we have identified that are needed to get a Moodle tool to work in LAMS. We'll go thru each of this individually:

type	key	summary	assignee	reporter	priority	status	resolution	created	updated	due
------	-----	---------	----------	----------	----------	--------	------------	---------	---------	-----

 Can't show details. Ask your admin to whitelist this Jira URL.

[View these issues in Jira](#)

Important Integration Principle

The **main principle** we follow is to **avoid any modification to existing code** whenever possible and **extend the current method or functions instead**.

Modifications to Moodle

- [moodle/lib/db/install.xml](#) (+2 lines)

We added a new column to the `!course_modules! table so it distinguishes between LAMS course module instance (!is_lams!). The !is_lams! is a integer value that can have only a 0 or 1 value. If 0, it means that it isn't a LAMS instance. We use this field to filter instances of course modules in the course page (as you can see below).`

- [moodle/course/modedit-lams.php](#) (+463 lines) new

This file is copy of [modedit.php](#) where we add only a few lines (this is an actual diff):

```
15d14
<     $lamsupdateurl = required_param('lamsUpdateURL', PARAM_TEXT);
```

We add a new parameter called `!lamsupdateurl!` which after saving the content of the mod, this is the URL where we are going to return to in LAMS.

```
53a53,57
>         // Turn off default grouping for modules that don't provide group mode
>         if($add=='resource' || $add=='glossary' || $add=='label') {
>             $form->groupingid=0;
>         }
>
```

We won't need Moodle groups within this tool, therefore we remove it.

```
167c171
<     $modmoodleform = "$CFG->dirroot/mod/$module->name/mod_form_lams.php"; \--\--
>     $modmoodleform = "$CFG->dirroot/mod/$module->name/mod_form.php";
```

As you can see, we also extend the `!mod_forum.php!` file (we'll show this later) by creating a customized copy `!mod_forum_lams.php!`

```
185,187d188
<     $mform->_form->_attributes\['action'\] = 'modedit-lams.php';
<     $mform->_form->addElement('hidden', 'lamsUpdateURL', $lamsupdateurl);
<     $mform->_form->setType('lamsUpdateURL', PARAM_TEXT);
```

Here is where we add the action parameter to the `<form>` as well as add the `!lamsupdateurl!` element to it.

```
215,216d215
<         $fromform->is_lams = 1;
<
```

We have added a column (`!is_lams!`) to the `!mdl_course_modules!` table as well as the `!mdl_forum!` table to identify which instances of forum and/or course modules are to be used for LAMS only.

```

418c417,421
<     redirect("$lamsupdateurl&extToolContentID=$fromform->coursemodule"); \--\--
>     if (isset($fromform->submitbutton)) {
>         redirect("$CFG->wwwroot/mod/$module->name/view.php?id=$fromform->coursemodule");
>     } else {
>         redirect("$CFG->wwwroot/course/view.php?id=$course->id"); >     }

```

After the saving, we return to LAMS using the *lamsupdateurl* appending the *coursemodule id*.

```

441c442
<     $navigation = ''; //build_navigation($navlinks); \--\--
>     $navigation = build_navigation($navlinks); 443c444
<     print_header_simple($streditinga, "LAMS", $navigation, $mform->focus(), "", false); \--\--
>     print_header_simple($streditinga, '', $navigation, $mform->focus(), "", false);

```

As mentioned before, in the authoring stage, the content we create for an activity/tool are independent of the course you create it on. Therefore we get rid of all the navigation breadcrumb.

```

457
<     //print_footer($course); \--\--
>     print_footer($course);

```

No need for course footer, so we remove this too.

That's all!

- [moodle/lib/datalib.php](#) (+1 -1 lines)

As we have added a new column to the *course_modules*, when a Moodle Course page is displayed, we don't want to show instances of Moodle that we have created for LAMS (as these instances might be used at authoring stage). Therefore, we add a new constraint to the *function get_course_mods (\$courseid)* function to show only course modules that aren't used in LAMS.

```

function get_course_mods($courseid) {
    global $CFG;
    if (empty($courseid)) {         return false; // avoid warnings
    }
    return get_records_sql("SELECT cm.*, m.name as modname
        FROM {$CFG->prefix}modules m,
            {$CFG->prefix}course_modules cm
        WHERE cm.course = ".intval($courseid)."
        AND cm.module = m.id AND m.visible = 1 AND cm.is_lams=0"); // no disabled mods }

```

See the *ND cm.is_lams=0* at the end of the query statement.

Modifications to Moodle Forum Tool

- [moodle/mod/forum/db/install.xml](#) (+1 line)

Just as we did with the *course_modules* table, we add an *is_lams* column to the *forum* table. Default value is 0.

- [moodle/mod/forum/index.php](#) (+1 lines)

```

\#$forums = get_records('forum', 'course', $course->id);    $forums = get_records_select('forum',
"course=$course->id and is_lams=0");

```

We replaced the *get_records* function for a *get_records_select* instead and add the *is_lams=0* so it only shows forums that weren't created for LAMS.

- [moodle/mod/forum/mod_form_lams.php](#) (+178 lines) new

This file is also a copy of the original [moodle/forum/mod_form.php](#) which we remove a lot of features that http://code.lamsfoundation.org/fisheye/browse/lams/temp_moodle_dev/moodle/mod/forum/mod_form_lams.php?r=1.1. The only additions/modifications are:

```

<     $this->standard_hidden_coursemodule_elements(); \--\--
>     $this->standard_coursemodule_elements($features);

```

We use *standard_hidden_coursemodule_elements* rather than *standard_coursemodule_elements*

```
138c144
<   $this->add_action_buttons(false, 'Save', false); \--\--
>   $this->add_action_buttons();
```

Instead of showing the standard Save options, we only show one Save button.

- [moodle/mod/forum/view.php](#) (+6 lines)

In order to make the Moodle Forum Tool "**workflow aware**", we need to introduce a "Next Activity" button so after the students have posted their messages they can continue to other activities in the sequence. However, we **only** show this button if this is a Moodle Forum instance used in LAMS (`is_lams = 1`).

- [moodle/mod/forum/lib.php](#) (+134 lines)

As part of adding a new Moodle Tool into LAMS, we need to add a few functions to the Moodle Forum tool so they can be use from LAMS. These functions are based on existing Moodle Forum functions.

moodle/mod/forum/lib.php

```
/**
 * LAMS Function
 * This function clones an existing instance of Moodle forum
 * replacing the course and the userid
 */

function forum_clone_instance($id, $sectionid, $courseid) {

    $cm = get_record('course_modules', 'id', $id);

    if ( ! $cm or ! $existingforum = get_record('forum', 'id', $cm->instance) ) {
        // create a new forum with default content
        $existingforum->course = $courseid;
        $existingforum->name = "Forum";
        $existingforum->intro = "";
        $existingforum->is_lams = 1;
        $existingforum->id = insert_record('forum', $existingforum);
    } else {
        // make a copy of an existing forum
        $existingforum->name = addslashes($existingforum->name);
        $existingforum->intro = addslashes($existingforum->intro);
        $existingforum->course = $courseid;
        $existingforum->is_lams = 1;
        $existingforum->oldid = $existingforum->id;
        $existingforum->id = insert_record('forum', $existingforum);
    }

    $module = get_record('modules', 'name', 'forum');
    $section = get_course_section($sectionid, $courseid);

    $cm->course = $courseid;
    $cm->module = $module->id;
    $cm->instance = $existingforum->id;
    $cm->added = time();
    $cm->section = $section->section;
    $cm->is_lams = 1;
    $cm->id = insert_record('course_modules', $cm);

    // clone forum discussions
    $newdiscussions= get_records("forum_discussions", 'forum', $existingforum->oldid);
    //add discussions and their posts
    add_discussions_instance($newdiscussions, $existingforum->id, $courseid);
    return $cm->id;
}
```

This function is called from LAMS when it needs to create a new instance of a Moodle Forum based on the content of *another* instance. For instance when the teacher in LAMS creates a sequence that includes a Moodle Forum activity, then she creates a new LAMS lesson, LAMS needs to create a separate new instance of the Moodle Forum tool that the teacher created (because we don't want to modify the original design so the teacher can reuse this sequence elsewhere). Same if the teacher creates a second or third LAMS lesson based on the same sequence. For each new LAMS lesson, new instances of Moodle Forum Tools will be created. The LAMS tool adapter calls the *forum_clone_instance* when it needs to create new copies.

moodle/mod/forum/lib.php

```
/**
 * LAMS Function
 * Serialize a forum instance and return serialized string to LAMS
 */
function forum_export_instance($id) {
    $cm = get_record('course_modules', 'id', $id);
    if ($cm) {
        if ($forum = get_record('forum', 'id', $cm->instance)) {

            $discussions= get_records("forum_discussions", 'forum', $forum->id);
            // serialize forum into a string; assuming forum object
            // doesn't contain binary data in any of its columns
            $all=array($forum,$discussions);
            $s = serialize($all);

            header('Content-Description: File Transfer');
            header('Content-Type: text/plain');
            header('Content-Length: ' . strlen($s));
            echo $s;
            exit;
        }
    }

    header('HTTP/1.1 500 Internal Error');
    exit;
}
```

Exports a Forum object into a file and sends it to LAMS so it can export it with a LAMS sequence (see [Export and Import Animation](#))

moodle/mod/forum/lib.php

```
/**
 * LAMS Function
 * Deserializes a serialized Moodle forum, and creates a new instance of it
 */
function forum_import_instance($filepath, $userid, $courseid, $sectionid) {
    // file contents contains serialized forum object
    $filestr = file_get_contents($filepath);

    $all=unserialize($filestr);
    $forum = $all[0];
    $discussions= $all[1];

    // import this forum into a new course
    $forum->course = $courseid;

    // escape text columns for saving into database
    $forum->name = addslashes($forum->name);
    $forum->intro = addslashes($forum->intro);

    if ( ! $forum->id = insert_record('forum', $forum) ) {
        return 0;
    }

    $module = get_record('modules', 'name', 'forum');
    $section = get_course_section($sectionid, $courseid);

    $cm->course = $courseid;
    $cm->module = $module->id;
    $cm->instance = $forum->id;
    $cm->added = time();
    $cm->section = $section->id;
    $cm->is_lams = 1;
    $cm->id = insert_record('course_modules', $cm);

    //add discussions and their posts
    if($discussions!=''){
        add_discussions_instance($discussions, $forum->id,$courseid);
    }
    return $cm->id;
}
```

This does the import of a LAMS sequence that contains a Moodle Forum. (see [Export and Import Animation](#)).

moodle/mod/forum/lib.php

```
/**
 * LAMS Function
 * Return a statistic for a given user in this Moodle forum for use in branching
 */
function forum_get_tool_output($id, $outputname, $userid) {
    $cm = get_record('course_modules', 'id', $id);
    if ($cm) {
        $posts = forum_get_user_posts($cm->instance, $userid);
        switch ($outputname) {
            case ("learner.number.of.words"):
                $numwords = 0;
                foreach ($posts as $postid=>$post) {
                    $numwords += count(explode(' ', $post->message));
                }
                return $numwords;
            case ("learner.number.of.posts"):
                return count($posts);
        }
    }
    return 0;
}
```

This function deals with the outputs that are required to do Branching.

moodle/mod/forum/lib.php

```
/**
 * LAMS Function
 * Return an HTML representation of a user's activity in this forum
 */
function forum_export_portfolio($id, $userid) {
    global $CFG;
    $text = 'This tool does not support export portfolio. It may be accessible via ';
    $text .= '<a href="'. $CFG->wwwroot. '/mod/forum/view.php?id='. $id. '">this</a> link.';
    return $text;
}
```

This function handles the export portfolio as is expected in LAMS. In this case, we only provide a link to the webpage that host the forum. Eventually it would be nice to have all the postings in the forum as an html file so the student (or teacher) can take home with them.

And that's all!

LAMS Integration Interface

[Completed](#), but not documented yet. Coming soon...

CVS, Code and Status

The code is available in our [CVS server](#). If you want to download the code, here are the [CVS details](#). As of 22/08/2008, this code is consider in Beta. Further testing is required.
