

# eNOSHA Moodle Module

## Technical Reference

Version 1.0

Isuru Balasooriya ([irb@ucsc.cmb.ac.lk](mailto:irb@ucsc.cmb.ac.lk))

# Table of Contents

1. Chapter 1 - Introduction
  - 1.1 About eNOSHA module
  - 1.2 About this document
  
2. Chapter 2 - Design Overview
  - 2.1 Introduction
  - 2.2 Conventions
    - 2.2.1 General
    - 2.2.2 User Interface
    - 2.2.3 PHP Code
    - 2.2.4 Database
  - 2.3 Technologies
  - 2.4 Installation
  - 2.5 Configuration
  - 2.6 Module Usage
  
3. Chapter 3 - Module Design
  - 3.1 Module Design
  - 3.2 Module File Structure
  - 3.3 Module Development
    - 3.3.1 PHP – XMLRPC
    - 3.3.2 HTML, Javascript and AJAX
      - 3.3.2.1 AJAX
      - 3.3.2.2 Ajax Handler
  
4. Chapter 4 – Module Functionality
  - 4.1 Login
  - 4.2 Search / Advanced Search
  - 4.3 Import
    - 4.3.1 Object Preview
    - 4.3.2 SCORM
    - 4.3.3 SCORM Preview
  - 4.4 Import Action
  
5. Chapter 5 – Obtaining More Information

# Preface

The NeLC project is to build up and maintain a sustainable National e-Learning Centre (NeLC) in Sri Lanka that will be a mechanism for the use of ICT for national development as to visualize the e-Sri Lanka program. The NeLC would do so by the promotion of teaching, learning and research for exploring and formulating new and innovative content development, delivery and evaluation exploiting the benefits of ICT.

eNOSHA moodle module is a open source component for moodle learning management system developed in collaboration between: University of Colombo School of Computing (UCSC) , Sri Lanka under the NeLC project, Stockholm University and Orebro University, Sweden.



# Chapter 1. Introduction

## 1.1 About eNOSHA Module

The e-learning Center at the University of Colombo School of Computing creates and publishes a large amount of learning material for its internal and external degree programs. The eNOSHA learning object repository was built to address the need for a consistent and stable repository to store and retrieve all that content. But since moodle was already an established learning management system in the institute, the need to combine moodle with eNOSHA arose, in which the result was the moodle module.

eNOSHA module will enable moodle users, especially teachers in courses to import learning objects from available enosha instances locally or through the web. The refined searching and advanced searching facilities will help the user to better locate and reuse content that are already available.

## 1.2 About this document

- Code samples, filenames and directories are presented in a courier typeface.
- Square braces are used in code samples, filenames and URLs to indicate a sample value: for example, `[anything]action.php` can be interpreted as any file name ending in `action.php`.
- Directories are presented as `/[anything]`
- The URL <http://www.enosha.com> used in many examples is intended as a fictional illustration only.

# Chapter 2. Design Overview

## 2.1 Introduction

eNOSHA moodle module is a approach at making it easy for current moodle users to make use of resources already available through eNOSHA repositories. In a user perspective, the module acts as an activity which the user can add to a course. The user beforehad must be a user of the particular external repositories he/she is going to use.

In a developer's perspective, building of a moodle module is made easy with the template provided by moodle.org, NEWMODULE. It is practically a skeleton in which the developers can build on required functionanlities. eNOSHA moodle module was also built on the NEWMODULE and customized according to eNOSHA repository structure. 3 basic needs were identified with the module.

- Login/Authentication
- Search for learning objects
- Import

These were new functionalities handled outside the moodle core. This led to the concept of remote procedure calls, where XMLRPC was utilized. The eNOSHA repository would act as a server, and the moodle module will be the client. The requests will be sent via XMLRPC and results will be retrieved and displayed. Almost all the resource types on eNOSHA will be classified as 'Adding a resource' to a course, except for SCORM packages. They will be considered as 'Adding SCORM/AICC Packages' to a course. The user is automatically redirected appropriately according to the file they import.

## 2.2 Conventions

### 2.2.1 General

- Directories are named using lower case, names longer than one word are joined with an underscore ('\_')

### 2.2.2 User Interface

- Layout is designed using Cascading Style Sheets (CSS)

### 2.2.3 PHP Code

- Methods are named in lower case, names longer than one word are joined with an underscore ('\_')

- `<?php ?>` is to delimit PHP code, not `<? ?>`
- Code commenting conventions
  - `//` A one-line comment
  - `/**`
    - \* Multi-line comment like they are written
    - \* to work with the javadoc documentation
    - \* `/`

## 2.3 Technologies

Recommended server configurations:

- PHP support (4.2.x or later)
- MySQL (3.23.23 or later)
- Apache (1.3.2x or later)
- Linux, Windows operating systems

## 2.4 Installation:

- The module is installed as any other moodle module is installed. It should be copied in to the `/mod` directory of the moodle directory.
- Then go to moodle homepage and the notification page.
- The module should install the database tables and the other necessary settings.
- If no problems are encountered, a successfully installed message and a continue button should be prompted.

## 2.5 Configuration:

- The moodle module for eNOSHA needs to be configured by an administrative user.
- Go to moodle homepage, on the left hand administrative menu, **Modules -> Activities**.
- The 'Learning Object' link should appear in the expanded list of activities.
- The settings page for the eNOSHA module will be displayed.
- External Repositories textbox: The external eNOSHA repositories which the module should access.
- Full paths should be given, comma separated for multiple URLs.
  - eg. `http://www.enosha.com, http://anotherenosha.com`

## 2.6 Module Usage:

- Once the module is activated, it can be accessed through a course in moodle.
- As a user with rights to edit a course, login to the moodle system.
- Browse to a course in which you have the editing rights.
- In the upper right corner, click on the button 'Turn Editing On'.
- Once the editing mode is enabled, go to a specific week and click on the dropdown 'Add an Activity'
- In the list, click on 'Learning Object'
- In the page that follows, user is prompted to enter a title for the learning object
- Then click on the button 'Locate a File in eNOSHA'.
- A popup window is opened with a login form, where the administrator defined external eNOSHA repositories are listed.
- With a correct username and password provided for the selected eNOSHA repository, the user can log in.
- User can then search for a learning object and import to the moodle course.
- According to the imported file type, the user is redirected to either a moodle default 'Resource' or a 'SCORM' upload page, with the previously entered title and the imported file name. The user then can fill in any necessary information about the learning object and add it to the course.

# Chapter 3. Module Design

## 3.1 Module Design

eNOSHA module has been derived from free and open source NEWMODULE template provided by the moodle development community.

The remote function calls are developed using XMLRPC methods.

## 3.2 Module File Structure

Apart from the default module scripts, these additional scripts have been written.

<b>New Source code</b>	
login.php	The login script with an HTML form
logout.php	Script to unset the logged in sessions
search.php	HTML form for the searching and Advanced searching
import.php	Displays the file selected for importing, with preview
import_action.php	PHP script to physically import the learning object and display a confirmation
ajax_handler.php	Script to send XMLRPC based AJAX calls
<b>Modified from the original source</b>	
mod_form.php	The main index page for the module
settings.php	The administrative settings page of the module

**Table 3.1**



## 3.3 Module Development

### 3.3.1 PHP - XMLRPC

XMLRPC employs the use of XML-RPC EPI PHP extension, located in `utils.php` inside the module.

All the scripts that use XMLRPC requests should include `utils.php`.

#### Function Structure:

```
$result= xu_rpc_http_concise(  
    array(  
        'method' => "myMethod",  
        'args' => array($arg1,$arg2),  
        'host' => $host,  
        'uri' => $uri,  
        'port' => 80  
    )  
);
```

A function as illustrated above calls for the remote function.

The remote function should return a result array, which is assigned to the variable `$result`.

The resulting array can then be used in the local script.

i.e. `$img = $result[0]` assigns the first value of the returned array to a variable named `$img`.

The remote function is located using the variables, `$host` and `$uri`.

- `$host`: The host server which contains the script with the remote procedures

e.g. `http://www.enosha.com`

- `$uri`: The absolute path to the script with the remote procedures

e.g: `http://www.enosha.com/xmlrpc.php`

## 3.3.2 HTML, Javascript and AJAX

The user interface employs the use of HTML, Javascript and a little AJAX.

### 3.3.2.1 AJAX

- AJAX is used to display the sequential drop down boxes in the advanced search, namely 'Category' and 'Format Type' elements.
- Direct AJAX request are not allowed between different domains, thus the `ajax_handler.php` script is used.
- The selected values of the dropdown boxes are sent as XMLRPC requests (Using the `ajax_handler.php` script) to the remote server and the sub level selection query is retrieved and displayed.
  - e.g. A Category 'Multimedia' will only list it's subcategories 'Images', 'Videos'.

### 3.3.2.2 Ajax Handler

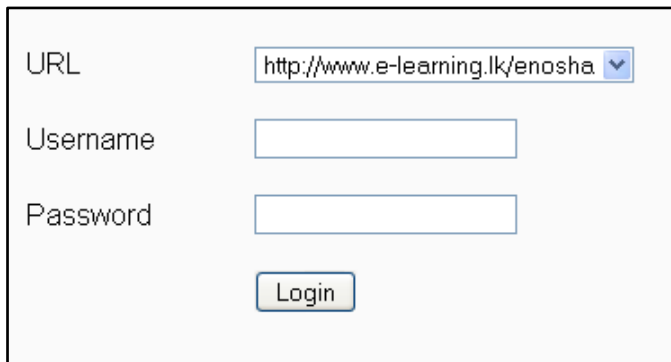
- 6 `$_GET` variables are used to keep track of the changes in the dropdown boxes.
- According to the dropdown box changed, the values are sent through the `ajax_handler.php` to the eNOSHA server.

# Chapter 4. Module Functionality

eNOSHA module has 3 major divisions regarding the nature of functionality.

- Login/Authentication
- Search/Advanced Search
- Import

## 4.1 Login



The screenshot shows a login form with the following elements:

- URL:** A dropdown menu with the value "http://www.e-learning.lk/enosha" and a downward arrow.
- Username:** A text input field.
- Password:** A text input field.
- Login:** A button with the text "Login".

- The result query will return either '1' or '0' for login XMLRPC request.
- If the result is '1' which is for authenticated login, the user is redirected to the search page.
- For authenticated users, the following session variables are set.

```
$_SESSION['url']  
$_SESSION['fullurl']  
$_SESSION['username']  
$_SESSION['pass']
```

- For the dropdown box with accessible external repositories, the \$CFG variable of moodle is used.

```
$external_servers = $CFG->external_servers;
```

## 4.2 Search/ Advanced Search

Search  Advanced Search

Atom  
 Collection of Atoms  
 Course Module  
 Full Course

Format Type

Catalog Type

Modifiable  Yes  No  
Audience  Internal  External  
Target Region   
Language   
Published Status   
Activity Type

- All search queries are authenticated with the saved login data, thus illicit use of the function is avoided.
- Simple search consists of a free-text box.
- The input value `searchtxt` is sent to the remote `xmlrpc.php` on the server and the search query is processed.
- Advanced Search consists of dropdown boxes and checkboxes.
- The dropdown boxes are populated with values from the remote server, retrieved through XMLRPC.
- All dropdown boxes are printed by calling the remote method `print_dropdown` with the argument with the specific dropdown box needed. The argument is processed in the server script `xmlrpc.php` and the required result is returned.

e.g. For listing Format types

Format Type

Catalog Type   
Flash Files   
Text files   
sql files   
Images   
Audio   
Presentation

```

$drop1= xu_rpc_http_concise(
    array(
        'method' => "print_dropdown",
        'args' => array('format_type_drop'),
        'host' => $host,
        'uri' => $uri,
        'port' => 80
    )
);

```

- Search results are returned as an array and displayed with radio button for continuing with a selection. User can import the selected object as a next step.

Select	Title	Description
<input type="radio"/>	What is programming?	What is Java programming?
<input type="radio"/>	Nature of Java	Nature of Java
<input type="radio"/>	Foot path of Java	Foot path of Java

## 4.3 Import

- Once the id (`$lo_id`) of the selected learning object has been passed to the `import.php`, it retrieves the full data row from the remote database and displays.

### 4.3.1 Object Preview

- Import method in the remote `xmlrpc.php` returns the array of object details row with some additional information, one being the object preview.  
`$file = $weburl."/lib/download/preview.php?filename=file&type=".$type."&path="/".$download_path;`
- The preview URL is arranged in the eNOSHA server and passed through the return array, which in turn is displayed in `import.php`.

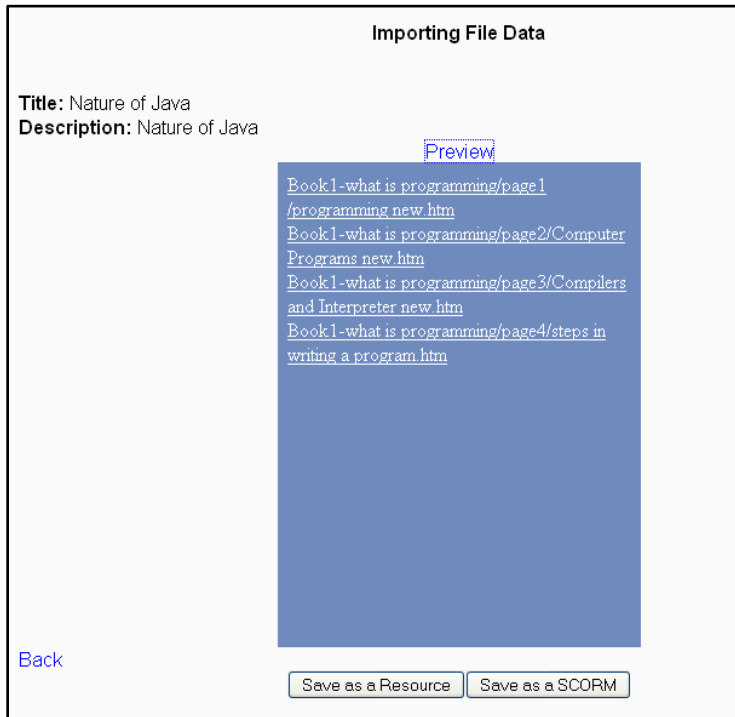
### 4.3.2 SCORM

- While importing zip archives, the script `xmlrpc.php` determines if the archive is a SCORM compliant archive or a regular zipped archive.
- It lists all the files in the archive first and checks for an `imsmanifest.xml` on the list.
  - `if(in_array("imsmanifest.xml", $files)) $scorm = 1;`
- This value `$scorm` is also passed on for `import.php` and if the value '1' has been set, it will display a button for importing it as a SCORM object.

- else, it will only display a 'resource' type button.

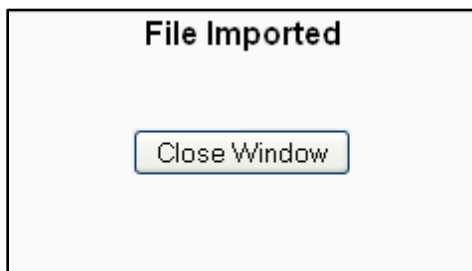
### 4.3.3 SCORM Preview

- Preview for SCORM is also handled via,  
`$file = $weburl."/lib/download/preview.php?filename=file&type=".$type."&path="/".$download_path;`
- Where, `preview.php` in eNOSHA will handle the previewing process and return only the file list of the SCORM package. Users can click on the hyperlinks and preview the contents.



## 4.4 Import Action

- `import_action.php` handles the physical importing of the learning object.
- `file_get_contents` method is used to read the file from the server .
- On a failure, prints 'Not Done' and a link back to search page.
- On a successful import, the following window is displayed.



- Once a file is imported, user is redirected to the moodle default page for adding a resource or a SCORM package.
- For a resource, (Which is any learning object except for a SCORM package),

```
onclick=post_value(1,$course,$section,$return,'".$weburl."');
```

- The javascript method, redirects to the moodle page for uploading a resource.  

```
window.opener.location.assign(value4 +  
"/course/modedit.php?add=resource&type=file&course=" +  
value1+"&section="+ value2 +"&return="+value3);
```
- The page will be autofilled with the title the user previously entered for the learning object, and the imported file name.

## Chapter 5. Obtaining More Information

For more information visit eNOSHA project page through [here](#).

There is an eNOSHA discussions group through [here](#).