

# Automatic forum translations with moderation/ message editing in Moodle (“tradauto/Moodle”)

Remark: the code examples used here are based on the first beta version of tradauto/Moodle from 2009.

The development mailinglists used until now can be read here as well:

- [tradauto-des@funredes.org](mailto:tradauto-des@funredes.org)
- [soporte@funredes.org](mailto:soporte@funredes.org)

## Rationale

For the Learning Management System (LMS) [Moodle](#), there is automatic translation software available, but this does only on-the-fly translations for each request or page view. For efficient online moderation of forum discussions, instead it is necessary to store translations in a persistent manner, and moderate them if necessary, as machine translations are far from perfect.

## First Specification

A configuration interface is needed for each forum, to specify the language and forum, to which the original message should be translated:

Original language forum	Target language	Translated target forum
Foro social  Foro abierto a todos los tem ▾	en	Social forum An open forum for chatting ▾ modify delete
Foro social  Foro abierto a todos los tem ▾	fr	Forum informel Un forum pour discuter de ▾ modify delete
Foro social  Foro abierto a todos los tem ▾	es	Foro social Foro abierto a todos los tema ▾ modify delete
Assign a new forum translation:		
Novedades Novedades y anuncios  ▾	fr	Novedades Novedades y anuncios  ▾ Create

*Example from the current implementation*

If a new post happens, the moderators of that forum should be alerted by email, and a moderation link in that mail should be provided.

After moderating/ editing the post of the original message, the moderated message should be automatically translated, the translation should be available online on Moodle for the community or public.

## Implementation

Looking for a public, legal and easy to use translation service, [Babel Fish](#) has been selected. It provides two modes:

1. Translation of plain text (no other restrictions than text size)
2. Translation of web pages/ HTML (restricted; user IP has to be different from web page IP, size limit is tolerant)

As the Moodle forum messages are in HTML format, and often it is useful to maintain the educative format of messages, the second mode has been chosen.

As a consequence, it was **necessary to use two servers** for the implementation:

1. The server with Moodle and forum contents.

2. Server which gets transmitted automatically new moderated forum posts as individual pages with separate URL's, which then are requested to translate by Moodle.

The implementation on the second server is very easy and has very low requirements in terms of system resources. Please [see the code here](#), that index.php file just has to be put in some server, and correctly configured in both Moodle and itself, using the allowed client IP's (of Moodle server) and the URL where it is installed. The restriction is not yet complete after IP checking, a simple "exit;" line is lacking in case of failure of IP check.

#### Database modifications

The first implementation used a secondary "zero-administration" database based on SQLite, which is normally included in PHP installations. In the time of this writing, the necessary database additions are made in the same database in which Moodle stores its data (which is normally MySQL). There is still code executed for the secondary database, for example for the dynamic creation of that database with SQLite. This is obsolete now.

## Integration of the new translation functionality into Moodle

Moodle's native way to react on new forum posts is a "cronjob", a periodically executed program, which is "admin/cron.php" in the moodle installation directory. In Moodle's API, the relevant function is for us `forum_cron()` in the file `mod/forum/lib.php`. This, together with the activation of the configuration interface, are the only places where new code has been added inside existent Moodle functions. All other code lives in new PHP functions explained in the following paragraph.

But, as some core functions had been modified, the new functionality can't be distributed by an extension on the Moodle plugin site, it has to be published as a patch (thankfully this is possible).

## Integration API functions

### Functions of configuration interface:

[`selectbox\_forum\(\$defaultid, \$formname\)`](#)

Muestra las opciones de foros y sus respectivos lenguajes.

[`envprint\(\)`](#)

optional debugging utility to display i.e. POST or GET variables.

### Functions for message translation:

[`moderation\_db\_create\(\)`](#)

Dynamic creation of secondary database tables, obsolete

[`moderation\_workflow\(\)`](#)

Triggered via `forum_cron()` by the cronjob; here the actions are taking place.

[`forum\_moderateplease\_email\(\$post\)`](#)

Emitting alert emails to forum moderators

[`sql\_escape\(\$string\)`](#)

Converting arbitrary text to SQL-quoted text (this is still SQLite/ SQL92 standard-like, MySQL/PHP work different).

[`forum\_translate\_post\(\$postid\)`](#)

Triggering translation of a forum message

[`LanguageOf\(\$htmltext\)`](#)

Returns the supposed language of the message. Be aware, as this is statistics-based, the error enhances very much with short messages, and above all small subject lines

[`Html2RoughPlaintext\(\$html\)`](#)

Not so sure: I think this is for the emailing process, where HTML would be problematic

[htmlupload\(\\$html\)](#)

Upload of the original message as a HTML page to the second server, as mentioned above

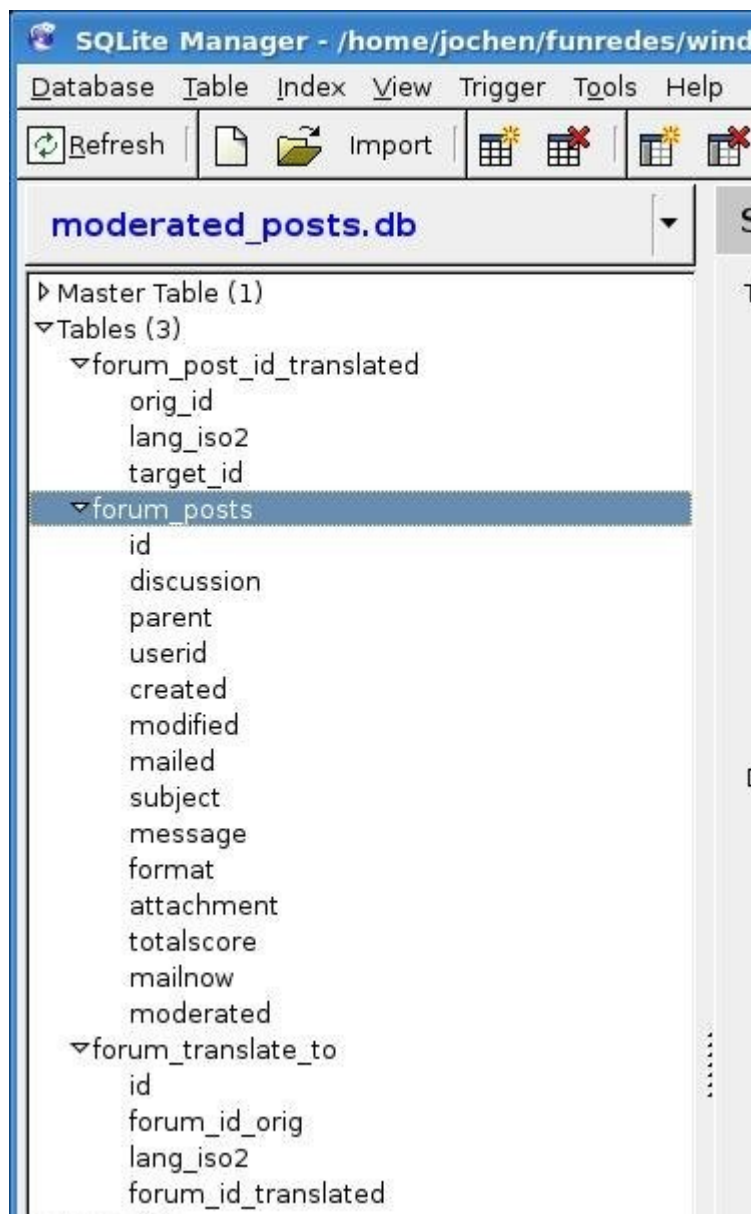
[babelfishtranslate\(\\$html, \\$TargetLanguage\)](#)

Translation of the uploaded message HTML page.

[babelfish\\_html\\_cleanup\(\\$html\)](#)

The Babelfish result contains strange links and other additional code. This function tries to get rid of that.

## Additional database table definitions



## Design decisions, magic numbers, special definitions, overview

When the cronjob is called (on the first production machine funredes.org this is the case each minute), Moodle's table "forum\_posts" is revised for new messages. The indicator for the status of a message is the column **mailed**. If the message is new, then this has the value 0. The function moderation\_workflow() is then called, which copies its content to another "forum\_posts" table (with this name in SQLite, or with the new approach within MySQL with the table name

"trans\_forum\_posts"), leaving just "This message is being moderated, please be patient" in the original forum, and the "mailed" column is changed to the magic number 42.

Then, the copied contents are translated, after successful translation trans\_forum\_posts.moderated column is set to 1, and the original message is set back, together with the posting of the translated messages, these translated messages are sent to their respective forums.

The thread structure is stored in the discussion.id values, which are part of each message/ post. As the discussion table has as well a parent column, the thread tree structure is stored by the discussion ID information.