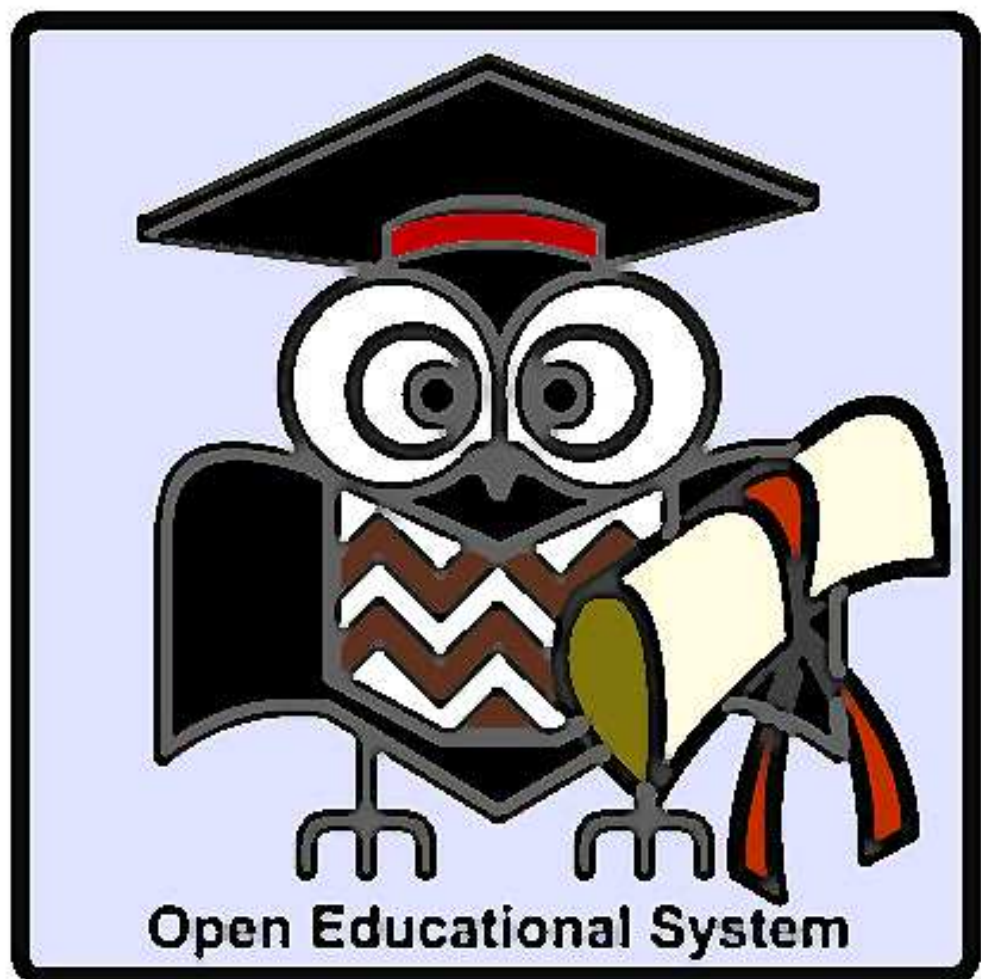


Final Year Project Report

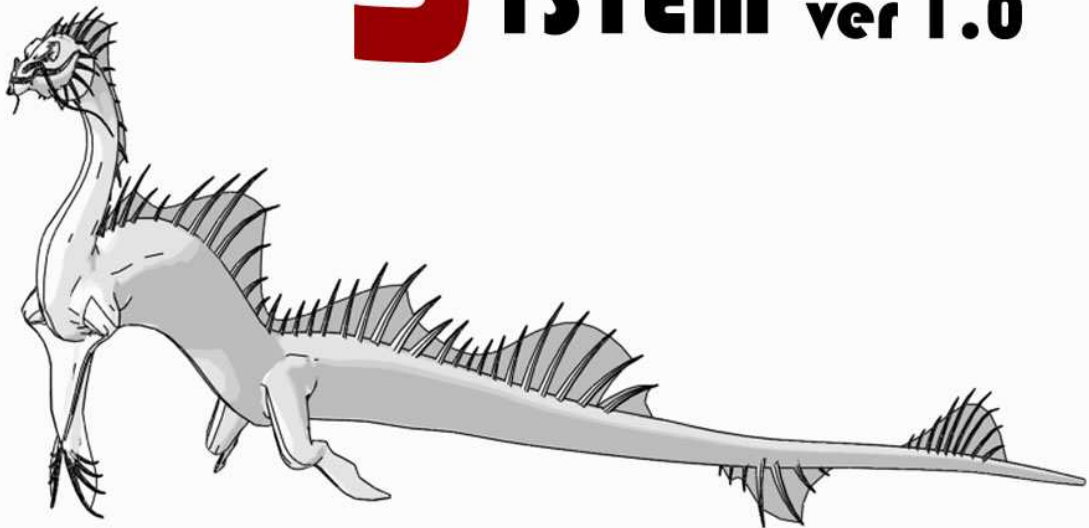
Title – Internet Application



— First Edition —

CD includes
all source code

Open **E**ducational **S**ystem ver 1.0



Danny & Chai

CD-ROM
included

Preface

Thanks for your interesting in this project report. This is a final year project report written by two students who study in the Department of Electronic and Information Engineering (EIE) in the Hong Kong Polytechnic University. We are Ho Wing Chuen and Hui Ka Chai, both of us are studying in High Diploma of Electronic and Information Engineering.

This is a technical report, We will explain the idea of our project and the implementation process in detail and systematically. So as to enable you to understand our project more clearly. This report is divided into six chapters; each chapter with a specific focus. We hope this approach allows you can read our project report easily. Lastly, if you have any comment on our project, please feel free to contact us.

Ho Wing Chuen (Danny) & Hui Ka Chai
Dep. of EIE (HD-42075)
May of 2003

Contents

Preface	X-I
Abstract	X-II
Acknowledgements	X-III

Chapter 1 : What is OES ?

1.1	Background.....	(P. 9)
1.2	Objectives.....	(P. 10)
1.3	Methodology	(P. 12)
1.4	Limitations	(P. 14)
1.5	Organization of this report	(P. 15)

Chapter 2 : OES's Architecture

2.1	Structure of OES.....	(P. 18)
2.2	System Design (Programming technique).....	(P. 26)
2.2.1	Modules Loader	(P. 26)
2.2.2	File Loader	(P. 30)
2.2.3	Directory Structure	(P. 40)
2.3	Programming technique on Kernel.....	(P. 47)
2.3.1	Themes Loader	(P. 51)
2.3.2	Login Processor	(P. 56)
2.3.3	Modules Loader	(P. 60)
2.3.4	Template Loader.....	(P. 60)
2.3.5	SESSION	(P. 61)
2.4	Programming technique on Modules	(P. 64)

Chapter 3 : Modules on OES

3.1	Flash Board	(P . 84)
3.1.1	ActionScript of the Flash Board.....	(P . 88)
3.2	Plot Graph.....	(P . 94)

3.2.1	ActionScript of PLOT GRAPH	(P. 96)
3.3	Plot Polar	(P . 99)

Chapter 4 : Getting start with OES

4.1	Installation of OES	(P. 101)
4.1.1	Download the OES.....	(P. 101)
4.1.2	Install OES on Linux	(P. 102)
4.1.3	Start OES in browser	(P. 105)
4.2	Administration Guide	(P. 105)
4.2.1	Change System Default Theme.....	(P. 106)
4.2.2	Update the system LINK.....	(P. 118)
4.2.3	Create / Modifier the administrator Account	(P. 119)
4.2.4	Another Theme on admin control page	(P. 122)
4.3	Adding Modules	(P. 123)

Chapter 5 : Further Improvement

5.1	Conceptually module.....	(P. 125)
5.2	Implementation on user side.....	(P. 127)

Chapter 6 : Conclusion

6.1	Conclusion	(P. 129)
-----	------------------	----------

Bibliography

Appendix

- (A) – List of OES Library (API)
- (B) – Helpful Tools
- (C) – List of the Figures and Tables
- (D) – Source Code

Abstract

One primary aim of this project is to develop a next generation online educational platform – Open Educational System (OES). The platform should allow students to log in the system, collect their learning materials, and discuss with their classmates, doing online test... . The major advantage is that, the entire things process is real-time and online, which means the student can study everywhere (e.g. at home). Thought the use of advanced web technology, the administrator can add a new module to the system to provide an extra function (Plug-and-Play modules) without upgrading the entire system. The multi theme (User interface) and multi language design allow different national students using the same system without the problem on language.

Acknowledgement

The project would not have succeeded without the help of the following people

First of all, we wish to thanks Dr. Siu-Chung Wong, for his patience, guidance, support and understanding. His suggestions were invaluable, which enabled us to overcome the difficulties we encountered during the project. There were guidelines and specific requirements for the project; yet Dr. Siu-Chung Wong encouraged and respected creativity on our part.

Thanks are also due to the PHP developer, Macromedia Flash developer, Apache developer, MySQL developer, ADODB developer, Linux developer and the Users Although we are to unable to name the individuals, their help is warmly appreciated.

Chapter 1

What is OES ?

Chapter 1 consists of five sections: Background, Objectives, Methodology, Organization of this report and the Limitations.

This chapter will describes the background to this project, lists the objectivities, outlines the methodology, examines the limitations and preview the organization of the project.

1.1 Background

The title of the final year project is calling Internet Application. From the name of title has already implied that it is related on Internet. The requirement of this project is writing an application software which can be running on Internet and perform some actions.

For example, the project can create a Discussion Board which can allow users discussion on-line; write a Bidding System to allow the bidder bid their interesting things, and so on. But this project is not the easy topic list on above. It is a new and difficult to implement system. The name of this system is calling *Open Educational System (OES)*. This is a new generation online educational platform; this will include much new technology on it. For example, it has ability to running on most machines with different design of operation system, and it is multi language support and can be installed or uninstalled modules mechanism. This is the background of this project.

1.2 Objective

Providing the information is an effective way and important in Hong Kong. Recently, the Hong Kong Government has encouraged the University, Secondary and Primary Schools to implement the use of E-learning. It means that the further studying will related on an electronic media, such as On-line Exam, studying in Electronic Book. There is no doubt that, the role of E-learning will more and more important especially for educational. Due to this trend, OES is the platform to help the students getting easier to studying in the Internet. This is the purpose of this project.

To achieve this mission, the system was developed into two parts. The first part is the administrator; another is normal user (for the students). The system should be able to provide two interfaces to allow these two types of users log in. The administrator should be able to control the whole system, including create/edit or delete the user account, add/drop the function on the system, and so on. So he/she will control the system by log in to administrator only page. The student is allowed to obtain the information provide by the administrator. For examples, the administrator can uploads the notes and the

announcement; the student can download it and getting the studying news.

This is the concerned of the system, which is simulator to current used by most university – WebCT. But this system will be must powerful and user-friendly than that.

1.3 Methodology

First of all, to implement the OES requires a web-server to deliver the information from the system to the client. The server will running many server programs, so a stable environment is required. Linux was chosen to be the operation system running the OES.

OES is designed to be able on running different operation system (OS). This is because cross-platform designed software is more expected in the market. And it is difficult to expert the user use a specific OS, so using a most general programming language that is supported in most OS is one of the considerations. This is the result of this project mainly program by PHP script.

PHP is the hit web programming language in this few years. The advantage is, the PHP script can be run on most operation system although the architecture of hardware and OS is totally different. That mean the software can be OS and hardware independence. Another major advantage is, it is free of charge. The server system only require to install the PHP run-time support and then do a little bit configuration, their web server will be support it. Also the performance of PHP script is very good, when

compare with PERL and ASP, especially running on Unix/Linux system.

OES is a modules base system, so the design of the system should be base on basic modules as possible. There are several advantages of write the program into modules structures. The programmer can add or remove the function (modules) more easily. The whole system should not require to modifier anything before or after add or delete the modules. The programmer only considers how to programming with modules, but they do not require to understand the deep of OES. And the codes are easier to read by other person.

And also OES is design to Multi-Language and Multi-Interface. Since the Internet is international. Any system running on the Internet is required to internationalization to allow the most reader getting easier to understands and get the most update information without the language problem. So the design of OES must provide a mechanism that allows displaying on different language and the user can select their preferred language.

1.4 Limitations

There are some limitations in the developing of OES.

Time Limitations & Consuming :

The difficulties when developing OES, is the “Time Limitations & Consuming”. Design the modules are basic on the structure and the rule of OES system. So decision and design the architecture is very important. It needs to take a relatively long time to construct the kernel part then the modules, it is a time consuming job. And there is difficult to set the finish point of OES. Because OES will grow continuously, it can develop a huge number of modules what people wanted and improve the system better and better. But this project is a one-year job, so it is not possible to be finish within a year. Developing OES is a long running process, and the system will be grown up by time.

Resource :

OES is a new concept of system design, so there is very difficult to find any reference or resource from before, all the thing are create by creativity and start from the beginning.

1.5 Organization of this report

This report is mainly divided into six chapters - Chapter 1 : What is OES?; Chapter 2: OES's Architecture; Chapter 3: Modules on OES; Chapter 4: Getting start with OES; Chapter 5: Further improvement; and Chapter 6: Conclusion. After these six chapters is the Appendix.

In the chapter 1, will introduce the basic idea of OES, and the goal of OES.

In the Chapter 2, will go into more deeply, such as the system architecture, the kernel design and the programming technique used; such as Modules Loader, Theme Loader etc. And also this is the most important chapter for people who are interested in the structure of OES.

Chapter 3, will introduce the design process of the modules, which come with the OES (i.e. the Modules on category 'Modules').

Chapter 4, will teach how to getting start with OES after downloading it from the official website and the installation process.

Chapter 5, will discuss on how the further improvement and

extension of OES.

Chapter 6, will make a conclusion of this project (OES).

Finally, is the Appendix. There will contain all the information, which is related to OES. Such as the API of OES, program sources, Linux configuration guide, and the tools are used.

Chapter 2

OES's Architecture

Chapter 2 consists of four sections: Structure of OES, System Design, Methodology, Programming technique on Kernel and Programming technique on Modules.

This chapter will describe the details structure of OES and the technology used. And also, this chapter will show the steps of system design, and the problem solving technique. So this is the most important chapter in the report.

2.1 Structure of OES

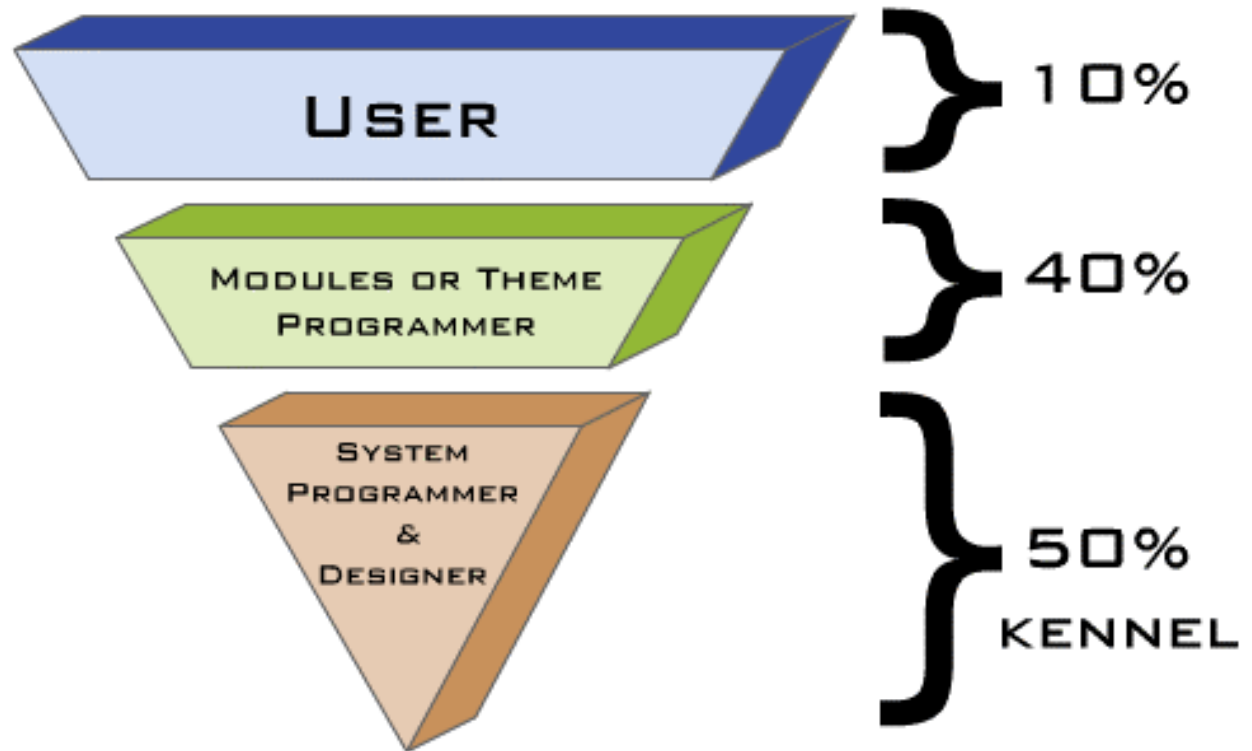


Fig 2.1 – Scope of OES

Figure (Fig 2.1) shows that, OES can be separated into three parts, each part can be independence on each other. The user only knows about 10% about the system, although they think OES is powerful. For the modules or theme programmer, they will know an extra 40% of the system. They can develop application software (modules) for the OES, by the extra 40% knowledge. The last one is the system designer and programmer. They must know 100% of the system. Because he/she are the most important person who design and develop with the system. If he/she cannot

clearly to understand the whole system, he/she is hardly to provide facility and the rule for the other programmer program with OES.

The major work for the System Programmer is writing the kernel for the system, which is the hardest but the important part. Without their work, the OES will not function. The kernel is the core of the system, which provides all the function/routine for the modules (application software). It also describe the system structure, how the system to perform an appropriate action and how to collect the information from the user. The system programmer also need to design the system architecture, designing a set of rule also writing a set of API (list on Appendix A) for the “Modules or Theme Programmer” and the “Normal Users” to follow.

Modules or Theme Programmer must base on the set of rule designed by System Programmer. He/she can use the set of API to create the modules or theme. For the new modules or theme will be announce in OES official website on future (<http://openeducation.sourceforge.net/>).

Users are the major clients; so their needs are the most

important things. Interview with them is the fast and easy way to collect their comments and requirements. The information is helpful to allow OES improve on future.

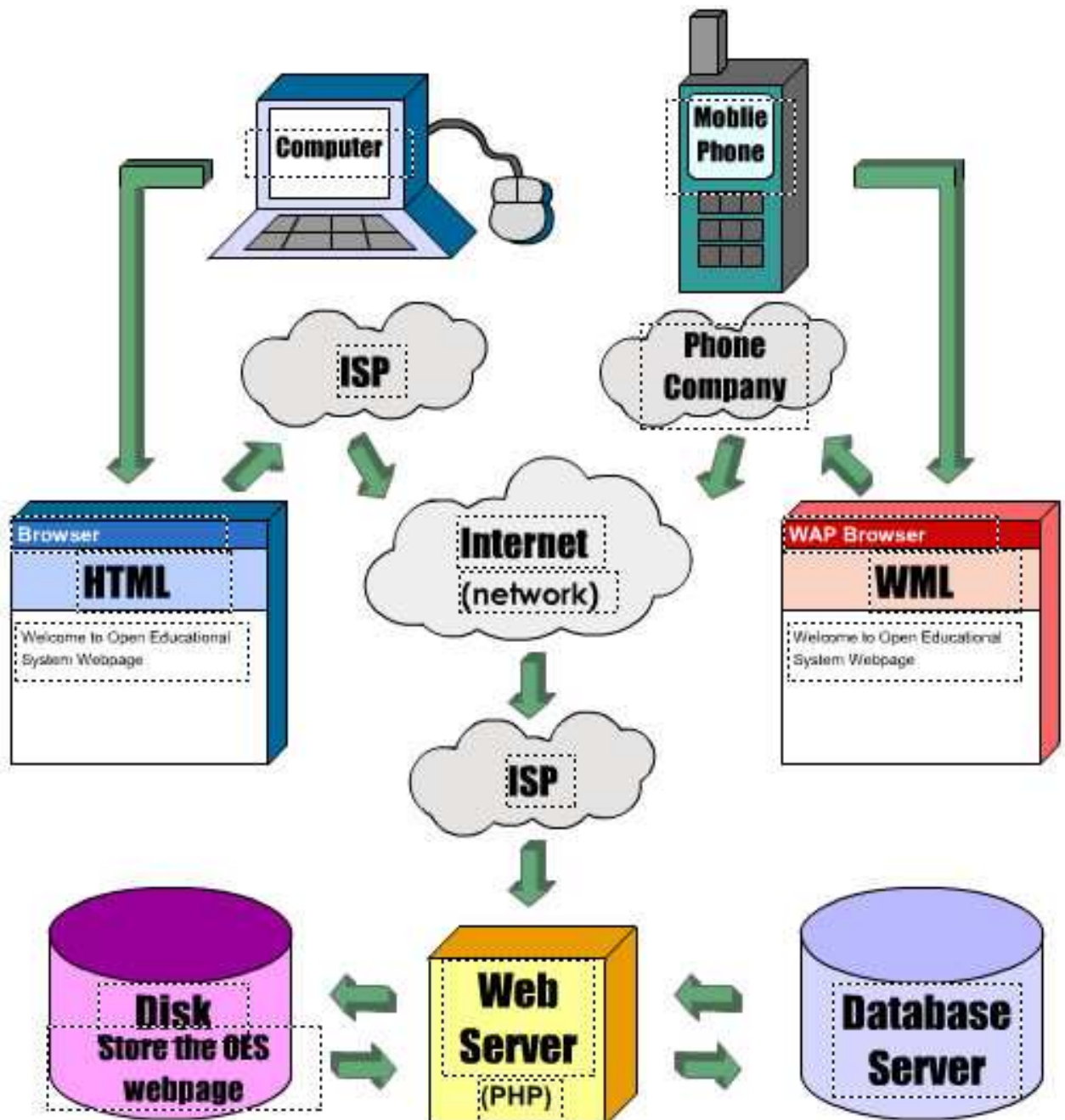


Fig 2.2 – Access to OES

The above picture (Fig 2.2), shows that how the OES work. Normally, OES allow the client work on two major platforms (client side). First one is computer user; it also included the Palm/Pocket PC users. They can access to the system via the standard HTML (Hyper Text Make-up Language) protocol. Another is mobile user, they can access to the system via the standard WAP protocol (Wireless Markup Language -- WML).

Although the current version of OES only allow the HTTP client, the WAP version will be announce on very soon. Because the system is combine with many modules, so only upgrade few module but allow WAP client to access with OES.

The next picture (Fig 2.3) shows more detail on how OES work on HTML compatible browser. In the picture, the "Result generator" generates the final HTML result. To allow the WAP user access to OES, this "Result generator" must be upgrade.

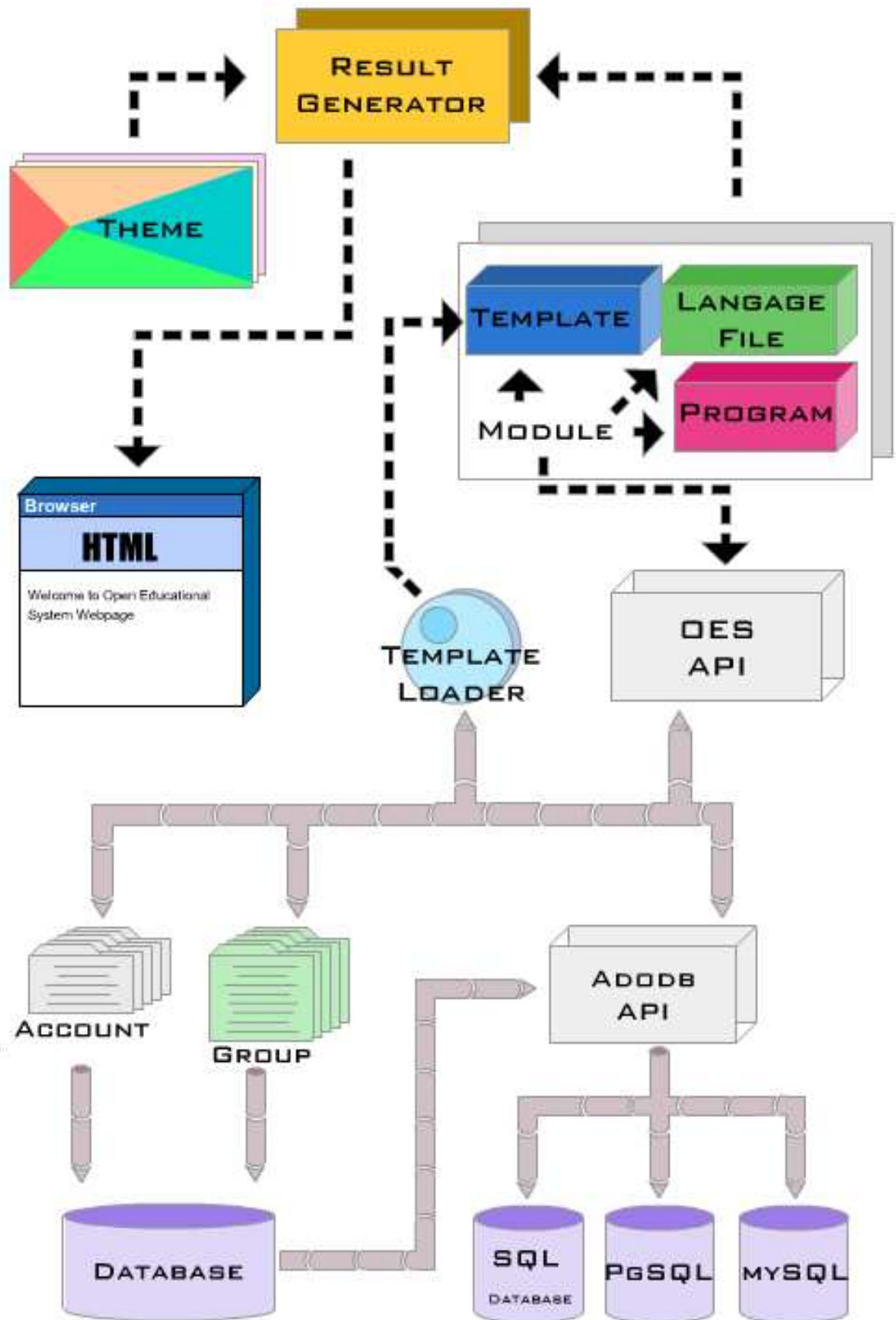


Fig 2.3 – Operation of OES

The information is store in the database; in real world, there are many type database systems in the market. But the problem is each is not 100% compatible with each other. So found a way to access those type of database but do not require modifier our code is very important. A predefined and powerful library set -- ADODB is used. ADODB, which is a set of database driver for PHP, the requirement understands the usage of this library, but do not need to consider the difference between each database system.

Although the modules programmer can access the database via ADODB directly, it is not recommended. The programmer must use the predefined database routine (API) to access the database. Because the usage of ADODB is not easy and the system may upgrade to use another set of database library instead with ADODB if a better library is founded.

From the above picture, the modules programmer can also use the predefined API to access the user account (include both administrator and normal user account), group account, and template loader.

Power by the design of template loader, the programmer

does not concern the final result display on the screen. They only need to design the layout for their own application (modules), and provide the language file set only. If they want the people can use the modules more easily, i.e. internationalization display is required. They can convert the language file into different language (e.g. English -> Traditional Chinese). The system has an ability to check how many languages on each module were installed on the system, and select the appropriate one display on the screen.

All the template design, language file and the application program combined together are call Modules. For each set of modules must contain the above three thing. The programmer also can consider this modules will place on which category. For the administrator control page, there are five categories on it.

<u>Category</u>	<u>Description</u>
OES	– The default setting for the system.
User account	– Setting for all account
System	– The setting for user side.
Modules	– Indicate which modules install on the system for the user.
Support	– Provide the help for the user.

On the user side, the system is not complete in this time. So this part will not discuss in this chapter. To understand how the user part design and working, please check the chapter 5 – Further Improvement. For instance, the user side is base on the technology used in the administrator control page.

2.2 System Design

This part is the important part of the project. Because this part will discuss the technology used, and how the kernel working in the project.

2.2.1 Modules Loader

OES is an organized design system, the entire thing are design into a set of modules. All the files are in fact managed from a few others that are located in the system folder and the “include” folder (library). Because of the security, we are not allowed to access the modules directory. To access the modules must call by the modules loader. But the modules may require some real-time response from the user, so it will collect those parameters from the browser and passing through to the demanded module. The standard form of the URL to access the modules is show on below:

[http://\[Host\].\[Domain\]/\[AdminPath\]/modules.php?category=\[Category\]&mod=\[ModulesName\]\[Parameters\]](http://[Host].[Domain]/[AdminPath]/modules.php?category=[Category]&mod=[ModulesName][Parameters])

The following table shows that the variable on the above URL link.

Variable	Descriptions
[Host]	Host name of the server. (e.g. www , admin ...etc)
[Domain]	The domain name. If the server does not have the domain name, it can replace with an IP address, but remember that no Host name when using IP address as the domain.
{ AdminPath }	This is a virtual directory map to the administrator page (optional). If the server does not have a sub domain name, or want to use a virtual director to access the administrator page, set a virtual director in the WEB server. (e.g. /admin/)
[Category]	This is the category name. For the administrator page, there were five choices: “oes”, “ <i>user_account</i> ”, “ <i>system</i> ”, “ <i>modules</i> ” and “ <i>support</i> ”
[ModulesName]	This is the name of modules. Actually is the folder name containing the module.
[Parameters]	This is an extra information will pass to the modules. Since the modules can't collect the user event.

Table 2.1 – Structure of URL link

For example, to access a module name ‘myModule’ and it is store on category ‘modules’ also it require two variables passing to the modules ‘var1’ and ‘var2’. For those variables with contain ‘This is var1’ and ‘Hello var2’. The URL shows on below (Fig 2.4) :

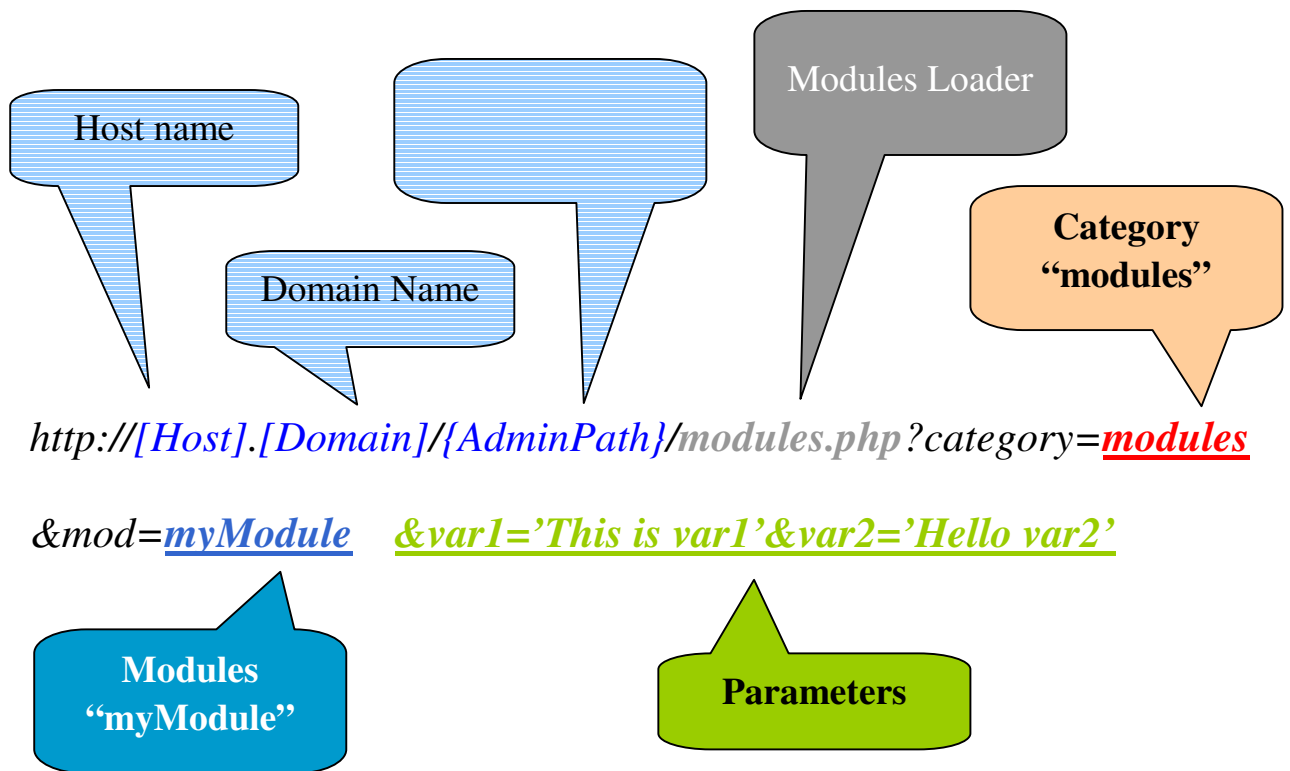


Fig 2.4 – Example of URL link

The above example shows that any variable can be assign dynamically and the modules loader will pass through those variable to the module. By using this technique, the user can assign different parameters and getting different result from the same modules. The modules programmer can collect the variable using the traditional way. The variable can be read by PHP by adding a '\$' before the variable name. The following is a PHP script shows how to get the variables.

```
<?php           // Modules example
echo $var1;     // result 'This is var1'
echo $var2;     // result 'This is var2'
?>
```

Another example, a variable call “**action**”. It is used to separate the different action operate by the modules.

*http://{Host}.{YourDomain}/{AdminPath}/modules.php?category=modules
&mod=myModule&**action=show***

The modules programmer can be use a “SWITCH...CASE” to separate the “**action**”. Following is an example.

```
<?php
// File : myModule.php

SWITCH ( $action )
{
    CASE 'show'      : . . . . .; break;
    CASE '.....'    : . . . . .; break;
    Default           : . . . . .; break;
}
?>
```

2.2.2 File Loader

The above examples shows how the modules working on OES. It was mentions before; OES don't want the client access the system directly, include loading the picture or download a file. So a modules loader was derivative. But how about the binary file, e.g. picture, flash movie, zip archive file? Fortunately, OES has a mechanism that can satisfy the above things. The method is call 'File loader'. Which is a program written by PHP. The purpose of this loader is redirecting the browser to download the file using the technology call 'Pass through'. But the requirement is, the file must contain within the OES. This is very important to improve the system security. Because the hacker want to download the system password through 'File loader'. To prevent this happen, the file loader was limit the client only allow download the file within the OES. Here is an example of using 'File loader' (Fig 2.5) :

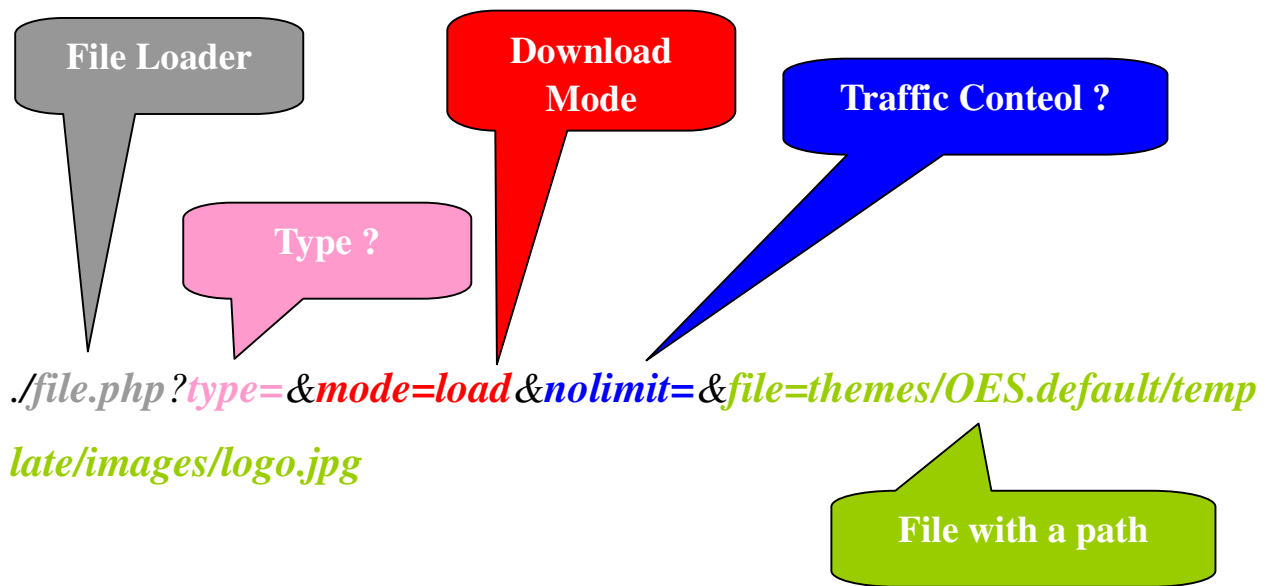


Fig 2.5 – File Loader

The parameter is list on below table (Table 2.2). For more details on the usage of this file loader, please read chapter 2.3

Parameter	Descriptions
<code>type</code>	For version 0.1. If 'type=JAVASCRIPT', it will move the base directory to '/include/javascript/'. Otherwise, it allow download the file within the OES.
<code>mode</code>	Allow 'load' or 'Get/Download'. 'Load' only load the result in the browser. 'Get/Download' will force the browser to download the file.
<code>nolimit</code>	This is a flag to control the download speed is it limited. 'nolimit=true' or 'nolimit=false'. By default is set to 'FALSE' (i.e. no traffic control).
<code>file</code>	Indicate the relative path of the file.

Table 2.2 – Structure of file loader

Basically, the file loader is using the CLASS call 'file' to complete this task. This CLASS contains many methods to access the file. The download speed limit and the Load and GET method is provide by this CLASS.

First of all, the LOAD/GET is a method to transmit the file to the client. But there are two different modes to distribute the contents to client. For Load method, it only transmits the file to the client browser and displays it. This is useful to display the picture, flash movie in the browser. Another is GET method, which is totally different to LOAD. This method will force the client browser to download the file without display on the screen. This is useful to allow the client download the file to their computer. For example, allow the client to download the notes.

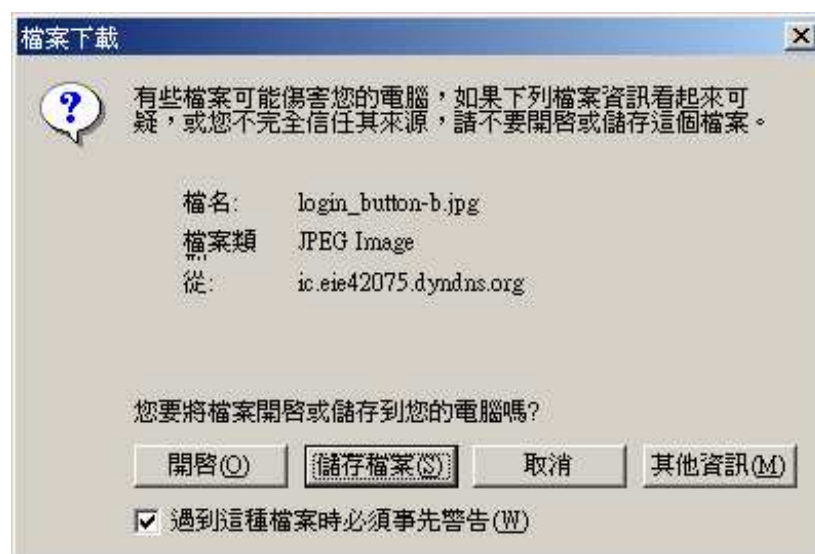


Fig 2.6 – Download a file (GET mode)

To LOAD the file to the client browser is an easy way. The major technique is to disable the CACHE on client browser when needed. Because sometimes the file was updated, but the client browser will load it from their cache, that mean the file is out dated. To prevent this happen, a technique is used to disable the CACHE. Here is the code to disable the cache (Fig 2.7).

```
header("Cache-Control: no-store, no-cache, must-revalidate"); // HTTP 1.1  
header("Cache-Control: post-check=0, pre-check=0", false);  
header("Pragma: no-cache"); // HTTP 1.0
```

Fig 2.7 – Disable Cache

The above code shows that the '*no-cache*' command is send by a PHP predefined function 'header'. 'header' is used to send the command to the client browser before sending any content to it (e.g. HTML code). It will tell the browser does not load the coming contents from cache. And the above command must be sent before deliver the content to the browser.

Another command must send to the browser before the contents is, the '*Content-Type*', '*Content-Disposition*', '*Content-Length*' and the '*Last-Modified*'. That information indicated that the coming contents (file) what is the file type; deliver mode; file name; file size and the last modified date.

The '*Content-Type*' indicates the MIME type of the file. It is very important to tell the browser what are the contents coming. For examples, sending a **JPG** picture to browser must use the MIME type '**image/jpeg**'; sending a **Flash movie** must use the MIME type '**application/x-shockwave-flash**'; sending a **JavaScript** must use '**application/x-javascript**'. The list of MIME type can be found in Appendix A.

The '*Content-Disposition*' can be separate into two parts. The '*filename*' is used to indicate what is the file name of the coming content. The important part is the '*Content-Disposition*'. Because it will affect the file deliver mode. For 'inline' mode, the content is display in the browser only; but '**attachment**' mode is not. In 'attachment' mode, it will force the browser to download the coming content according to the MIME type. Because some browser (e.g. Internet Explorer IE) will not force download the file when the MIME type is not '**application/octet-stream**', although it receive '**attachment**' in the '*Content-Disposition*'. But some other browsers see another header information '*Content-transfer-encoding*' (e.g. Mozilla). They only pop-up a save-as menu when it receives '**binary**' in this encoding type. So to allow the client download the file and pop-up a save-as menu to let them select the location store the file in local disk. Here is the

code to display the file content in browser (Fig 2.8); force the browser downloading the file (Fig 2.9).

```
// Sending the header information
if ( $this->noCache ) header("Cache-control: private");
header('Content-Type: '.getMIME($filename) );
header("Content-Disposition: inline; filename=$filename");
header("Content-Length: ".filesize( $this->filename ));
header("Last-Modified: " . gmdate("D, d M Y H:i:s") . " GMT");
if ( $this->noCache ) header("Cache-Control: no-store, no-cache, must-revalidate");
if ( $this->noCache ) header("Cache-Control: post-check=0, pre-check=0", false);
if ( $this->noCache ) header("Pragma: no-cache");
```

Fig 2.8 – Display the file content in browser

```
// Sending the header information
if ( $this->noCache ) header("Cache-control: private");
if ( $this->noCache ) header("Cache-Control: no-store, no-cache, must-revalidate");
if ( $this->noCache ) header("Cache-Control: post-check=0, pre-check=0", false);
if ( $this->noCache ) header("Pragma: no-cache");
header("Content-type: application/octet-stream");
header("Content-Disposition: attachment; filename=$filename");
header("Content-transfer-encoding: binary");
header("Content-Length: ".filesize( $this->filename ));
```

Fig 2.9 – Force browser to download file

The '*Content-Length*' and '*Last-Modified*' is used to tell the browser the file size and the last modifier date respectively.

The download speed limit (traffic control) is a technology to reduce the bandwidth usage. In traditional, the traffic control is the function only provide by the networking equipment. Usually it is expensive and not flexible to use. But using a simple technique, OES can simulate this function without expensive equipment. The following picture shows the download before (Fig 2.10) and after (Fig 2.11) using the traffic control technology.

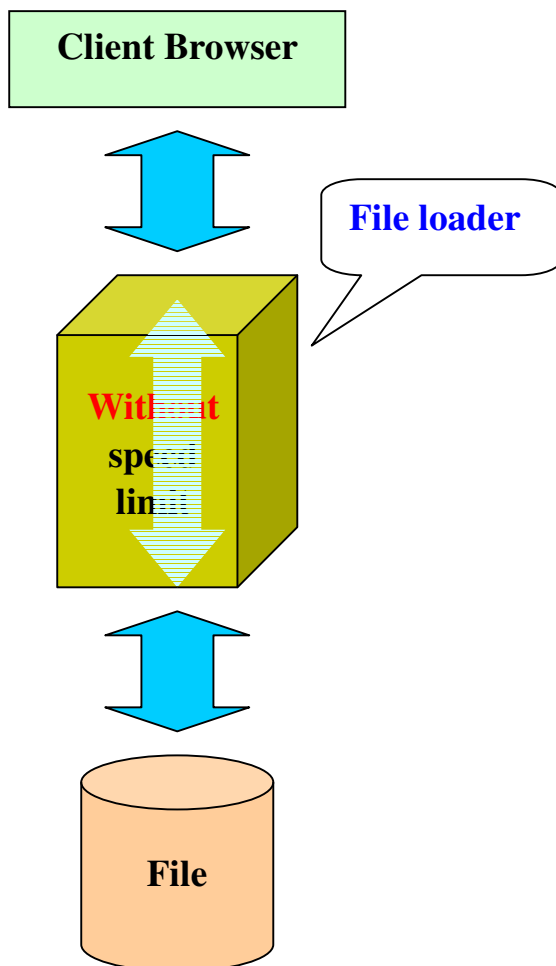


Fig 2.10 – Without Traffic

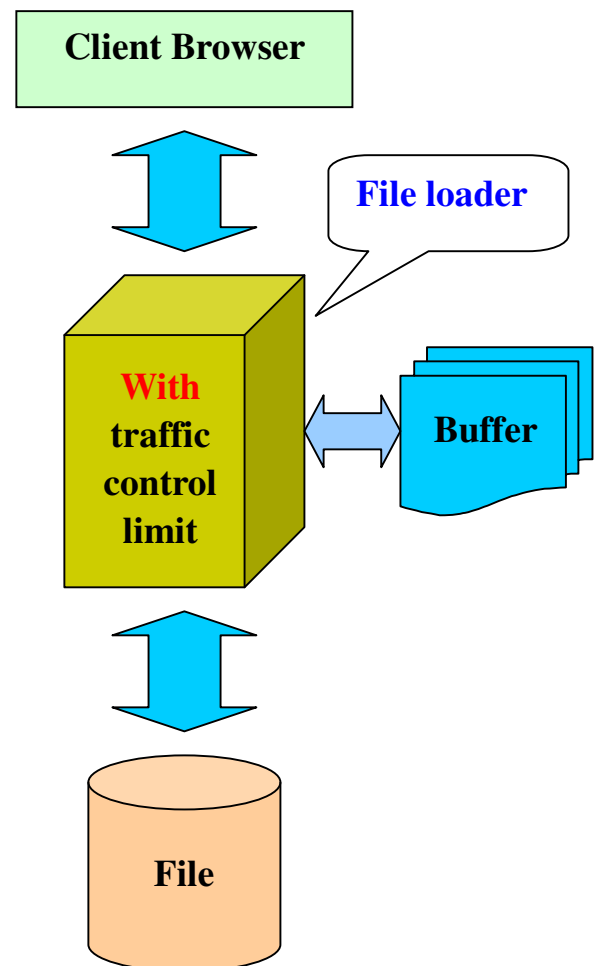


Fig 2.11 – With Traffic

The above picture shows the traffic control was added, but the operation on client side is the same. That mean the client do not know the speed was limited when he/she download the file from OES. It is a transparency work running on server side. The working principle of speed limit is, the server read a part of file into memory (buffer) and separates it into some small block. And then limit the client to download that small block in a time.

For example, to limit the client download a file not over 100 KB per second. The server will read 200 blocks and then transmit it in a second. To control the speed, the server can control the size read on each blocks. Following is the flow chart (Fig 2.12) shows how speed limit works and the formula to calculate the speed limit is:

$$\text{Speed} = \text{No. of Block read per second} \times \text{Size per read}$$

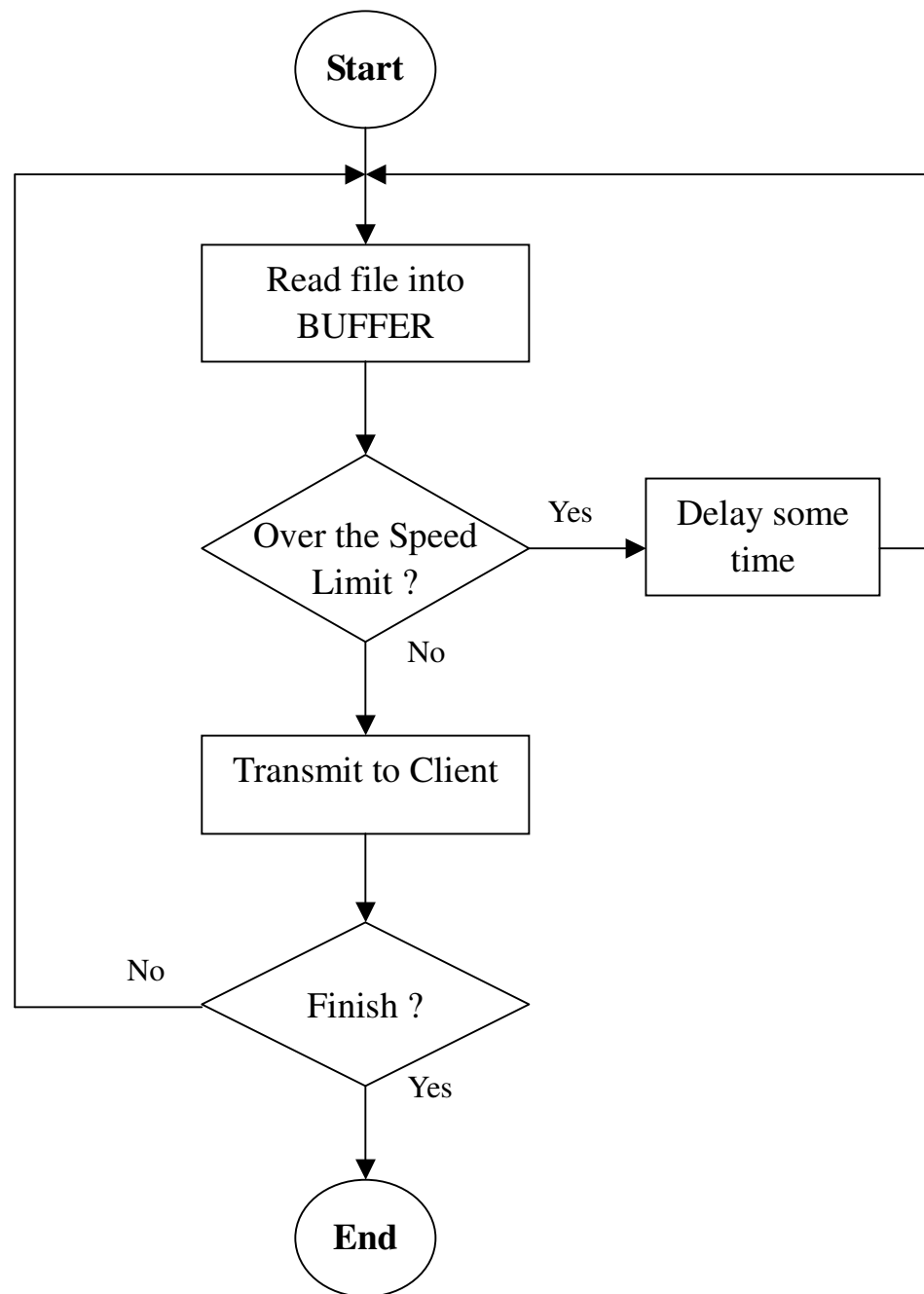
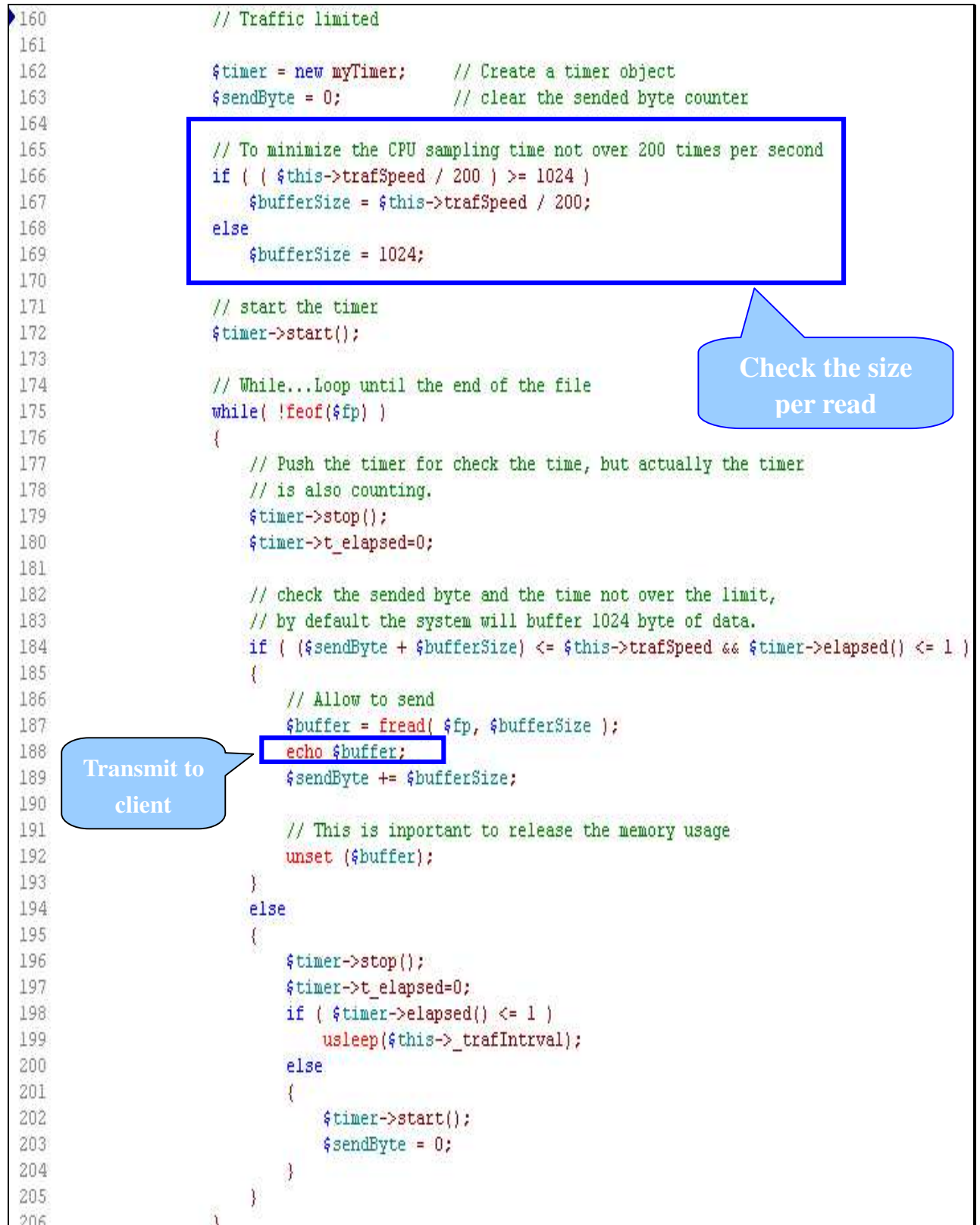


Fig 2.12 – Flow chart of Speed Limit

Part of program code shows how traffic control work (Fig 2.13) – '*lib-file.inc.php*' :



The image shows a snippet of PHP code from a file named `lib-file.inc.php`, specifically lines 160 through 206. The code implements a traffic limit by creating a timer and checking the elapsed time and buffer size. Two callouts highlight specific parts of the code:

- A blue box around lines 165-169 highlights the logic for determining the buffer size based on the current traffic speed.
- A blue callout bubble points to the `echo $buffer;` line (line 188), indicating that the data is transmitted to the client.
- A blue callout bubble points to the `if (($sendByte + $bufferSize) <= $this->trafSpeed && $timer->elapsed() <= 1)` condition (line 184), indicating that the size is checked per read.

```

160          // Traffic limited
161
162          $timer = new myTimer;    // Create a timer object
163          $sendByte = 0;          // clear the send byte counter
164
165          // To minimize the CPU sampling time not over 200 times per second
166          if ( ( $this->trafSpeed / 200 ) >= 1024 )
167              $bufferSize = $this->trafSpeed / 200;
168          else
169              $bufferSize = 1024;
170
171          // start the timer
172          $timer->start();
173
174          // While...Loop until the end of the file
175          while( !feof($fp) )
176          {
177              // Push the timer for check the time, but actually the timer
178              // is also counting.
179              $timer->stop();
180              $timer->t_elapsed=0;
181
182              // check the send byte and the time not over the limit,
183              // by default the system will buffer 1024 byte of data.
184              if ( ($sendByte + $bufferSize) <= $this->trafSpeed && $timer->elapsed() <= 1 )
185              {
186                  // Allow to send
187                  $buffer = fread( $fp, $bufferSize );
188                  echo $buffer;
189                  $sendByte += $bufferSize;
190
191                  // This is important to release the memory usage
192                  unset ( $buffer );
193              }
194              else
195              {
196                  $timer->stop();
197                  $timer->t_elapsed=0;
198                  if ( $timer->elapsed() <= 1 )
199                      usleep($this->_trafInterval);
200                  else
201                  {
202                      $timer->start();
203                      $sendByte = 0;
204                  }
205              }
206          }

```

Fig 2.13 – Code of Traffic Limit

2.2.3 Directory Structure

Basically, OES consist six directories, 'admin', 'conf', 'documents', 'include', 'session' and 'users'. The function of each directory is shows on below table (Table 2.3).

Directory	Descriptions
admin	This directory contains the administrator page. It includes the modules, template, theme and the language file for the administrator page only.
conf	This directory the configuration for the system. If this folder missing the configuration, the system will redirect to a setup page and generate the configuration automatically.
documents	This directory contains all the documents to help the user understand the usage of the system.
include	This directory store all the CLASS, FUNCTION, APIs provide by OES. The programmer must refer the API list on Appendix A.
session	Session file is store in this directory. The session is used to keep the user data instead of using Cookies.
users	The user side page.

Table 2.3 – Directory Structure of OES

Directory – Administrator page : (/admin)

It contains two subdirectories; also the file loader ([/admin/file.php](#)) and modules loader ([/admin/modules.php](#)) can be found in this directory. The subdirectories are 'modules' and 'theme'. For the subdirectory 'modules' also contain five subdirectories, which are the name of the category (chapter 2-1 and Fig 2.6). The modules are store on different category. Each set of modules also contain it own program, template and the language file. More details about modules will discuss on next chapter – chapter 2.4.

Another subdirectory 'theme' contains all themes for the administrator page. Each set of theme will packet into a directory and also it contain the language file for the theme.

Directory – Configuration File : (/conf)

This directory only contains one file. The file is call '[config.inc.php](#)', which is the main configuration for the system. Basically, this file is generated by the system setup wizard. For a new setup OES, this directory is missing this configuration file. The system will detect the configuration is it appear. If not, a setup wizard will appear and ask the question to enable the administrator configuration the system in first time. The picture of wizard is shows on below (Fig 2.14). For more details on how to setup the OES, please read the chapter 4.

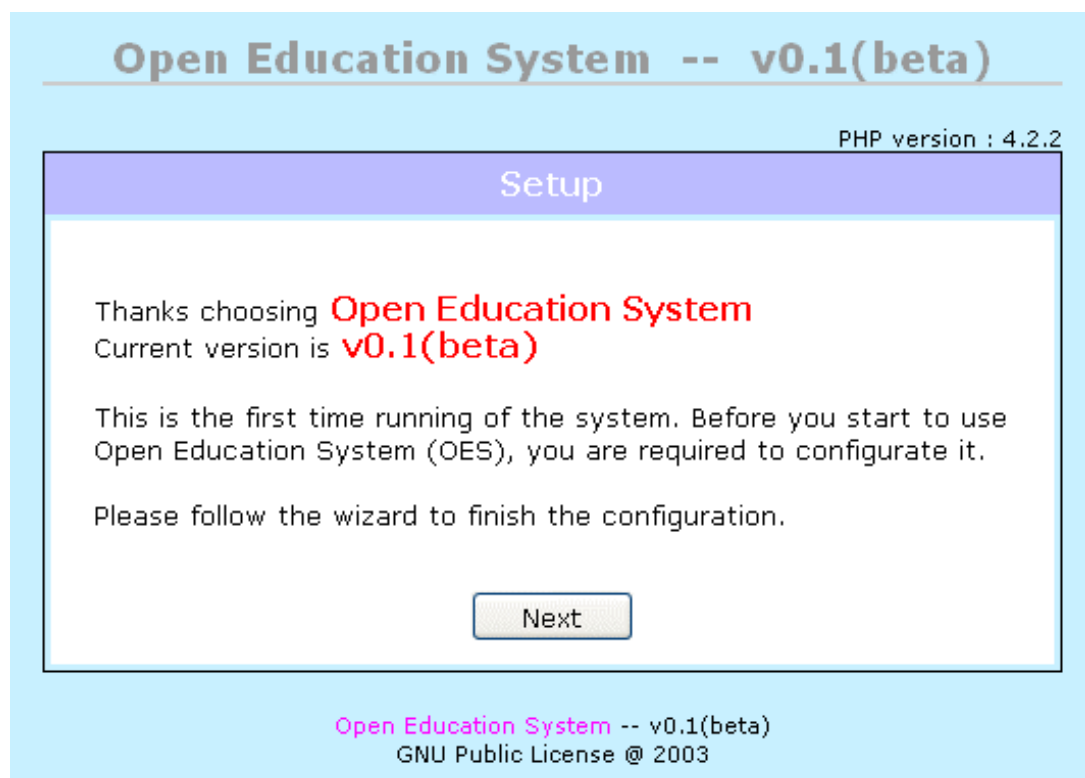


Fig 2.14 – Setup Wizard

How to check whether the configuration is it appear? Here is the example code shows that how the system to check the configuration (Fig 2.15).

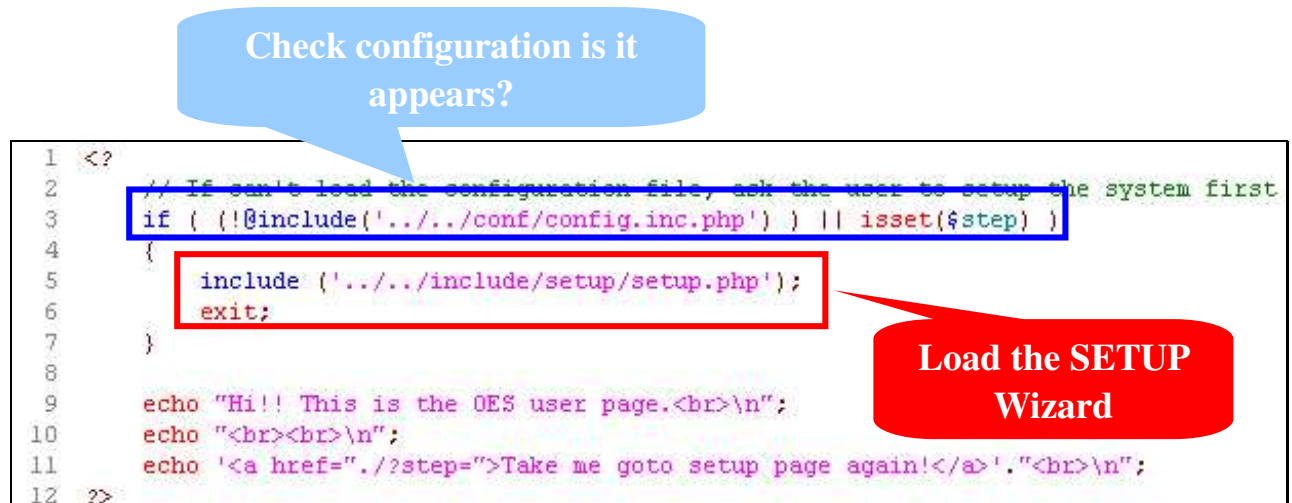


Fig 2.15 – Code to check Configuration file

Directory – Documentation : (/documents)

The purpose of this directory is storing all documentation files. The documentations may be in TEXT, HTML or MS-WORD format. Up to now, this folder will only story this report on it. To getting help with the installation or FAQ of OES, please go to OES official web site download the updated documentations file (<http://openeducation.sourceforge.net/>).

Directory – OES's API / Library : (/include)

In this directory, all the CLASS / Program Library (Application Programming Interface 'API') will be place on here. This is a part of the kernel of OES. Since the kernel are build on a set of library, and the entire library are place in this directory. The following table shows the entire library in the 'include' directory (Fig 21). To use the following library, only include it in a PHP script, and then refer to the API lists on Appendix (A).

File Name	Description
lib-account.inc.php	OES Library – User Account (Admin & User)
lib-file.inc.php	OES Library – File access
lib-group.inc.php	OES Library – Group account
lib-log.inc.php	OES Library – Log to database
lib-mime.inc.php	OES Library – MIME type checking
lib-modules.inc.php	OES Library – Modules
lib-mydb.inc.php	OES Library – Database access
library.inc.php	OES Library – General Routine
lib-template-mod.inc.php	OES Library – Template generator
lib-themes.inc.php	OES Library – Theme generator
lib-time.inc.php	OES Library – Timmer
javascript/script.js	OES Javascript – Mouse OnOver
adodb/adodb.inc.php	ADODB Library – General
adodb/*	Other ADODB Library (http://php.weblogs.com/)
setup/setup.php	OES Setup Wizard
setup/config.inc	OES default configuration file template
setup/lang/*	OES Setup Wizard Language set

Table 2.4 – List of library file

Directory – Session File : (/session)

This directory is used to store the session file generated by the system automatically. The purpose of session file is replacing the cookies store on client's computer. The log in password and all the user data will be store in the session temporary. And the file name is a random number, so it is a good method to store the sensitive data created by the client. The hacker is difficult to get the client information, because the data is store on server side. To increase the security, a good management of server is enough.

By the way, this directory also includes a subdirectory call 'admin'. The subdirectory also store the session file created by the system. The different is, the session file store in the subdirectory generated by the administrator.

Directory – User Page : (/user)

This directory is store the user page. This part is not finish at this time. But the architecture will simulate to the administrator page. The development on this part will be done on future.

2.3 Programming technique on Kernel

This chapter will discuss on, how the kernel was developed and what techniques are used.

Kernel is meaning that the core of the system. This is an important part of the system. Without the kernel, the system is a set of program, not organized and not functional. Basically, the kernel is a program that controls the system's behavior. For example, it receives the command by the client and then passes to the module; using the user preferred layout or an appropriate language to display on screen.

For the kernel of OES, that can be separate into few parts. There are the Themes Loader, Login Processor, Modules Loader and Template Loader. Before talk about these, there is another important part must introduce before. This is the Database structure, because most of information is store on the database system. So it must be introduce first.

Power by the third party database access library - ADODB. OES is able to access most database system. But in this project, only MySQL and PostgreSQL are considered. OES require at least

three TABLE to store the administrator account, group account and the LOG. The following picture shows the structure of those TABLE for MySQL (Fig 2.16-2.18).

```

/*
*****
--= TABLE of 'adm_users' =-
1. ac_enable      : 'Y' or 'N' indicate the admin account status
2. parent         : indicate the account is create by which admin. Default
                   is '0' mean this a/c create by ROOT admin
3. adm_group      : mean the administrator is under which group, it store
                   the group ID (e.g. '0;1;') [allow mutiple group]
4. expire_date_time : The default expire date is '9999-12-31 23:59:59',
                   mean Nerver Expire.
5. modules_allow   : mean the administrator allow to use which modules, it
                   store the modules ID. (e.g. '1;2;3;4;...')
*****
*/
$DBTABLE_NAME[0] = 'adm_users';
$CREATETABLE[0] = "CREATE TABLE $DBTABLE_NAME[0] ( ";
$CREATETABLE[0] .= "      id                INT UNSIGNED NOT NULL AUTO_INCREMENT, PRIMARY KEY (id), ";
$CREATETABLE[0] .= "      ac_enable      CHAR(1)      NOT NULL DEFAULT 'Y', ";
$CREATETABLE[0] .= "      parent         INT UNSIGNED NOT NULL DEFAULT '0', ";
$CREATETABLE[0] .= "      adm_login_id   CHAR(20)     NOT NULL, KEY (adm_login_id), ";
$CREATETABLE[0] .= "      adm_login_passwd VARCHAR(255) NOT NULL, ";
$CREATETABLE[0] .= "      adm_name       VARCHAR(100), ";
$CREATETABLE[0] .= "      adm_describe    VARCHAR(255), ";
$CREATETABLE[0] .= "      adm_email       VARCHAR(255), ";
$CREATETABLE[0] .= "      adm_group       VARCHAR(255), ";
$CREATETABLE[0] .= "      default_lang    CHAR(20)     DEFAULT 'EN', ";
$CREATETABLE[0] .= "      default_themes  VARCHAR(255), ";
$CREATETABLE[0] .= "      create_date     DATETIME     NOT NULL DEFAULT '9999-12-31 23:59:59', ";
$CREATETABLE[0] .= "      expire_date_time DATETIME     NOT NULL DEFAULT '9999-12-31 23:59:59', ";
$CREATETABLE[0] .= "      ip_access_list  VARCHAR(255), ";
$CREATETABLE[0] .= "      ip_deny_list    VARCHAR(255), ";
$CREATETABLE[0] .= "      modules_allow   TEXT ";
$CREATETABLE[0] .= ");";

```

Fig 2.16 – DB table : Administrator Account (MySQL)

```

/*
*****
--= TABLE of 'group' =-
This table only store the admin group information. It is used to separate
the administrator with different permission setting.
*****
*/
$DBTABLE_NAME[1] = 'oes_group';
$CREATETABLE[1] = "CREATE TABLE $DBTABLE_NAME[1] ( ";
$CREATETABLE[1] .= "      id                INT UNSIGNED NOT NULL AUTO_INCREMENT, PRIMARY KEY (id), ";
$CREATETABLE[1] .= "      group_enable      CHAR(1)      NOT NULL DEFAULT 'Y', ";
$CREATETABLE[1] .= "      group_name        VARCHAR(255) NOT NULL, ";
$CREATETABLE[1] .= "      group_describe    VARCHAR(255) ";
$CREATETABLE[1] .= ");";

```



```

/*
*****
-- TABLE of 'log' ==
This table used to store all the system log, including user login,
logout, download, post message ...etc.

1. log_code      : the event code (see the event_map.txt)
*****
*/
$DBTABLE_NAME[2] = 'log';
$CREATETABLE[2] = "CREATE TABLE $DBTABLE_NAME[2] ( ";
$CREATETABLE[2] .= "      id                INT UNSIGNED NOT NULL AUTO_INCREMENT, PRIMARY KEY (id), ";
$CREATETABLE[2] .= "      log_code          INT(4) UNSIGNED NOT NULL DEFAULT '1', ";
$CREATETABLE[2] .= "      log_time          DATETIME      NOT NULL DEFAULT '9999-12-31 23:59:59', ";
$CREATETABLE[2] .= "      log_event         TEXT          NOT NULL, ";
$CREATETABLE[2] .= "      log_user_id       VARCHAR(255)  ";
$CREATETABLE[2] .= ");";

```

Fig 2.18 – DB table : Log

Unfortunately PostgreSQL is different from the MySQL, the SQL query is not same as MySQL. Therefore, PostgreSQL cannot execute the above query string. So OES must prepare another set of query for the PostgreSQL. The following shows the table 'log' using the PostgreSQL's format (Fig 2.19).

```

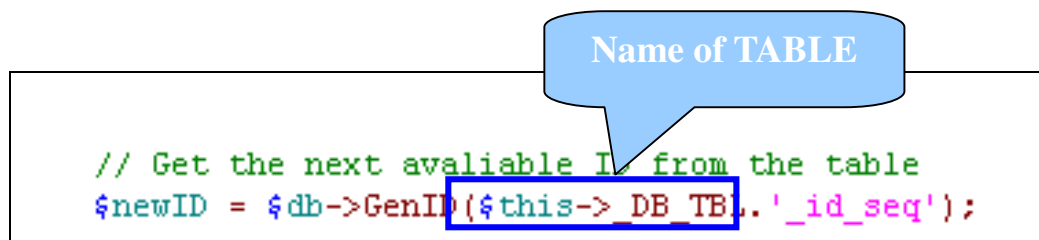
/*
*****
-- TABLE of 'log' ==
This table used to store all the system log, including user login,
logout, download, post message ...etc.

1. log_code      : the event code (see the event_map.txt)
*****
*/
$DBTABLE_NAME[2] = 'log';
$CREATETABLE[2] = "CREATE TABLE $DBTABLE_NAME[2] ( ";
$CREATETABLE[2] .= "      id                SERIAL,          PRIMARY KEY (id), ";
$CREATETABLE[2] .= "      log_code          INT          NOT NULL DEFAULT '1', ";
$CREATETABLE[2] .= "      log_time          DATETIME     NOT NULL DEFAULT '9999-12-31 23:59:59', ";
$CREATETABLE[2] .= "      log_event         TEXT         NOT NULL, ";
$CREATETABLE[2] .= "      log_user_id       VARCHAR(255) ";
$CREATETABLE[2] .= ");";

```

Fig 2.19 – DB table : LOG (PostgreSQL)

The above picture shows the PostgreSQL does not support 'Auto Incensement' to automatically increase the number. This number is very important, because it is the PRIMARY KEY of the table. Without this number, it is difficult to select the unique the ROW of record. PostgreSQL use another method to do the same thing, they call this 'SERIAL'. For the library ADODB, it uses a method to simulate this on MySQL. It create a table only have one ROW and one COLUMN. That mean only one Field can be use. The field only stores the number of total ROWs in the table. To insert a new ROW to the table, it must call the function provide by ADODB to generate the new ID. Here is the code to generate the new ID (Fig 2.20).



The diagram shows a code snippet within a rectangular box. A blue callout bubble with the text "Name of TABLE" points to the variable `$this->DB_TBL` in the code. The code is as follows:

```
// Get the next available ID from the table
$newID = $db->GenID($this->DB_TBL, '_id_seq');
```

Fig 2.20 – Generate a new ID

```
// Query an empty record set
// Assume the 'id' is the key of a table
// (by default $this->keyField='id')
$sql = "SELECT * FROM ".$this->DB_TBL." WHERE ".$this->keyField." = ".$inData[$this->keyField];
@ $rs = $db->Execute($sql);

// Get the insert query string
@ $query = $db->GetUpdateSQL($rs, $inData);
```

Fig 2.21 – Example on Update ROW on Table

```
// Get the next available ID and the current date and time
// (by default the key field $this->keyField='id')
    $inData[$this->keyField] = $db->GenID($this->_DB_TBL.'_id_seq');

// Query an empty record set
$sql = "SELECT * FROM ".$this->_DB_TBL." WHERE id = -1";
@ $rs = $db->Execute($sql);

// Get the insert query string
@ $query = $db->GetInsertSQL($rs, $inData, true);
```

Fig 2.22 – Example on Insert ROW into Table

```
// ----- //
// Delete an old record from table //
// ----- //
$query = "DELETE FROM ".$this->_DB_TBL." WHERE ".$this->keyField."='".$inData'";
```

Fig 2.23 – Example on Delete ROW from Table

2.3.1 Themes Loader

The Themes Loader will check what Themes are install on the system. It will use the default Theme as the default layout style display on the screen. The default setting is store on the configuration file where place on '/cong/conf.inc.php'. The following shows the example of configuration file (Fig 2.24 and Fig 2.25).


```

1  <?
2  // File      : /conf/config.inc.php
3  //
4  // Description : Configuration file.
5
6  /*
7   * Open Education System
8   *
9   * Latest version is available at http://openedusys.sourceforge.net
10  * This is the configuration file for OES (Open Education System).
11  */
12
13  // *** You should modifier this line for the first run.      ***
14  // *** This variables contains the path of the OES is installed. ***
15
16  // The system path of Open Education System? (FULL PATH)
17  // *** REMEMBER!! If OES is running on windows system, please make ***
18  // *** sure the system path is correct. ***
19  $CONF_SYSTEM_PATH      = '/home/system/var/www/hd2_project';
20
21  // The URL will OES be work on?
22  $CONF_SYSTEM_URL       = 'http://project.eie42075.dyndns.org';
23
24  // The administration URL
25  $CONF_ADMIN_URL        = 'http://project.eie42075.dyndns.org/admin';
26
27
28
29  // ***** Warning!! *****
30  // *** NEVER change this to TRUE unless root administrator is forget the password **
31  // *** ( the default ID and Password would be 'rootAdmin' , 'rootPassword' ) ***
32  // *****
33  $CONF_FORGET_ROOT_PWD   = FALSE;
34
35
36  ////////////////////////////////////////
37  //                               Warning!!                               //
38  //                               //                                       //
39  // Don't modifier the following part, unless you understand //
40  // what you are doing.                                     //
41  ////////////////////////////////////////
42  $CONF_SERVER_NAME       = 'Open Educational System';
43  $CONF_SERVER_VER        = 'v0.1(beta)';
44
45  // URL for OES Home
46  $CONF_OES_URL            = 'http://openededucation.sourceforge.net';
47  $CONF_SESSION_TIMEOUT   = 600; // Default timeout is 600 second
48
49
50  // Database setting
51  $CONF_DATABASE_TYPE     = 'mysql';
52  $CONF_DATABASE_ADDR     = '';
53  $CONF_DATABASE_PORT     = '';
54  $CONF_DATABASE_DBNAME   = 'oes';
55  $CONF_DATABASE_ID       = 'oes';
56  $CONF_DATABASE_PWD      = 'oes';
57
58
59  // Start Administrator Page Setting
60  $CONF_ADMIN_PATH        = $CONF_SYSTEM_PATH.'/admin';
61  $CONF_ADMIN_THEMES_PATH = $CONF_SYSTEM_PATH.'/admin/themes';
62  $CONF_ADMIN_MODULES_PATH = $CONF_SYSTEM_PATH.'/admin/modules';
63  $CONF_ADMIN_SESSION_PATH = $CONF_SYSTEM_PATH.'/session/admin';
64  $CONF_ADMIN_SESSION_TIMEOUT = 0; // Default timeout is NEVER
65
66
67  // The path of OES library
68  $CONF_CONFIG_PATH       = $CONF_SYSTEM_PATH.'/conf';
69  $CONF_INCLUDE_PATH      = $CONF_SYSTEM_PATH.'/include';
70  $CONF_THEMES_PATH       = $CONF_SYSTEM_PATH.'/users/themes';
71  $CONF_MODULES_PATH      = $CONF_SYSTEM_PATH.'/users/modules';
72  $CONF_SESSION_PATH      = $CONF_SYSTEM_PATH.'/session';
73
74
75  // Default Themes
76  $CONF_DEFAULT_ADMIN_THEMES = 'OES.default';
77  $CONF_DEFAULT_USER_THEMES  = 'OES.default';
78
79
80  // Define the default download speed
81  $CONF_DOWNLOAD_SPEED     = 0;
82  >>

```

Default Theme

Fig 2.24 – Example of Configuration file

```

75      // Default Themes
76      $CONF_DEFAULT_ADMIN_THEMES  = 'OES.default';
77      $CONF_DEFAULT_USER_THEMES   = 'OES.default';

```

Fig 2.25 – Default Theme

Following is a part of code shows how to change the theme in a PHP script (Fig 2.26), the code is from '/admin/index.php'.

The diagram illustrates the usage of a theme in a PHP script. It features a code block with several callouts explaining specific parts of the code:

- Change the Theme:** A blue callout pointing to the line `$theme = new myTheme($themeName, $SESS_DEFAULT_LANG);`.
- Loading a Template and Dynamic assign the variable:** An orange callout pointing to the block of code starting with `$theme->getTemplate('login');` and including several `$theme->assign()` calls.
- Display the result on Screen:** A pink callout pointing to the line `echo $theme->getResult();`.

```

340      // Loading the Themes
341      $theme = new myTheme( $themeName, $SESS_DEFAULT_LANG );
342      $theme->debug=0;
343      $theme->isAdmin=1;
344      $theme->getTemplate('login');
345      $theme->assign( 'PROGRAM_NAME', $CONF_SERVER_NAME );
346      $theme->assign( 'PROGRAM_VER', $CONF_SERVER_VER );
347      $theme->assign( 'OES_URL', $CONF_OES_URL );
348      $theme->assign( 'TXT_LICENSE', 'GNU Public License © 2003' );
349
350      // Display if the user type wrong password
351      if ( $WRONGPASSWORD )
352          $theme->assign( 'WRONG_PASSWORD', 'LANG=WRONGPASSWORD' );
353      else
354          $theme->assign( 'WRONG_PASSWORD', '' );
355
356      // The login box will larger when the browser support CSS
357      if ( cssSupport() )
358          $theme->assign( 'LOGIN_BOX_SIZE', '24' );
359      else
360          $theme->assign( 'LOGIN_BOX_SIZE', '18' );
361
362      $theme->getHeader();
363      echo $theme->getResult();

```

Fig 2.26 – Usage of Theme

This example shows to load the 'Theme', a Object call 'myTheme' must be create first. There are two parameters passing to the CLASS Constructor 'myTheme'. The first one is the name of

theme set to be use, second one is the language of the theme set will be display on the screen. After that, it must to get the template where are store on the theme set. In this example, it load the template 'login' where is place on '/admin/themes/[THEMES_NAME]/template/login'. This is the template for LOGIN page. But the template only contains the HTML layout, but no any TEXT on it. Because the TEXT is language dependence. So the system using another method to replace the TEXT, the data will be separate into two types 'Static' and 'Dynamic'. To understand the template design, please read the chapter 3.1. After load the template 'LOGIN' and assign the dynamic TEXT or Content to it, the result are display by the CLASS function 'getResult()'.

But, unfortunately although the TEXT are display on different language, the browser are not change the language automatically. The system send a Header information to the browser, tells them use appropriate language to display the contents, and the problem are solved. Here is the information send to the client browser (Fig 2.27).

```
// send the current charset to the client browser  
header ('Content-Type: text/html; charset='.$LANG_CHARSET);
```

Fig 2.27 – Automatic select language

The variable '**\$LANG_CHARSET**' is the system variable. It used to store the default character set will be use. To understand which character set is accepted, please visit the website : '<http://www.iana.org/assignments/character-sets>'.

Before to use the above code, load the library into the program is required. The library should be loaded is 'lib-themes.inc.php'. Actually all the library are store on directory '/include'. But it is not recommend load the library with a full path. The right method to include the library is shows on below (Fig 2.28).

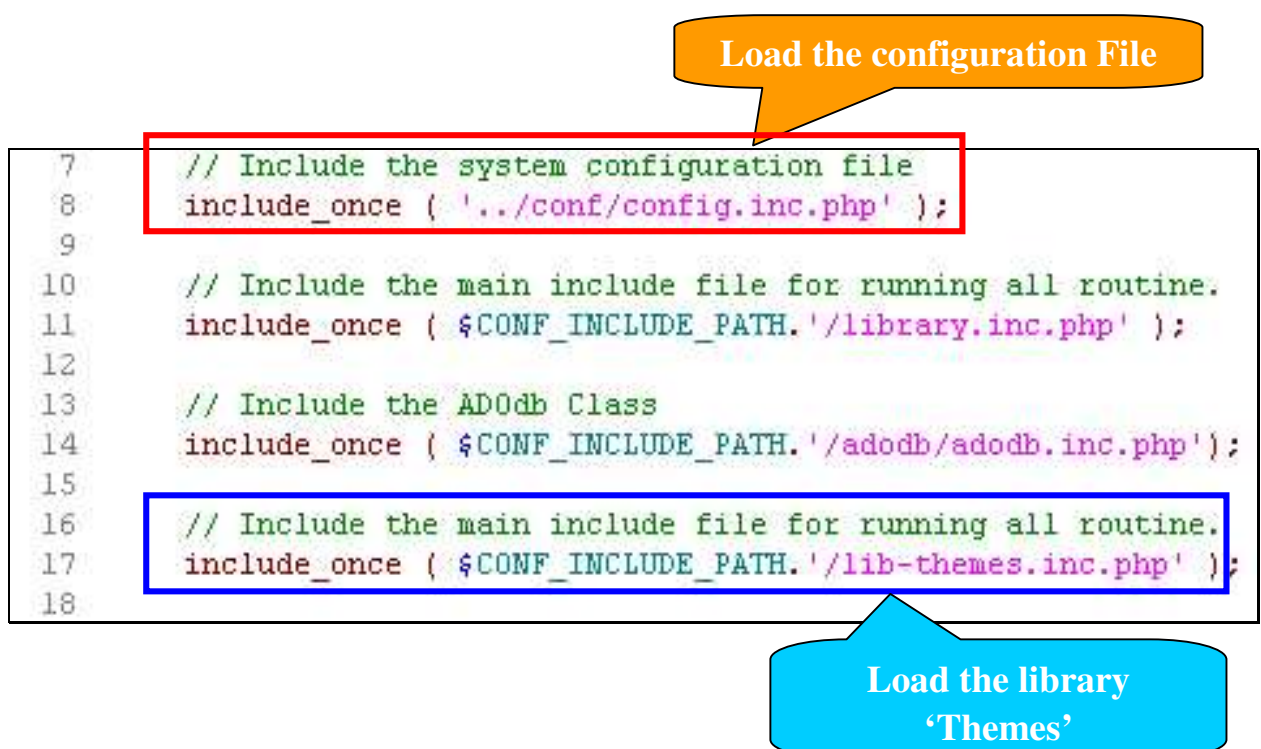


Fig 2.28 – Load the library

The above example to load the library 'lib-themes.inc.php', which is not include it in normal way, a variable was added before. Because only the configuration file knows the system path, so to include an additional library must include the configuration file first and then add a variable '`$CONF_INCLUDE_PATH`' to the include path of the library. Which the variable can indicate the absolute path of 'include' directory on the server.

2.3.2 Login Processor

The login processor shows the first page on the administrator control page. So it will check whether the administrator is login successful. If the login was fail or just start to display the login page. It will request the Theme Loader to display the login page using the default Theme layout. The following is the flow chart shows how the login processor works (Fig 2.29). When the login is success, the Login Processor will create a session and store the information to identifier who is logged to the system, and the session can be use within the system. That mean only login once time, the system can read the data store in the session. The session will be destroy when the user was log out or

the session time was expired. The flow chat of log out process shows on (Fig 2.30). To check the *LoginID* and *Password* is correct, the Login Processor will call the library '[lib-account.inc.php](#)' and use the provided function try to select one row of record that the login id and the password is the same as the user entered. If the record was found, that mean the login process will be success. But there also has a requirement, the record has a field call '[ac_enable](#)', which indicate the account is active. Although can select one record that is match the entered ID and password, if the '[ac_enable](#)' is '[N](#)' the login process will be fail. This field is good for temporary disable the account. Because sometimes, may want to disallow some person login to the system, but will be resume it very soon. This is a good facility provide to the administrator disable the account temporary.

After the successful login to the system, the Login Processor will give the hand to the Modules Loader. Which the Modules Loader will access the Themes Loader to display the layout with an approbate language and access to the module. By the way, the Language and Theme can be change by each administrator. When they do not select their preference, the system will use the default language and theme to display.

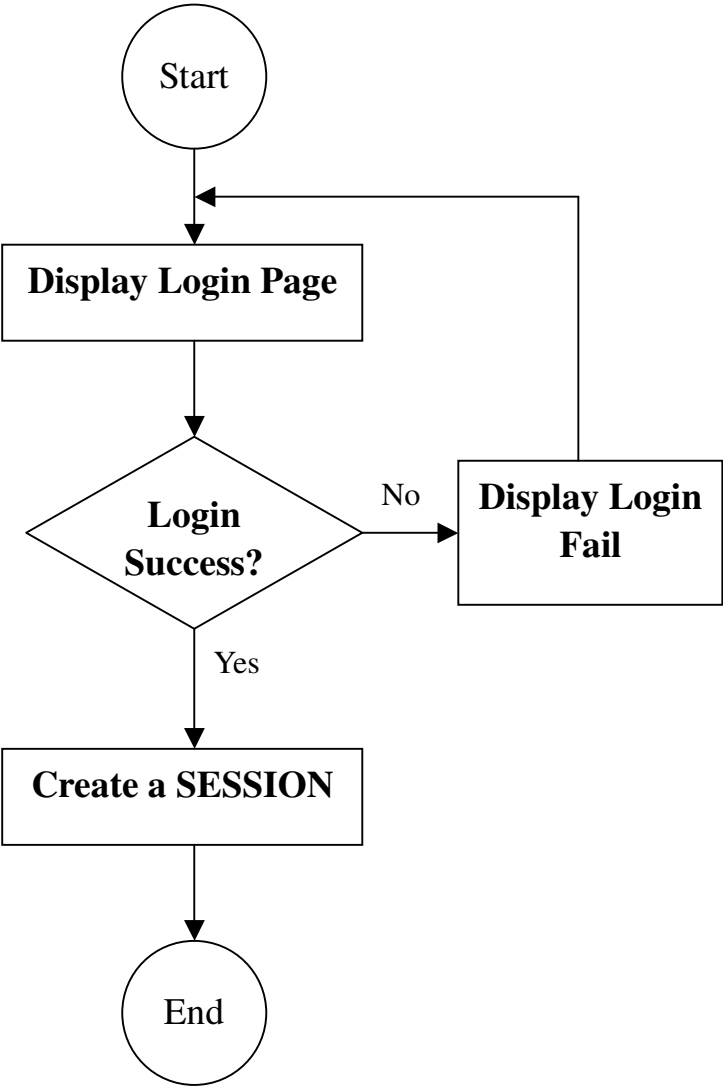


Fig 2.29 – Login flow chart

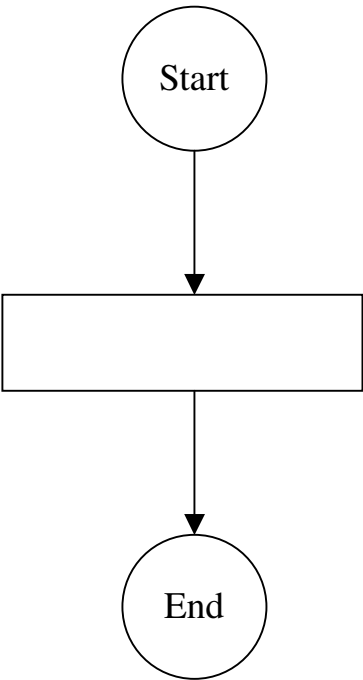


Fig 2.30 – Logout flow chart

```

21 // Save the session file into a specific folder
22 session_save_path ( $CONF_ADMIN_SESSION_PATH );
23
24 // Change the session name to SSID
25 session_name ( 'SSID' );
26
27 // Start session
28 session_start();

```

Fig 2.31 – Prepare the SESSION Environment

```

// The adminID and password verified, save the admin account
// information into session
/*
    SESS_ID
    SESS_AC_ENABLE
    SESS_PARENT
    SESS_ADM_LOGIN_ID
    SESS_ADM_LOGIN_PASSWD
    SESS_ADM_NAME
    SESS_ADM_DESCRIBE
    SESS_ADM_EMAIL
    SESS_ADM_GROUP
    SESS_DEFAULT_LANG
    SESS_DEFAULT_THEMES
    SESS_CREATE_DATE
    SESS_EXPIRE_DATE_TIME
    SESS_IP_ACCESS_LIST
    SESS_IP_DENY_LIST
    SESS_MODULES_ALLOW
*/
$array = $rs->fields;
$arrayFlip = array_keys($array);
foreach ( $arrayFlip as $arrayName )
{
    $sess_var_name = 'SESS_'.strtoupper( $arrayName );
    $$sess_var_name = noDoubleSpace( $array[$arrayName] );
    session_register ( $sess_var_name );
}

```

These are the SESSION variable for the administrator. These variables can be read within the system.

The code to save the administrator account information into SESSION

Fig 2.32 – Create SESSION (Login)

```
// Destroy all session
session_unregister ( "sess_admLoginID" );
session_unset();
session_destroy();
// Redirect to login page
Header ( 'Location: '.$CONF_ADMIN_URL.'/?' );

// Exit the logout process
exit;
```

Fig 2.33 – Destroy the SESSION (Logout)

More details about the LOGIN and LOGOUT process. Please refer the program code on Appendix (B) – '/admin/index.php'.

2.3.3 Modules Loader

Mode details on his part will be discussed on next chapter 2.4.

2.3.4 Template Loader

Basically the template loader is a CLASS of a library '[lib-template-mod.inc.php](#)' and it is a sub part of modules loader. So it will be discussed on next chapter 2.4.

2.3.5 *SESSION*

To start a session, it must do it before sending any contents to the client's browser (i.e. no echo before). Because the session is a type of control information, that must be send to the client's browser at the beginning. The following code shows how to start the session in PHP (Fig 2.34).

```
// Save the session file into a specific folder
session_save_path ( $CONF_ADMIN_SESSION_PATH );

// Change the session name to SSID
session_name ( 'SSID' );

// Start session
session_start();
```

Fig 2.34 – Start SESSION

The first two lines of code are optional. If not provide the session save path, it will use the OS default temporary file directory to store the session file. For the Linux system, it will be save into '/tmp'. The second line is providing the name of the session. Basically the name will be the prefix of the file name. By PHP default, the session name is 'PHPSESSID'. Only the third line must stay at the beginning of the PHP script. To pass the SESSION variable to another PHP script, it is only require to start the

session on each PHP script which require the variable store in the session. To read the session is same as read the content on any variables, but PHP do not require declare the variable before to use. Then, a function is used to check the session variable is it created. The function is '[session_is_registered\(\)](#)'.

To save the variable into session, there is not same as assign any variable on PHP. Firstly, create a variable and store the data that is wanted. Then use a function call '[session_register\(\)](#)' to register the variable in the session. Here is the example (Fig 2.35).

```
$var = "data";  
session_register("var");
```

Fig 2.35 – Register Session

Now the variable can be read and save into session. How to pass the session to another PHP script? It was mentioned before, start the session in another PHP script. Normally it is work. But the operation of session has a requirement. The requirement is found a way to pass the Session ID to another PHP script, then the script can access to the same session created before. One method is sending a cookies to the client's browser and tell them the session id, then the session id can be pass automatically. Another method is placing the session id at the end of the URL

link, but this does not require the client start cookies. The first method is recommended and is used on OES.

2.4 Programming technique on Modules

This chapter will discuss on, how the module system was developed and what techniques are used.

The modules must be load by the modules loader. Because the modules cannot be access directly, so a mechanism to load the modules is required. The 'Modules Loader' is used to perform this task. A brief explain of modules loader was introduce in pervious chapter. In this chapter, will discuss the structure of a module.

The module must be packet in a directory. The directory must contain few of things. First of all, a file call '[module.php](#)' must appear in the directory. The purpose of this file is used to define the behavior of the module and tell to the Modules Loader. The following is the example of a modules configuration file '[module.php](#)' (Fig 2.36). All the value are store in an array '[\\$MODULES\[\]](#)'. And the array is limited on some fields, the function of some fields are shows on the following table (Table 2.5).



```

1  <?
2
3
4  // This is the configuration file for one of the administrator module.
5  // The usage of this file is define the activity of this module.
6
7  $MODULES["ENABLE"] = TRUE;
8  $MODULES["NAME"] = 'OES Group'; // Short name for module
9  $MODULES["VERSION"] = 'v1.0'; // version number of module
10 $MODULES["DEPENDS"] = ''; // depend on which modules (in modules name)
11 $MODULES["DEFAULT_LANG"] = 'en'; // default language, if the system can't find
12 $MODULES["LANG_DIR"] = '/lang'; // language file dir
13 $MODULES["CATEGORY"] = 'user_account'; // category of this module in
14 $MODULES["ICON_MOUSEOUT"] = '/group-a.jpg'; // default icon
15 $MODULES["ICON_MOUSEOVER"] = '/group-b.jpg'; // default icon
16 $MODULES["EXECUTION_FILE"] = 'mod_group.php'; // the execution file of this module
17
18 $MODULES["TEMPLATE"] = 'mod_group_template'; // the default template for this module
19 $MODULES["TEMPLATE_NO_RECORD"] = 'mod_no_record_template';
20 $MODULES["TEMPLATE_EDIT"] = 'mod_edit_template';
21 $MODULES["TEMPLATE_CREATE"] = 'mod_create_template';
22 $MODULES["TEMPLATE_ERROR"] = 'mod_error_template';
23
24 >>

```

Fig 2.36 – Module Configuration file ‘module.php’

From the above code example, there are some variables must be defined. Here is the description of those variable (Table 2.5).

Variable	Description
<code>\$MODULES["ENABLE"]</code>	‘TRUE’ = enable the module. ‘False’ = disable
<code>\$MODULES["NAME"]</code>	The name of the module. (Type in English)
<code>\$MODULES["VERSION"]</code>	The version of the module. (Optional)
<code>\$MODULES["DEPENDS"]</code>	Name of other module, which the module is depend on that. (Optional)
<code>\$MODULES["DEFAULT_LANG"]</code>	The default language. Only be use when the system can’t found in the language directory. (Usually the default language is ‘en’ = English)
<code>\$MODULES["LANG_DIR"]</code>	The folder stores the language file.
<code>\$MODULES["CATEGORY"]</code>	The name of category, which the module in.
<code>\$MODULES["ICON_MOUSEOUT"]</code>	The ICON of the module.
<code>\$MODULES["ICON_MOUSEOVER"]</code>	The ICON of the module. (Mouse is MoveOver)
<code>\$MODULES["EXECUTION_FILE"]</code>	The main executions file that is the main program of the module.
<code>\$MODULES["TEMPLATE"]</code>	The main template file.

Table 2.5 – Define Module Configuration File

The above table was showing the required variable on the module configuration file. It also accepts an optional variable, which is depends on the programmer. In the above example, the module has five templates. The first (major) template must be defined in `$MODULES["TEMPLATE"]` which is 'mod_group_template', the other template (optional) can use another variable to store the template file in name.

```
18  $MODULES["TEMPLATE"]           = 'mod_group_template';
19  $MODULES["TEMPLATE_NO_RECORD"] = 'mod_no_record_template';
20  $MODULES["TEMPLATE_EDIT"]      = 'mod_edit_template';
21  $MODULES["TEMPLATE_CREATE"]    = 'mod_create_template';
22  $MODULES["TEMPLATE_ERROR"]     = 'mod_error_template';
```

Fig 2.37 – Optional variable in module configuration file

In the modules loader, it will check the module's configuration file when access to it. It only allows the module, which is 'ENABLE' to be executing and display on the screen. Then the Modules Loader will check the ICON defined in the module configuration file, and generate the link by File Loader to be display on the client browser. Finally it will check the name in the language directory and find an appropriate language to be display. The following picture shows the module is showing on the client browser (Fig 2.38).

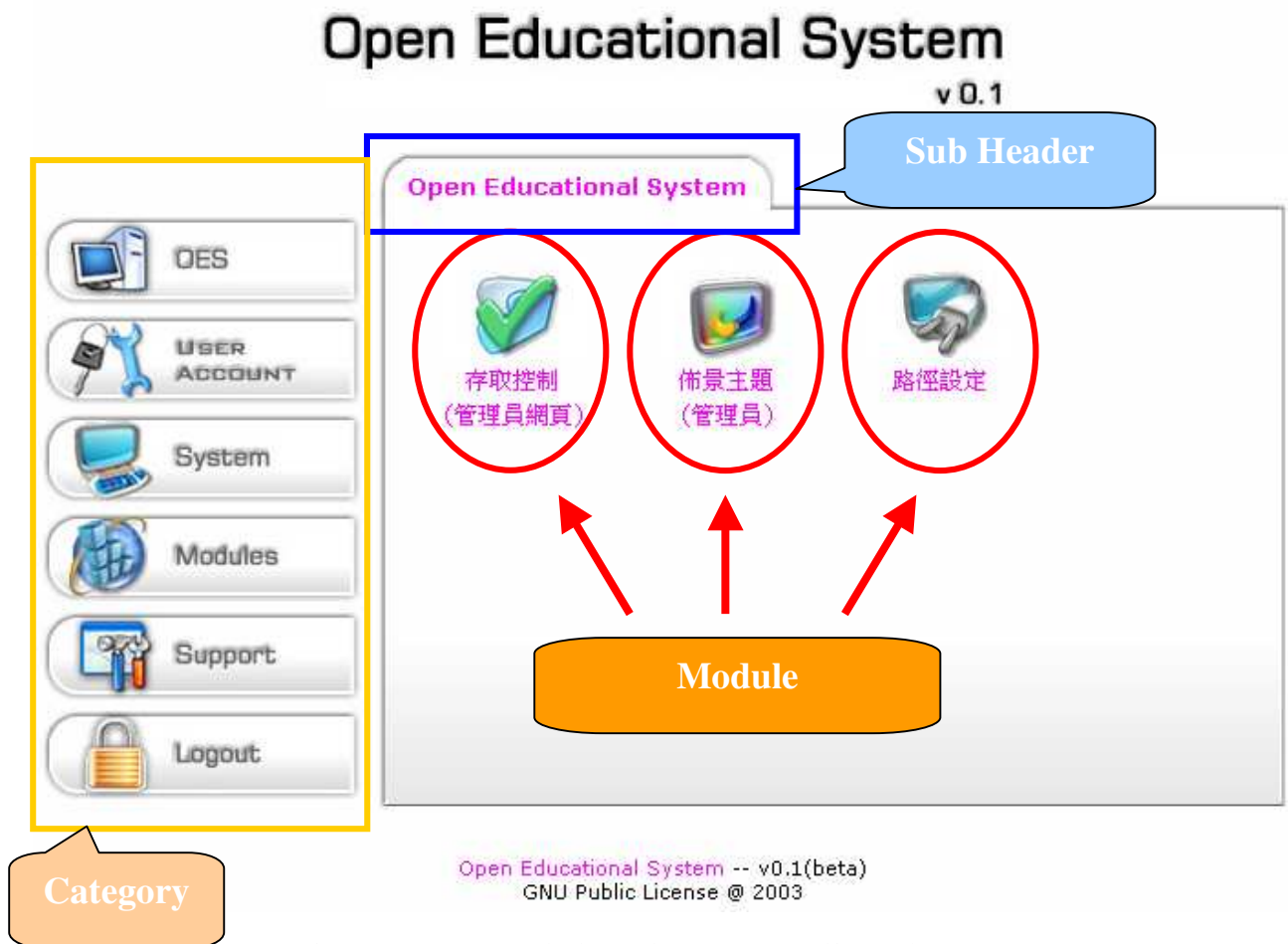


Fig 2.38 – Modules shows on client browser

The above picture shows there is three modules are waiting to be executed. But the modules how to indicate each of them? This page was generated by the Modules Loader, so the link on each module is under control. Here is the link of the above three module.

<http://ic.dyndns.org/admin/modules.php?category=oes&mod=acl>

http://ic.dyndns.org/admin/modules.php?category=oes&mod=admin_theme

http://ic.dyndns.org/admin/modules.php?category=oes&mod=path_config

The Modules Loader will check the variable `$mod` to indicate which module to be executed, and the variable `$category` to identify the module is under which category. The code to identify the module is `/admin/modules.php` (Fig 2.39).

```
//-----  
// Check which modules is available on this category  
//-----  
  
// Try to open the modules directory  
if ( !$d = dir( $category_path ) )  
{  
    // read the sub directory of this category  
    while (false != ($entry = $d->read())) {  
        if ( $entry != "." && $entry != ".." && is_dir($category_path.'/'.$entry) )  
            // Only display the modules which the administrator is allow to enter  
            if ( is_array($modAllow[$category]) && in_array( $entry, $modAllow[$category] ) || $SESS_ID<=1 )  
                $modulesList[$entry] = '';  
    }  
    $d->close();  
}
```

Fig 2.39 – Check the modules is available (Enable)

The next picture (Fig 2.40) shows the code of display the modules ICON on the screen. For more details on the modules, please read the source code of `/admin/modules.php` on Appendix (B).


```

echo "<table width=\\"100%\" border=\\"0\" cellpadding=\\"0\" cellspacing=\\"0\" align=\\"center\">\n";
echo "<tr>\n";

// Dumps all modules contain in this category on the screen
$i = 1;
$colNum = 1;
foreach ( array_keys($modulesList) as $mod )
{
    // Define the image path
    $imgPath = FILE_PHP_LINK.$category.'/'. $mod;
    $urlLink = THIS_FILE_NAME.'?category='.$category.'&mod='.$mod;

    // Display the row
    echo "    <td width=\\"\".(100/ROW_MODULES_NUM).\"%\" align=\\"center\" valign=\\"top\">\n";
    echo "        <table width=\\"100%\" border=\\"0\" cellpadding=\\"0\" cellspacing=\\"0\" align=\\"center\"
    valign=\\"top\">\n";

    // Display the javascript, mouse over.
    // if the module have two set image icon
    if ( $modulesList[$mod]['ICON_MOUSEOVER'] )
        $onMouseOver = "onMouseOut=\\"MM_swapImgRestore()\"
        onMouseOver=\\"MM_swapImage('mod-".$i.",'','',$imgPath.$modulesList[$mod]['ICON_MOUSEOVER']).",1)\\"";
    else
        $onMouseOver = '';

    // Display the modules name with different language
    if ( @include($category_path.'/'. $mod.'/'. $modulesList[$mod]['LANG_DIR'].'/'. $SESS_DEFAULT_LANG.'.lang.php') )
    {
        // try the admin user preferred language first
        $modName = $MODLANG["MODULE_NAME"];
    }
    else
    {
        // all false, use the name defined in the modules configuration file
        $modName = $modulesList[$mod]['NAME'];
    }

    // Display the modules icon and link
    echo "        <tr><td align=\\"center\"><a href=\\"".$urlLink."\" ".$onMouseOver."><img
    src=\\"".$imgPath.$modulesList[$mod]['ICON_MOUSEOUT']."\" alt=\\"".$modulesList[$mod]['NAME']."\"
    name=\\"mod-".$i."\" width=\\"80\" height=\\"60\" border=\\"0\">'\n";
    echo "        <tr><td align=\\"center\"><a href=\\"".$urlLink."\" ".$onMouseOver."><font
    class=\\"font\">".$modName.'</font></a></td></tr>'\n";
    echo "    </td></tr></table>\n";
    echo "    </td>\n";

    // Display the empty block on the last row
    if ( ($i == count(array_keys($modulesList))) && ($colspan = ( ceil ($i/ROW_MODULES_NUM) * ROW_MODULES_NUM )
    - $i ) )
    {
        for ($x=0; $x<$colspan; $x++)
            echo "        <td width=\\"25%\">&nbsp;</td>\n";

        // Print out the html table link break if meet the column
        // limit of each row. (i.e. the limit is set by 'ROW_MODULES_NUM')
        if ( $colNum >= ROW_MODULES_NUM )
        {
            echo "</tr>\n";
            echo "<tr><td height=\\"10px\" colspan=\\"".ROW_MODULES_NUM."\">&nbsp;</td></tr>\n";
            echo "<tr>\n";
            $colNum = 0;
        }

        // Counter
        $i++;
        $colNum++;
    }

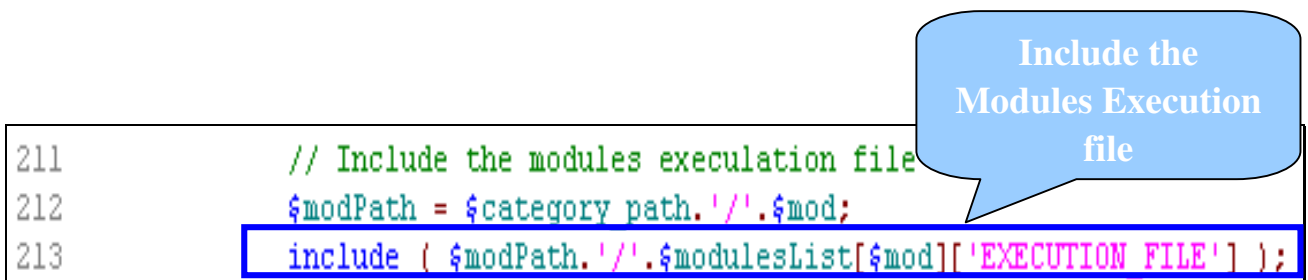
    echo "</tr>\n";
    echo "</table>\n";
}

```

Display
modules
ICON

Fig 2.40 – Display the Modules Icon on the screen

When the user click the ICON of a modules that will execute it. The Modules Loader will check the execution file for that module and include it into the modules loader. Then the user will feel that the module is seem to be execute but has an additional HTML header and footer. Which the header and footer is the design of Theme. Here is the code shows how the modules loader include a modules (Fig 2.41).



```
211 // Include the modules execution file
212 $modPath = $category path.'/'.$mod;
213 include ( $modPath.'/'.$modulesList[$mod]['EXECUTION FILE'] );
```

A blue speech bubble points to the third line of code, containing the text: "Include the Modules Execution file".

Fig 2.41 – Include a Modules (Execute)

The following program fully shows how the module loader executes the module with adding the HTML Header, Sub Title and Footer (Fig 2.42). To have a clearly display, please refers to the Appendix (B), the code can be found on '/admin/modules.php'.

```

156 // ----- //
157 // execute the modules when it was clicked
158 // ----- //
159 if ( isset($mod) && $mod!='' )
160 {
161     // store the modules variable into array
162     // '$modulesList[modules_name][modules_variable]'
163     foreach ( array_keys($MODULES) as $modKeyConf )
164         $modulesList[$mod][$modKeyConf] = $MODULES[$modKeyConf];
165
166     // Try include the language file for this module
167     $modLang = $category_path.'/'.$mod.$modulesList[$mod]['LANG_DIR'].'/'.$SESS_DEFAULT_LANG.'.lang.php';
168     if ( @include($modLang) )
169         $SESS_CURRENT_LANG = $SESS_DEFAULT_LANG;
170     else
171     {
172         // Can't use the user defined language for this modules,
173         // it will try to use the theme default language
174         $modLang = $category_path.'/'.$mod.$modulesList[$mod]['LANG_DIR'].'/'.$THEMES_DEFAULT_LANG.'.lang.php';
175         if ( @include($modLang) )
176             $SESS_CURRENT_LANG = $THEMES_DEFAULT_LANG;
177         else
178         {
179             // can't use the theme default language for this modules,
180             // it will use the default language 'English'
181             $modLang = $category_path.'/'.$mod.$modulesList[$mod]['LANG_DIR'].'/.en.lang.php';
182             $SESS_CURRENT_LANG = $modulesList[$mod]['LANG_DIR'];
183             @include( $modLang );
184         }
185     }
186
187     // Save the current language
188     session_register ( 'SESS_CURRENT_LANG' );
189
190     // The full URL query path of
191     $URL_QUERY = THIS_FILE_NAME.'?category='.$category.'&mod='.$mod;
192
193
194     // ----- //
195     // Load the header and the menu first
196     // Set the sub modules link
197     $subModulesLink = '<a href="'.THIS_FILE_NAME.'?category='.$category.'&mod='.$mod.'">'.ereg_replace( '<br>', '',
198     $MODLANG["MODULE_NAME"] ).' </a>';
199
200     // Display a sub modules quick link in the sub title area
201     // (This is a experimater function, version 0.1 only allow 1 level of sub modules,
202     // i.e. "Category > Modules Name > Sub Modules" and no more)
203     if ( count($modList) )
204     {
205         for ( $i=1; $i<count($modList); $i++ )
206             $subModulesLink .= ' > <a href="'.PHP_SELF.'?category='.$category.'&mod='.implode('|',$modList).'&action='.$action.'">'.ereg_replace(
207             '<br>', '', $modList[$i] ).' </a>';
208     }
209     // ----- //
210
211     $theme->assign( 'SUB_MODULES', ' > '.$subModulesLink );
212     $theme->getHeader();
213     echo $theme->getResult();
214
215     // Include the modules execution file
216     $modPath = $category_path.'/'.$mod;
217     include ( $modPath.'/'.$modulesList[$mod]['EXECUTION_FILE'] );
218 }

```

Check what Language to be display

Display the Modules name on Sub Title

Include the modules (Execute)

Fig 2.42 – Full code to Execute a module

The above was introducing how the modules work on the Modules Loader. Now will be introducing how the modules design. It was mentioned before, a modules was packed by modules configuration '[MODULES]/module.php'; a language file and the template file. From the above, only introduce the modules configuration and the working principal on a module. Here will be talk about the language file design. An example of a language file shows on below '/admin/modules/user_account/ogroup/lang/en.lang.php' (Fig 2.43).



Fig 2.43 – Language file (English)

The above example shows the language file basically is an array call '**\$MODLANG[]**'. The key of the array is an english name. For different language, only the value (right hand side) of the item should be change, following is the Chinese version for the same modules '`/admin/modules/user_account/ogroup/lang/big5.lang.php`' (Fig 2.44).

```

1  <?
2  // File      : /admin/themes/MODULES/user_account/ogroup/lang/big5.lang.php
3  //
4  // Description : Language file for modules 'admin_user' (BIG5 = Chinese Traditional)
5
6  $MODLANG["MODULE_NAME"]      = 'OES 群組';
7
8  $MODLANG["GID"]              = 'ID';
9  $MODLANG["GENABLE"]          = '狀態';
10 $MODLANG["GNAME"]            = '組別名稱';
11 $MODLANG["GDESCRIBE"]        = '組別說明';
12 $MODLANG["START"]            = '啟用';
13 $MODLANG["STOP"]             = '停用';
14 $MODLANG["EDIT"]             = '編輯組別';
15 $MODLANG["GROUP_MESSAGE"]     = '在下面列出的是系統中存在的群組。你可以在上面點選一下，進入另一個選單來更新這個組別的資訊。',
16                               '如果你要建立一個新組別，只要到本頁尾部按<建立新組別>即可。';
17 $MODLANG["EDIT_MESSAGE"]      = '下面是系統內某組別的資訊，你可已在更新後按<儲存>來更新這個組別。如果你想取消這個組別則按<刪除>即可。';
18 $MODLANG["CREATE_MESSAGE"]    = '要建立一個新組別，只要填寫下面的表格後按<建立>即可。';
19 $MODLANG["DELETE_GROUP"]      = '你確定永久地刪除這個組別嗎?';
20
21
22 $MODLANG["RECORD_NOT_FOUND"]  = '找不到記錄。';
23 $MODLANG["DUPLICATE_GNAME"]   = '找到了一個相同的組別名稱，請選用另一個組別名稱代替。';
24 $MODLANG["CANNOT_UPDATE"]     = '不能夠更新記錄。';
25 $MODLANG["CANNOT_DELETE_ROOT"] = '你不能够刪除組別 "ROOT"。';
26 $MODLANG["GNAME_TOO_LONG"]    = '組別名稱不能太長。';
27 $MODLANG["NO_SPACES_ALLOW"]   = '組別名稱不可以包含任何空間。';
28 $MODLANG["CANNOT_CREATE"]     = '新增組別失敗。';
29
30
31 $MODLANG["NONE_RECORD"]       = '沒有任何資料!!!';
32 $MODLANG["ERROR_MESSAGE"]     = '發生錯誤!!!';
33 $MODLANG["CREATE_NEW"]        = '建立新組別';
34
35 $MODLANG["BUTTON_BACK"]       = '返回';
36 $MODLANG["BUTTON_CREATE"]     = '建立';
37 $MODLANG["BUTTON_SAVE"]       = '儲存';
38 $MODLANG["BUTTON_DELETE"]     = '刪除';
39 $MODLANG["BUTTON_CANCEL"]     = '取消';
40
41  >>

```

The NAME of the Module

Fig 2.44 – Language file (Chinese – BIG5)

The item on the language file can be design freedom. But only ONE item must be including in the language file. The item is '**\$MODLANG["MODULE_NAME"]**'. This item will be affecting the name display in the ICON form. The following pictures show the differences (Fig 2.45 & Fig 2.46).

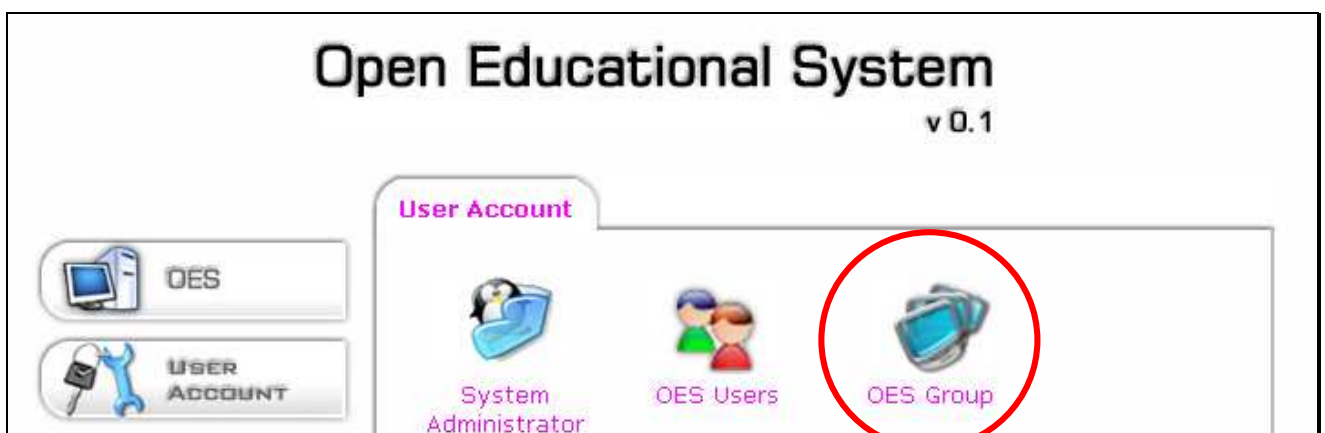


Fig 2.45 – Display on English



Fig 2.46 – Display on Chinese (BIG5)

The other item on the array can be assign freedom, it depend on the designer. But, how to design the Template with different language support? OES has a mechanism to archive this gole. Let look at the code of a Template (Fig 2.47).



Fig 2.47 – Template file example

The above example is come from '/admin/modules/user_account/ogroup/mod_group_template'. The example shows there are two type of variable on that, expect the original HTML code. One is bundle by a square brackets '[]', another is bundle by brackets '{ }'. For a square brackets '[]', is call a *Static Variable*. Which will be assigning by the Template Loader automatically. The main purpose of this *Static Variable* is for the language string. The Template Loader will find the language file and replace this with the template file automatically. Another bracket '{ }' is call *Dynamic Variable*. This is a little bit different from the static variable. The dynamic variable can be assign by the PHP program, that mean the programmer can control what thing will be replace with. The is good for dynamic change the URL, TEXT and etc.

The idea to replace the static variable is, check what language should be display. Then, check the module has the appropriate language file. If the language was not found, try to use the English version. Now the language was found and then try to include it. The memory will have an up-to-date array '\$MODLANG'. Then the Template Loader will try to search the pattern (the key of array bundle by a square bracket) in the template file. If found, replace it with the value of the array

element. Here is the code shows how the Template Loader replace the TEXT with the language file 'lib-template-mod.inc.php' (Fig 2.48).

```

320 // ----- //
321 // Include the modules language file
322 // ----- //
323 GLOBAL $MODULES;
324
325 // Stop when the user selected language file readed
326 if ( !@include( $this->_langPath = $basePath.$MODULES["LANG_DIR"].'/'.$this->lang.'.lang.php' ) )
327 {
328     if ( $this->debug )
329         // Display the can't open language file error message when debug mode is ON
330         echo "Can not open the language file '".$this->_langPath."'<br>\n";
331 }
332 // ----- //
333
334
335 // ----- //
336 // Find the key of string from the language file
337 // ----- //
338 if ( !is_array($MODLANG) )
339     echo "Can not replace with language patten, please check your language file";
340 else
341     @langString = array_keys($MODLANG);
342     for ( $i=0; $i<count($langString); $i++ )
343     {
344         @ $templateContent = str_replace( '['.$langString[$i].']', $MODLANG[$langString[$i]], $templateContent );
345     }
346 // ----- //

```

Replace the patten.
(Language file)

Fig 2.48 – Replace Static Variable (Template Loader)

For the Dynamic Variable, this is assign by the programmer. So a interface to replace the dynamic variable with something is required. Provide by the Template Loader, there is a function call 'assign({Pattern}, Value)' to active this goal. This function can replace the pattern in the template file with any value. Here is the example of usage.

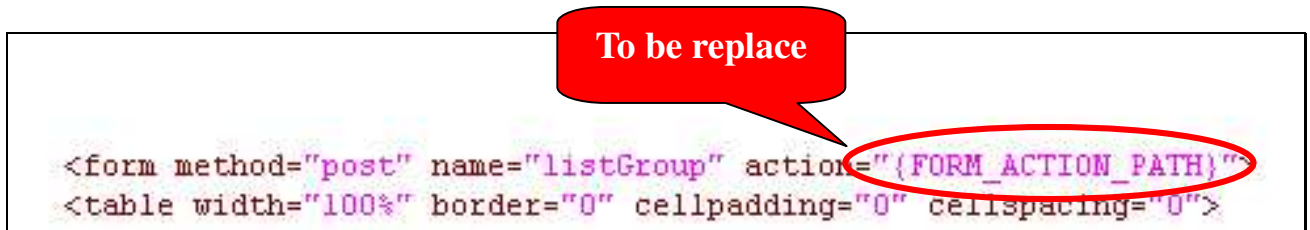


Fig 2.49 – Step 1 : Original HTML Code ‘[mod_group_template](#)’

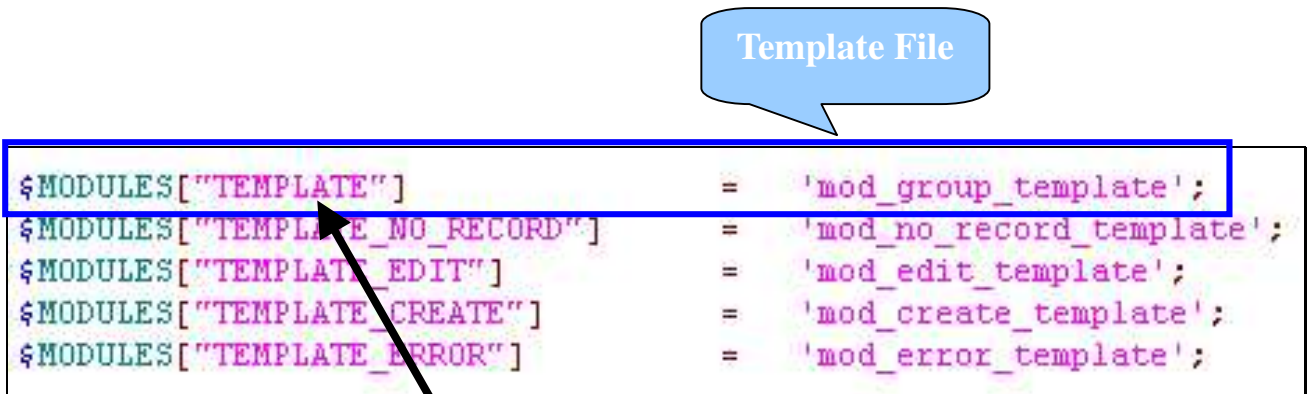


Fig 2.50 – Step 2: Define the template in configuration file ‘[module.php](#)’

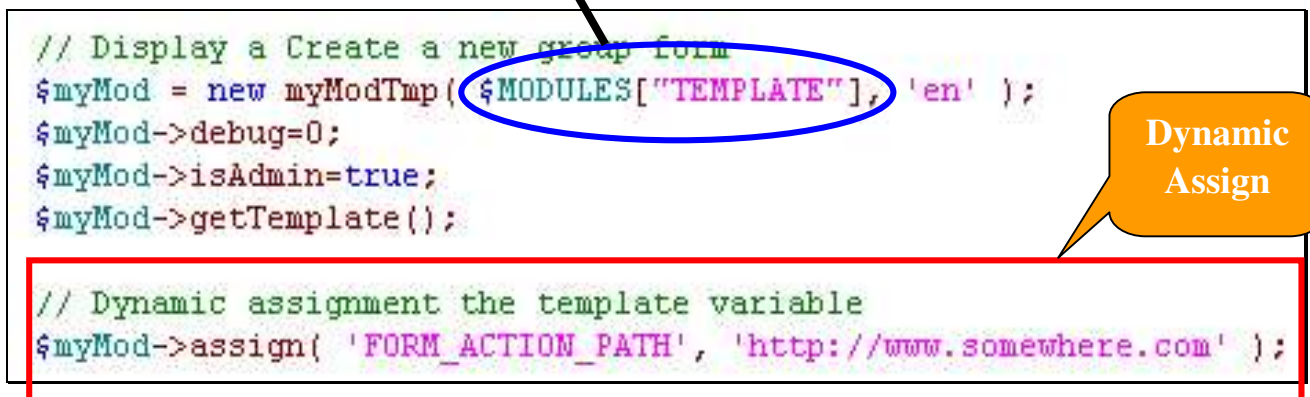


Fig 2.51 – Step 3: Program codes, dynamic assign the variable

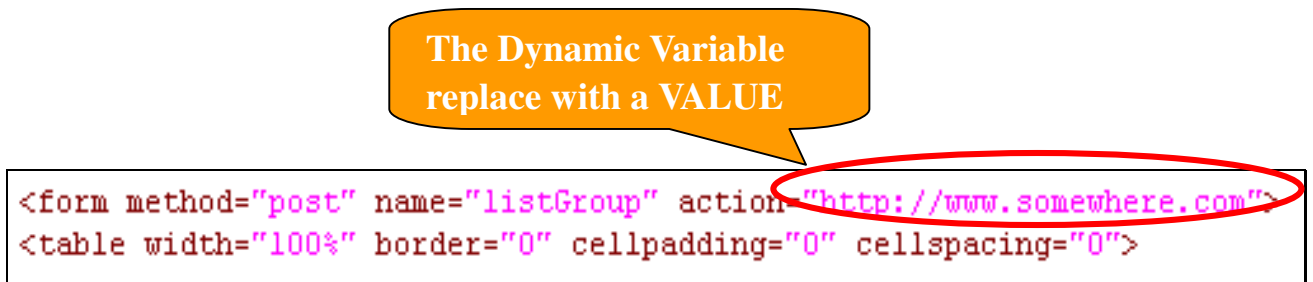


Fig 2.52 – Step 4 : HTML Result

The above four pictures, show the step on the design a dynamic variable. The dynamic assignment is a powerful tool to replace the pattern with anything. Here is the code to do the dynamic assignment (Fig 2.53).

```
// Replace the pattern with the dynamic data.  
$this->_result = str_replace( '({.$pattern.})', $replace, $this->_result );
```

Fig 2.53 – Code to do dynamic assignment the value (Template Loader)

In traditional, the designer wants to repeat *n* times of a part of HTML codes, they must copy those codes *n* times. It is non-convenience and duplicates the codes. Power by the Template Loader, it has a powerful function to prevent the above problem. The template designer can mark the HTML code with a specific coding, which the code only can be process by the template loader but no the client's browser. And then the programmer can write their code to repeat the Segment *n* times

using the function provided by Template Loader. Here is the code adding to quote the original HTML codes (Fig 61).

```
<!-- [ROW NAME="inRow"] -->
    Original HTML code to be repeat N times.....
<!-- [ENDROW NAME="inRow"] -->
```

Fig 2.54 – Repeat a part of HTML codes

The above picture shows there are two line of code adding to the beginning and ending of an original HTML codes. These two line is ignore by the Browser, because the will consider it is the comment string. By the Template Loader using this behavior to design a repeat *n* times system. From the code shows on above, a name must be defined on each ROW, which will be repeating. Remember, the name must the same as the ENDROW to complete a BLOCK of ROW. In the example, the name is 'inRow' (The name is Case-Sensitive). Here shows the step to repeat the block.

```
<!-- [ROW NAME="groupList"] -->
<tr bgcolor="#ffffff" onMouseOver="this.style.backgroundColor='#F4F4F4'; this.style.cursor='hand';"
onMouseOut="this.style.backgroundColor='#ffffff';"
onclick="window.location.href='{EDIT_URL}'">
<td width="30" align="center"><a href="{EDIT_URL}"><font class="font">{GID}</font></a></td>
<td width="50" align="center"><a href="{EDIT_URL}"><font class="font">{GSTATUS}</font></a></td>
<td width="200" align="center"><a href="{EDIT_URL}"><font class="font">{GNAME}</font></a></td>
<td width="300" align="left">&nbsp;<a href="{EDIT_URL}"><font class="font">{GDESCRIBE}</font></a></td>
</tr>
<!-- [ENDROW NAME="groupList"] -->
```

Fig 2.55 – Step 1: Original HTML code '[mod_group_template](#)'


```
// Create a modules template object
$myMod = new myModTmp( $MODULES["TEMPLATE"], $SESS_CURRENT_LANG );
$myMod->debug=0;
$myMod->isAdmin=true;
$myMod->getTemplate();

$myMod->assign( 'FORM_ACTION_PATH',
$requestURL.''.$myMod->getLangVAR('LANG=CREATE_NEW').'&action=create' );

$myMod->assign( 'SORT-BY_GID_URL'      , $requestURL.'&sortby=id|'.$sortmode );
$myMod->assign( 'SORT-BY_GSTATUS_URL'  , $requestURL.'&sortby=status|'.$sortmode );
$myMod->assign( 'SORT-BY_GNAME_URL'    , $requestURL.'&sortby=name|'.$sortmode );
$myMod->assign( 'SORT-BY_GDESCRIBE_URL', $requestURL.'&sortby=describe|'.$sortmode );

for ( $i=0; $i<count($result); $i++)
{
    // Dynamic assignment the template variable
    $myMod->assign( 'GID'      , $result[$i]['id'] );
    if ( strtoupper($result[$i]['group_enable']) == 'Y' )
        $myMod->assign( 'GSTATUS' , 'LANG=START' );
    else
        $myMod->assign( 'GSTATUS' , 'LANG=STOP' );
    $myMod->assign( 'GNAME'      , strtoupper($result[$i]['group_name']) );
    $myMod->assign( 'GDESCRIBE' , $result[$i]['group_describe'] );
    $myMod->assign( 'EDIT_URL'   ,
$PHP_SELF.'?category='.$category.'&mod='.$mod.''.$myMod->getLangVAR('LANG=EDIT').'&action=edit&
id='.$result[$i]['id'] );

    if ( ($i+1) < count($result) )
        // Repeat the ROWs
        $myMod->repeatRow( 'groupList' );
}

// display the final result in form of HTML
echo $myMod->getResult();
```

Dynamic Assign

Dynamic Repeat ROWs (Segment)

Fig 2.56 – Step 2: Program code to Dynamic Assign
and Dynamic Repeat ROWs (Segment)

Before to repeat the ROW, the Template Loader will save the content of a Segment into memory. The Segment is the initialize block, that mean do not replace with the Dynamic Variable. When call the function '[repeatRow\(\)](#)' each time, the

Template Loader will add a initialize BLOCK at the end of the Segment (Fig 2.57). That means adding a block without do the dynamic assignment at the end. But there is one requirement, before adding an empty block to the end of a segment. Before adding an empty block, must do the dynamic assignment to replace all *Dynamic Variable*. Because if don't do this, assign a value to a dynamic variable will replace with the before BLOCK. So this is the only requirement when using the repeat ROWs.



Fig 2.57 – Adding a Initialize BLOCK to the End

Now it was shows the design of a Template and the design of a Module. More examples will be found on next Chapter – 3. For more details on the template API usage, please refer to Appendix (B) – '[lib-template-mod.inc.php](#)'.

Chapter 3

Modules on OES

Chapter 3 will introduce the modules, which come with the OES. This chapter only introduce and show the design theory of the modules where are placing in the category 'Modules'.

3.1 Flash Board

File Structure :

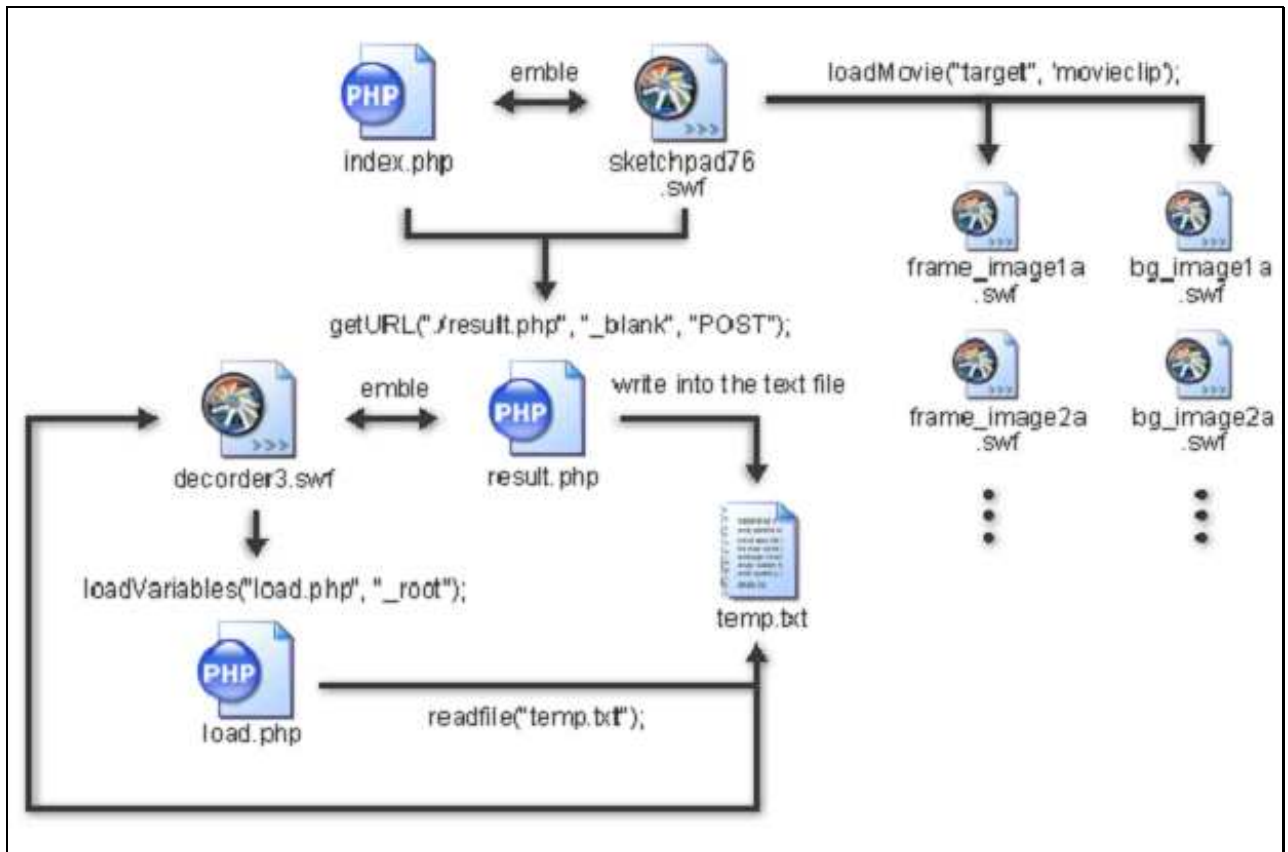
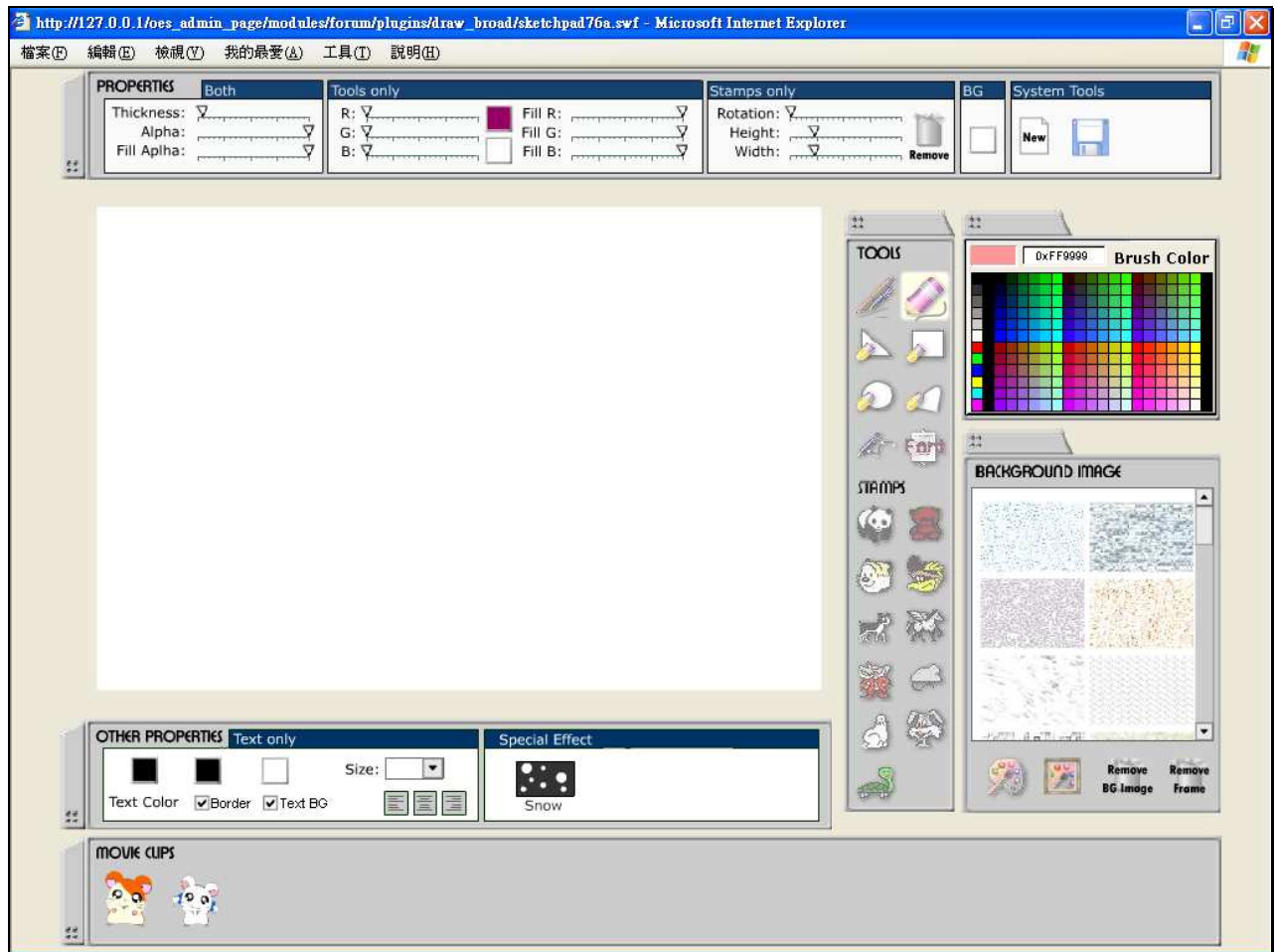










Figure 5 File Structure

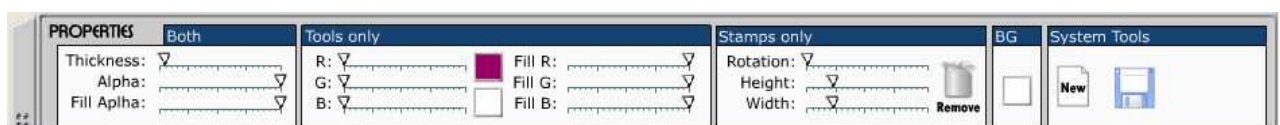
Fig 3.1. – File structure of flash board



Tools Description:

	Draw Straight Line		Draw Free Line
	Rectangle Draw Tool		Triangle Draw Tool
	Draw Circle		Curve Draw with Color Filled
	Curve Line Draw		Insert Text Field

Properties Menu:



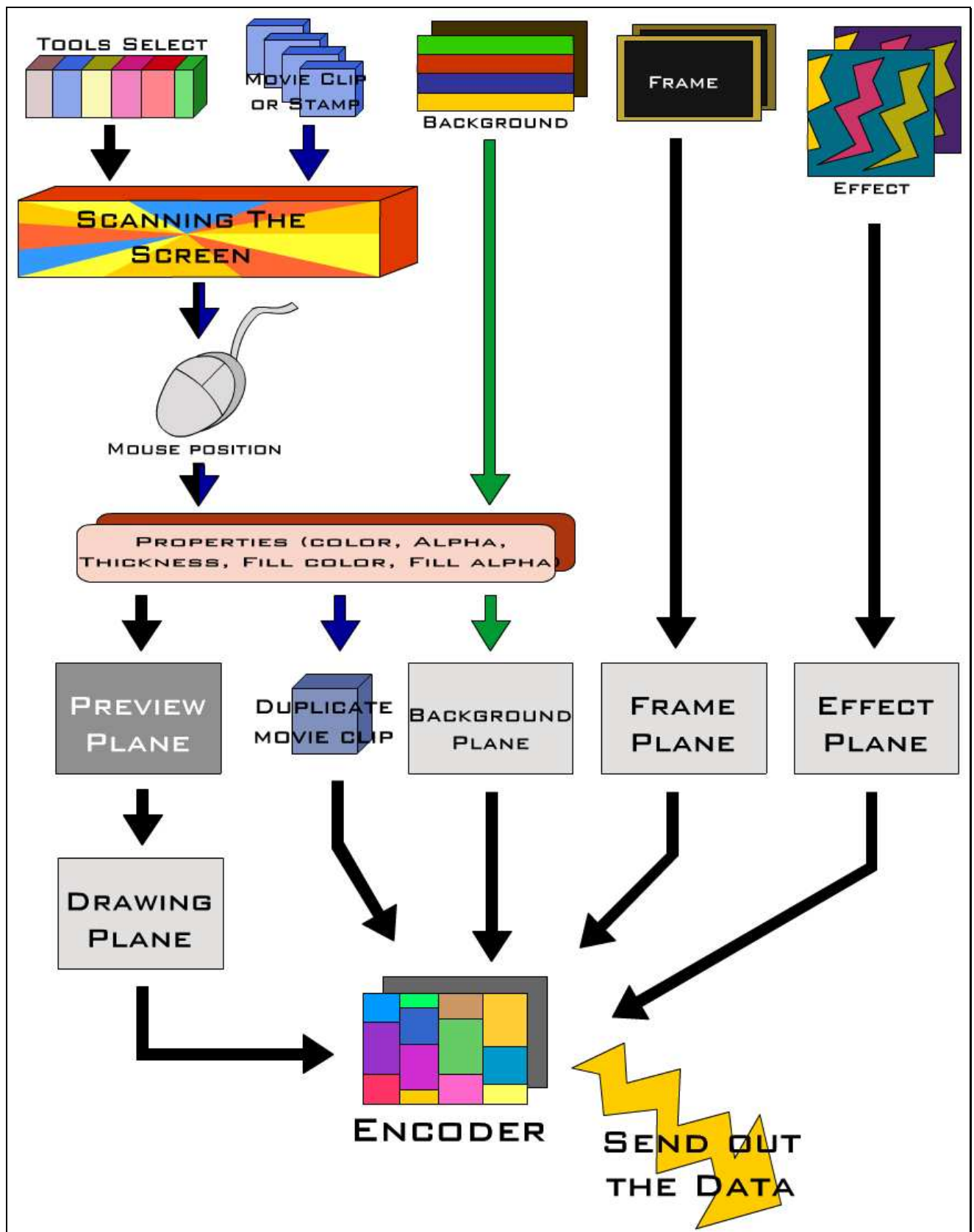


Fig 3.3 – Architecture of the flash board before encode

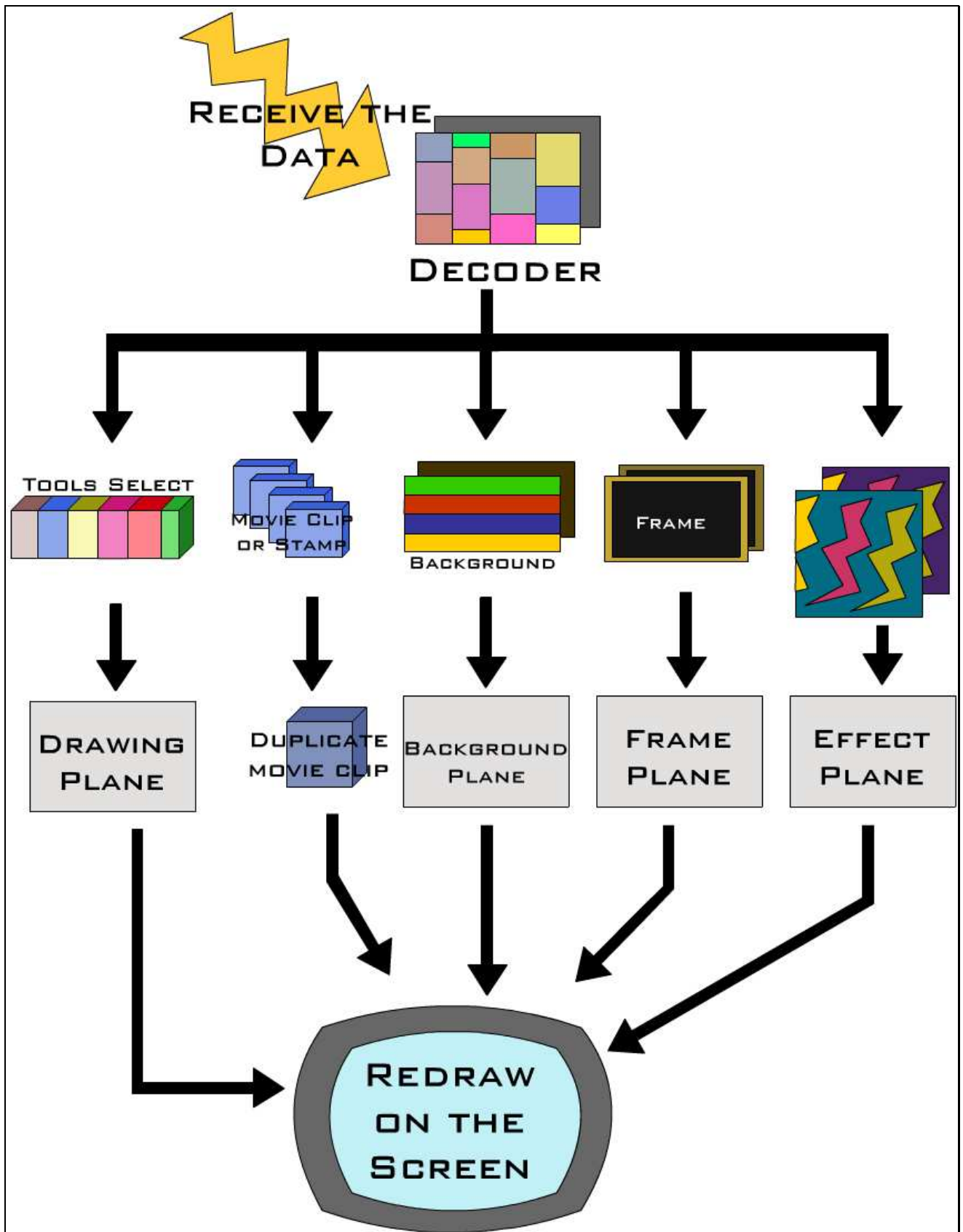


Fig 3.4 – Architecture of the flash board after encode

3.1.1 ActionScript of the Flash Board

At the timeline, there is firstly defining the event when mouse is press and when the mouse press is released.

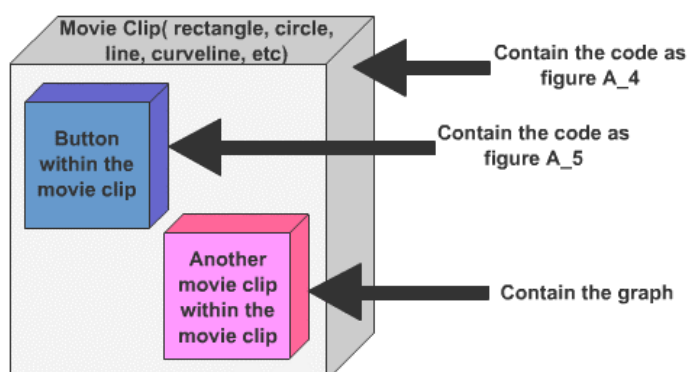
```
_root.onMouseDown = function () { // If mouse button is pressed, set line style, start drawing.
    _root.actions.pendown = true;
    with(_root.drawingPlane){
        lineStyle(_root.actions.thickness, _root.actions.HEXcolor, _root.actions.cA);
        moveTo(_root.actions.lastX, _root.actions.lastY);
    }
}
_root.onMouseUp = function () { // If mouse button is released, stop drawing.
    _root.actions.pendown = false;
}
```

Fig 3.5 – Define mouse up/down function

And then the following is to make sure the draw board is available in drawing mode when the mouse is within the drawing region.

Fig 3.6 – Set the region of drawing plane

```
#####CHECK THE REGION CAN DRAW#####
if (_root._ymouse <= 100 || _root._ymouse >= 541 || _root._xmouse > 688.5 || _root._xmouse < 40 || _root.alert_mode == true) {
    setting = true;
}
else if (_root._ymouse > 100 && _root._ymouse < 541 && _root.alert_mode != true) {
    setting = false;
}
```



The structure of the drawing tool's movie clip(rectangle, circle, curveline, etc)

Fig 3.7 – The structure of the draw tools

In the draw rectangle movie clip, there is the follow code.

```
onClipEvent(enterFrame) {  
    if (_root._xmouse < 670 && _root.actions.rect_press == true) {  
        with(_root.preview_plane) {  
            lineStyle(_root.actions.thickness, _root.actions.HEXcolor, _root.actions.cA);  
            beginFill(_root.actions.F_HEXcolor, _root.actions.fA);  
            moveTo(_root.actions.rect_lastX, _root.actions.rect_lastY);  
           .lineTo(_root._xmouse, _root.actions.rect_lastY);  
           .lineTo(_root._xmouse, _root._ymouse);  
           .lineTo(_root.actions.rect_lastX, _root._ymouse);  
           .lineTo(_root.actions.rect_lastX, _root.actions.rect_lastY);  
            endFill();  
        }  
        if (_root.actions.lastX != _root.actions.X || _root.actions.lastY != _root.actions.Y) {  
            _root.preview_plane.clear();  
        }  
    }  
  
    else if (_root.actions.tool == "null" || _root.actions.tool == "brush") {  
        _root.preview_plane.clear();  
    }  
}
```

Fig 3.8 – The rectangle movie clip code

The highlighted part by using the flash MX drawing API to create the rectangle dynamically, keep in mind that, this rectangle is create into the `_root.preview_plane`. I will explain it later.

In the rectangle movie clip, there is an invisible button. The purpose of it is to mark sure the mouse is click when this rectangle movie clip has been selected. The follow is the code of this button.

```
on(press) {
    _root.actions.rect_lastX = _root._xmouse;
    _root.actions.rect_lastY = _root._ymouse;
    _root.actions.rect_press = true;
    startDrag(this, true, 40, 120, 670, 540);
}
on(release) {
    _root.actions.rect_X = _root._xmouse;
    _root.actions.rect_Y = _root._ymouse;

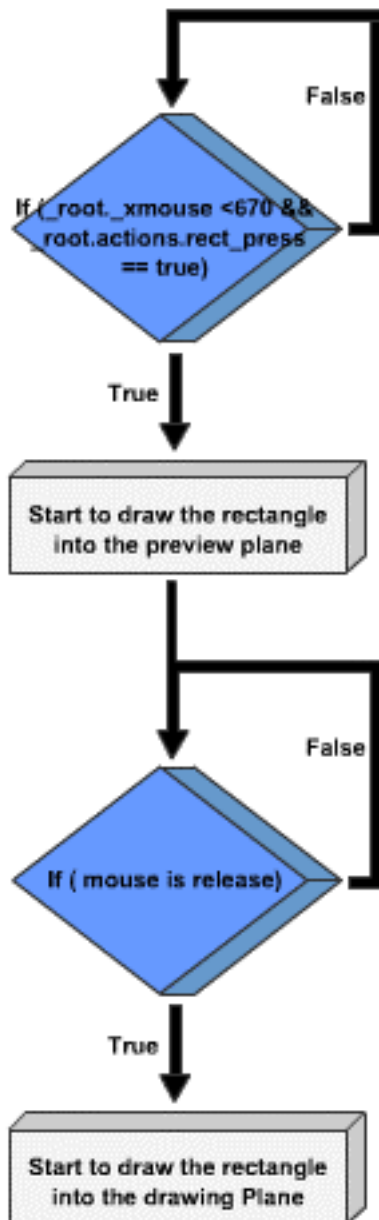
    //#####STORE THE DATA OF RECTANGLE#####
    tellTarget("_root.actions") {
        RectangleData = "R,"+rect_lastX+","+rect_lastY+","+rect_X+","+rect_Y+","+thickness+","+cA+","+HEXcolor+","+F_HEXcolor+","+fA+",";
        FinalData[countNum]= RectangleData;
        countNum++;
    }
    _root.result_field = _root.actions.FinalData;

    with (_root.drawingPlane) {
        lineStyle(_root.actions.thickness, _root.actions.HEXcolor, _root.actions.cA);
        beginFill (_root.actions.F_HEXcolor, _root.actions.fA);
        moveTo( _root.actions.rect_lastX, _root.actions.rect_lastY);
        lineTo( _root.actions.rect_X, _root.actions.rect_lastY);
        lineTo( _root.actions.rect_X, _root.actions.rect_Y);
        lineTo( _root.actions.rect_lastX, _root.actions.rect_Y);
        lineTo( _root.actions.rect_lastX, _root.actions.rect_lastY);
        endFill ();
    }

    _root.actions.rect_press = false;
    stopDrag();
}
on(releaseOutside) {
    _root.actions.rect_press = false;
}
```

Fig 3.9 – The rectangle button code

The highlight part also draws the rectangle, but this rectangle is drawn into the `_root.drawingPlane`, not the `_root.preview_plane`. The purpose of this is to make people can preview the rectangle before release the mouse button. Below is the flow chart of this movie clip.



All the drawing tools are base on this concept of flow chart.

At the same time on drawing the shape into the drawing plane, the data of it will also be saved into a variable (`_root.FinalData`). This variable will be send out or in by `getURL` or `loadVariable` to the PHP program.

Fig 3.10 – Simple flow chart of rectangle draw tool

This variable structure will be liked this.



Fig 3.11 – Structure of encoded data

Example: `FinalData` =

`R,"+X1+", "+Y1+", "+X+", "+Y+", "+thickness+", "+strok alpha+", "+stroke`

colour+"," fill color+"," fill alpha+");

The variable structure

Properties	Values
Type of the Tool	Z, L, U T, R, Q, O,, M, G, N
X, Y coordinates	The coordinates of the point
Properties	Stroke thickness, Stroke Color, Stroke's Color alpha, Fill Color, Fill's Color alpha,

Table 3.1 – list of the parameter

Then by using

\$HTTP_POST_VARS to get

the variable from Flash, and then put the data into the database.

Decode the data

The data which send out is as the form as above, so in the draw board, there is a movie clip which respond to decode the data.

Fig 3.12 – Decoding code

```

if (i <= LengthOfArray) {
    tmp = Final_Data[i].split(",");
    if (tmp[0] == "Z") {
        with (_root.drawingPlane) {
            lineStyle( tmp[5], tmp[7], tmp[6]);
            moveTo( tmp[1], tmp[2]);
            lineTo( tmp[3], tmp[4]);
        }
    }
    if (tmp[0] == "L") {
        with (_root.drawingPlane) {
            lineStyle( tmp[5], tmp[7], tmp[6]);
            moveTo( tmp[1], tmp[2]);
            lineTo( tmp[3], tmp[4]);
        }
    }
    if (tmp[0] == "U") {
        with (_root.drawingPlane1) {
            lineStyle( tmp[7], tmp[9], tmp[8]);
            moveTo( tmp[1], tmp[2]);
            curveTo( tmp[5], tmp[6], tmp[3], tmp[4]);
        }
    }
    if (tmp[0] == "T") {
        with (_root.drawingPlane) {
            lineStyle( tmp[7], tmp[9], tmp[8]);
            beginFill( tmp[10], tmp[11]);
            moveTo( tmp[1], tmp[2]);
            lineTo( tmp[3], tmp[4]);
            lineTo( tmp[5], tmp[6]);
            lineTo( tmp[1], tmp[2]);
            lineTo( tmp[1], tmp[2]);
            endFill ();
        }
    }
    if (tmp[0] == "R") {
        with (_root.drawingPlane) {
            lineStyle( tmp[5], tmp[7], tmp[6]);
            beginFill( tmp[8], tmp[9]);
            moveTo( tmp[1], tmp[2]);
            lineTo( tmp[3], tmp[2]);
            lineTo( tmp[3], tmp[4]);
            lineTo( tmp[1], tmp[4]);
            lineTo( tmp[1], tmp[2]);
            endFill ();
        }
    }
    if (tmp[0] == "Q") {
        with (_root.drawingPlane) {
            lineStyle( tmp[7], tmp[9], tmp[8]);
            beginFill( tmp[10], tmp[11]);
            moveTo( tmp[1], tmp[2]);
            curveTo( tmp[5], tmp[6], tmp[3], tmp[4]);
            endFill ();
        }
    }
}

```

The following actionscript of this movie clip as above. Firstly, it would split the **FinalData** variable by “,” and split again by “.” then it will consider what is the type of this tools should be use. Let see the example of below.

Example:

The following data is containing the value of the rectangle. R,234.95,214.65,554.5,442.4,3,105,0,16711422,105,) By tmp = FinalData[i].splite(“,”) After the splite by “,” , then the value will be store into an array tmp, tmp[0] = R, tmp[1] = 234.95, tmp[2] = 214.65, tmp[3] = 554.5, tmp[4] = 442.4, tmp[5] = 3, tmp[6] = 105, tmp[7] = 0, tmp[8] = 0xFFFF33, tmp[9] = 105

Those values will be putted into the highlight part of the above figure, and then a rectangle can be drawn on the screen.

The other geometry drawings are using the same technique as rectangle.

3.2 Plot Graph

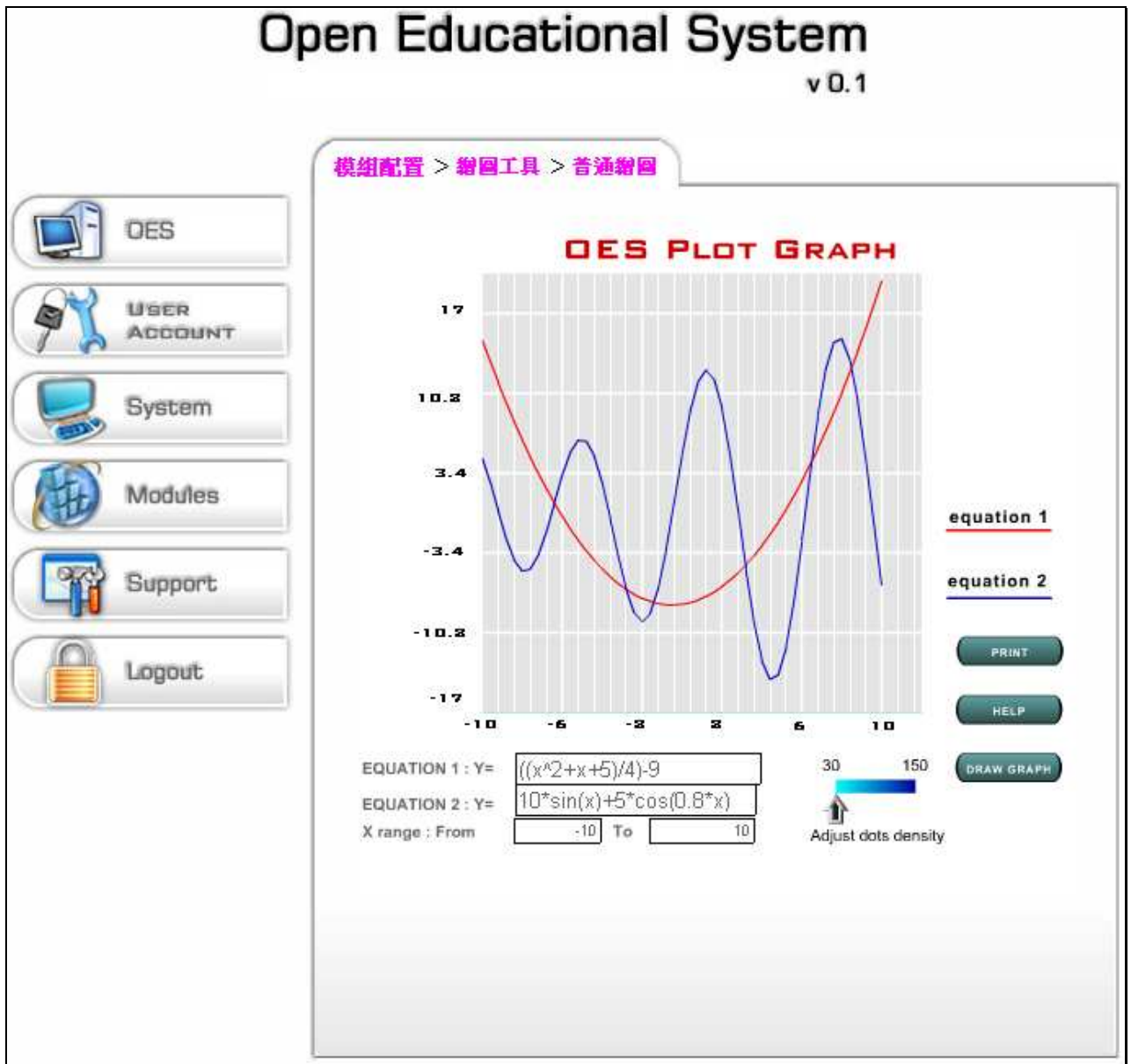


Fig 3.13 – Interface of plot graph module

The above module can help you to plot the x-y coordinate graph by entering the equation. You can draw two equation at the same time, also you can print out the result by clicking the “[print](#)” button.

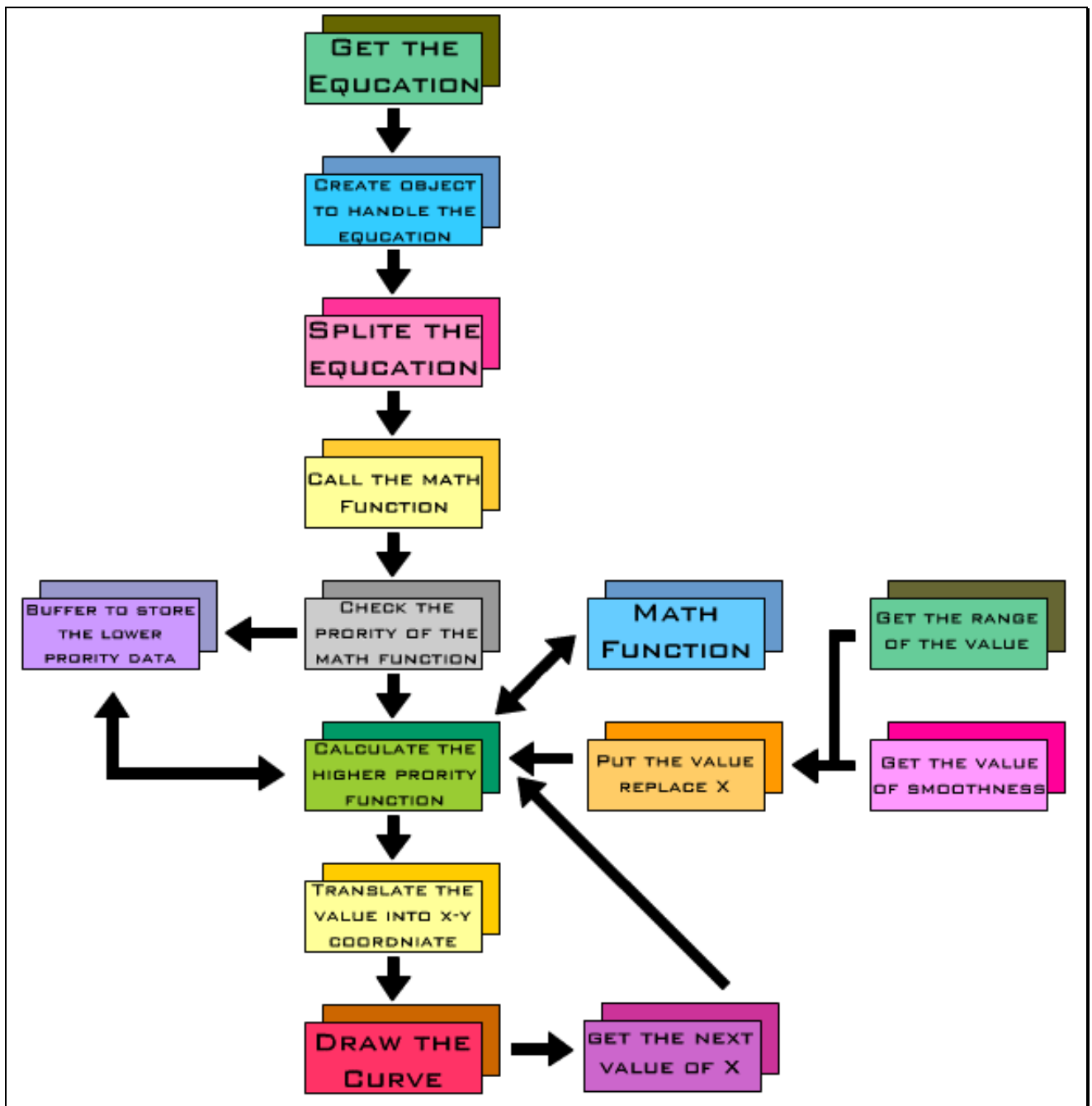


Fig 3.14 – Arch texture of plot graph

3.2.1 *ActionScript of PLOT GRAPH*

```
Math.sum = function (param1, param2) { return (param1+param2);};
Math.subtract = function (param1, param2) { return (param1-param2);};
Math.multiply = function (param1, param2) { return (param1*param2);};
Math.divide = function (param1, param2) { return (param1/param2);};
Math.byPass = function (param1) { return param1;};
Math.pow2 = function (param1, param2) { if ((param1 == 0) && (param2>=0)) {return 0;}
    else if (param1<0) { var p = (((param2+1)%2)*2-1)*Math.pow(0-param1, param2);return p;}
    else {return Math.pow(param1, param2);}};
```

Fig 3.15 – Define the math funtion

At the timeline of the scene, there is defined the some function to replace the math function.

And then define the function of **createObj**, this function can help to create a Object with three properties, **obj.fun**, **obj.param1** and **obj.param2**.

```
function createObj (fun, param1, param2, objIndex) {
    _root["obj"+objIndex] = new Object();
    var obj = _root["obj"+objIndex];
    obj.fun = fun;
    obj.param1 = param1;
    obj.param2 = param2;
}
```

Fig 3.16 – Define of create object function

The following part is using the `lastIndexOf("+")` to return the position of the "+" in the equcation. It can help to consider do the

sum function or not by createObj("sum", xxxx, xxxx,xxxx,xxx), and the value of objIndex is aimed to marked down the object is referred to , such as the pointer in C++.

```
var k = orgTxt.lastIndexOf("+");
if (k>1) {
    objIndex++;
    createObj("sum", orgTxt.substr(0, k), orgTxt.substr(k+1, orgTxt.length-k-1), objIndex);
    return (" $obj"+objIndex);
}
```

Fig 3.17 – Code of calculate the sum

Similarity, the same technique is applied the subtract, multiply, divide and power square.

```
var k = orgTxt.lastIndexOf("-");
if (k>1) {
    objIndex++;
    createObj("subtract", orgTxt.substr(0, k), orgTxt.substr(k+1, orgTxt.length-k-1), objIndex);
    return (" $obj"+objIndex);
}
var k = orgTxt.lastIndexOf("*");
if (k>1) {
    objIndex++;
    createObj("multiply", orgTxt.substr(0, k), orgTxt.substr(k+1, orgTxt.length-k-1), objIndex);
    return (" $obj"+objIndex);
}
var k = orgTxt.lastIndexOf("/");
if (k>1) {
    objIndex++;
    createObj("divide", orgTxt.substr(0, k), orgTxt.substr(k+1, orgTxt.length-k-1), objIndex);
    return (" $obj"+objIndex);
}
var k = orgTxt.lastIndexOf("^");
if (k>1) {
    objIndex++;
    createObj("pow2", orgTxt.substr(0, k), orgTxt.substr(k+1, orgTxt.length-k-1), objIndex);
    return (" $obj"+objIndex);
}
```

Fig 3.18 – Code of calculate the math function

Then the program will start to put the value of the range from the left value to the right left, and each spacing of the value will depend on the smoothness.

By applying the similar technique of flash draw broad to draw the graph.

3.3 Plot Polar

Similar to the Plot Graph module, the below module can help you to plot the polar coordinate graph by entering the equation. Also you can print out the result by clicking the “print” button.

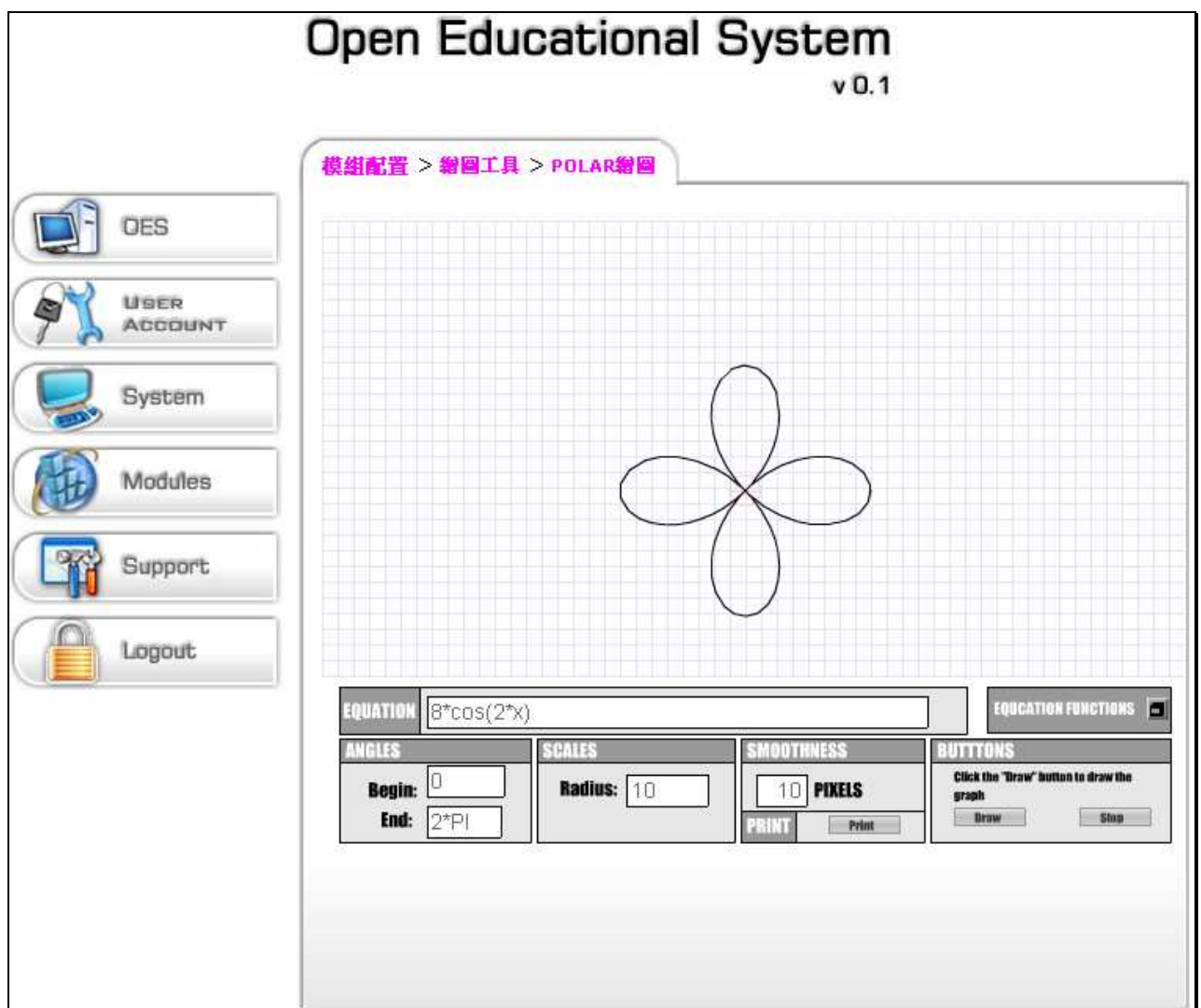


Fig 3.19 – Interface of the polar graph

Chapter 4

Getting start with OES

This chapter will focus on the Installation and Setup process for the OES. How to Install OES on a Linux (Redhat 8) machine, configuration the Apache WWW server and prepare the PHP environment. And also, this chapter will teach the Setup process of OES. Although OES provide a GUI setup wizard, there is few RULE must be follow.

4.1 Installation of OES

OES is mainly focus on a Linux Platform, although the system was designed to cover most system; including Windows. Basically, the system should be work on Windows, but it is not tested. The system was tested on RedHat 7.3, the Apache version 1.3.23 and PHP 4.1.2. And also it was tested on RedHat 8.0, the Apache version 2.0.40 and PHP 4.2.2. Both machine are working with OES perfectly.

4.1.1 Download the OES

Go to the OES official website download the latest version of OES '<http://openeducation.sourceforge.net/>'. The current version is 0.1 (beta). Download the file '[oes-0.1b.src.tar](#)' from the above web site or find it in the CD appends in this report.

4.1.2 *Install OES on Linux*

1. Extract the package.

```
mv oes-0.1b.src.tar /usr/local/  
cd /usr/local  
tar xvf oes-0.1b.src.tar
```

2. Change the owner of the package

```
cd /usr/local  
chown apache:apache oes-0.1b -R  
chmod 755 oes-0.1b -R  
chmod 777 oes-0.1b/conf -R  
chmod 777 oes-0.1b/session -R
```

Assume the system has a user and group of Apache. The above command will force the Apache to be the owner of OES. That mean only the ROOT and Apache can fully control the OES.

3. Configuration the PHP environment.

Please make sure the following line is **ON** in the PHP configuration file.

/etc/php.ini

```
short_open_tag = On
register_globals = On
magic_quotes_gpc = On
```

4. Configuration the Apache 2.0.40.

This is the important process to make OES work on the apache server. Assuming the system has a DNS server pointing to it (e.g. oes.eie.polyu.edu.hk). Then it can configuration the Apache server working on a virtual hosting. The IP_ADDRESS and PORT are depends on the server. If the server work on a standard port (i.e. 80), the PORT can be empty. Line 4, is the alias of the admin control page. If the system do no have more than one sub domain name. Then, line 4 is required, otherwise can create another virtual host point to the admin control page.

/etc/httpd/conf/httpd.conf

```
1: <VirtualHost IP_ADDRESS:PORT>
2: DocumentRoot /var/www/hd2_project/users/htdocs
3: ServerName oes.eie.polyu.edu.hk
4: Alias /admin /var/www/hd2_project/admin
5: DirectoryIndex index.php index.html
6: AddDefaultCharset ISO-8859-1
7: </VirtualHost>
```

After do such things, restart the Apache server.

```
[root@linux]# /etc/rc.d/init.d/httpd restart
Stopping httpd:          [ OK ]
Starting httpd:          [ OK ]
```

5. Create a Database on the database server.

Here is the example to create a database on MySQL.

Type the command 'mysql -u DB_USER -p' in linux console. Remember the 'DB_USER' is the account name to access the database system; 'DB_NAME' is the name of database contains the tables.

```
[root@linux]# mysql -u DB_USER -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or
\g. Your MySQL connection id is 16 to server version:
3.23.55

Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.

mysql> create database DB_NAME;
Query OK, 1 row affected (0.00 sec)
```

The configuration was successful now !!

4.1.3 Start OES in browser

Open a URL '<http://oes.eie.polyu.edu.hk/>' in a browser. A setup wizard will be show on the screen.

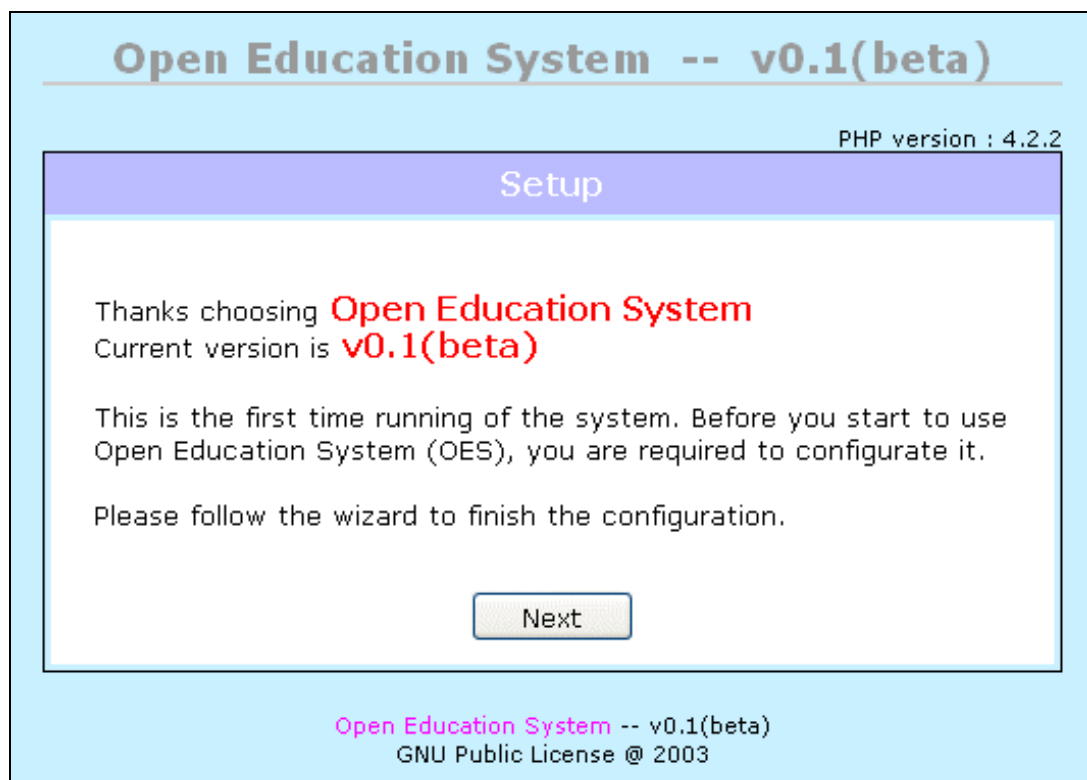


Fig 4.1 – Installation of OES

Click the Next button to continous.

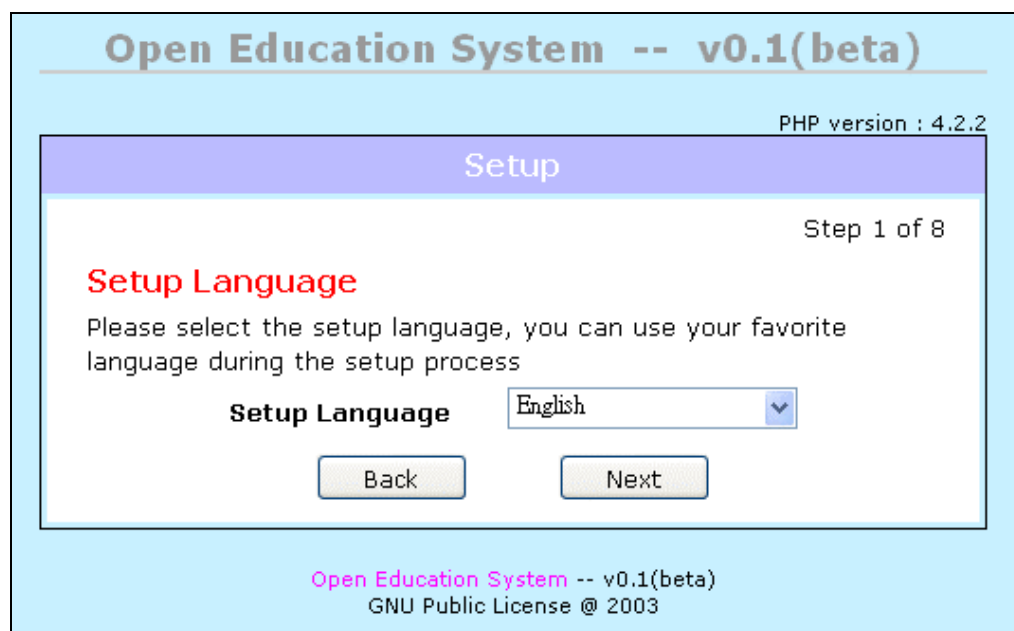


Fig 4.2 – Choose language

Step 1: Change the language during the Setup process

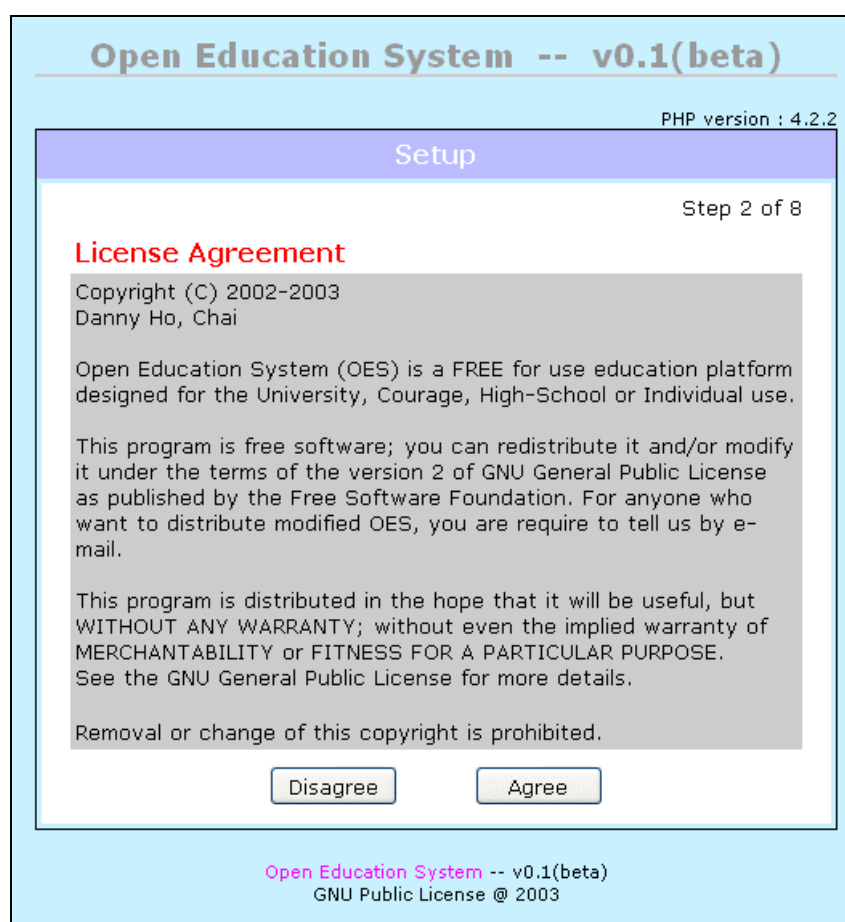


Fig 4.3 – License of OES

Step 2: License Agreement.

Open Education System -- v0.1(beta)

PHP version : 4.2.2

Setup

Step 3 of 8

Database Setup

This system must work with a SQL base database system. You must provide a database account for the system access to the database server. OES are support the following database server, please select any ONE database system you want the OES work with.

Database Type MySQL (Default) ▼

Back Next

Open Education System -- v0.1(beta)
GNU Public License @ 2003

Fig 4.4 – Choose Database

Step 3: Select the Database system to be use with OES.

Database System	Test with OES
Microsoft SQK server	Fail (Don't test it)
Microsoft Visual Fox	Fail (Don't test it)
MySQL	OK
ODBL	Fail (Don't test it)
Oracle	Fail (Don't test it)
Oracle 8	Fail (Don't test it)
PostgreSQL	OK

Type of Database is supported

Open Educational System -- v0.1(beta)

PHP version : 4.2.2

Setup

Step 4 of 8

Database Setup -- MySQL

Please complete the following. (*) **require**

Server IP or Domain
Please provide the database server IP or Domain name, otherwise use default setting.

Server Port
Please provide the port that database server work on. Empty it for default port number.

Database Name *
Please provide the database name. (e.g. OES)

Account *

Password *

Open Educational System -- v0.1(beta)
GNU Public License @ 2003

Fig 4.5 – Enter Database information

Step 4: Database and Account Setting.

Server IP or Domian	The IP or domain of the database server.
Server Port	The PORT of the database server. (Empty to use default)
Database Name	Enter the database name. (e.g. OES).
Account	The account can access the database.
Password	The password of the above account.

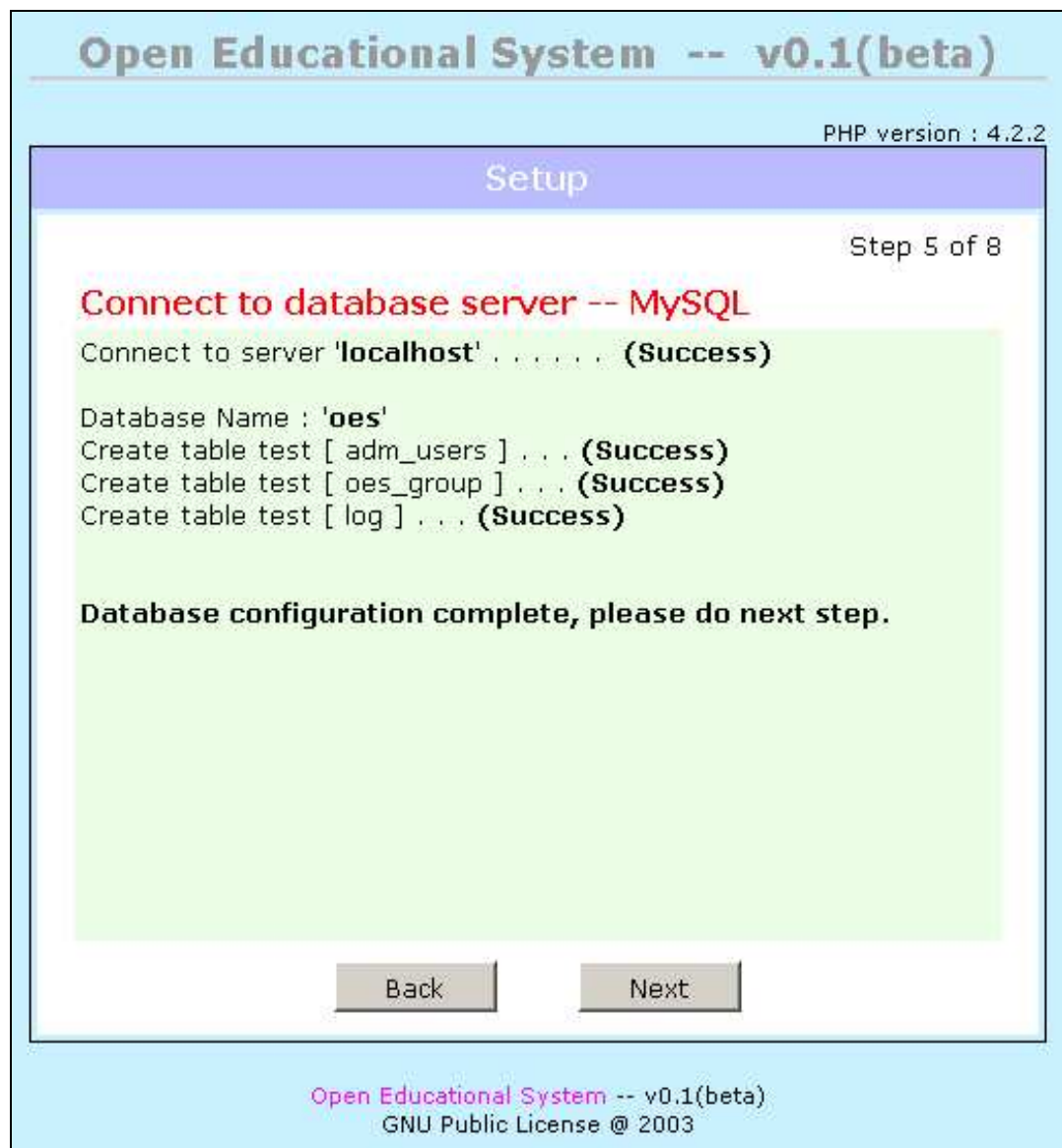


Fig 4.6 – Database create success

Step 5: Test the database 'OES' successful.

If the database 'OES' is not ready, it will display the error and do not allow go to next page.

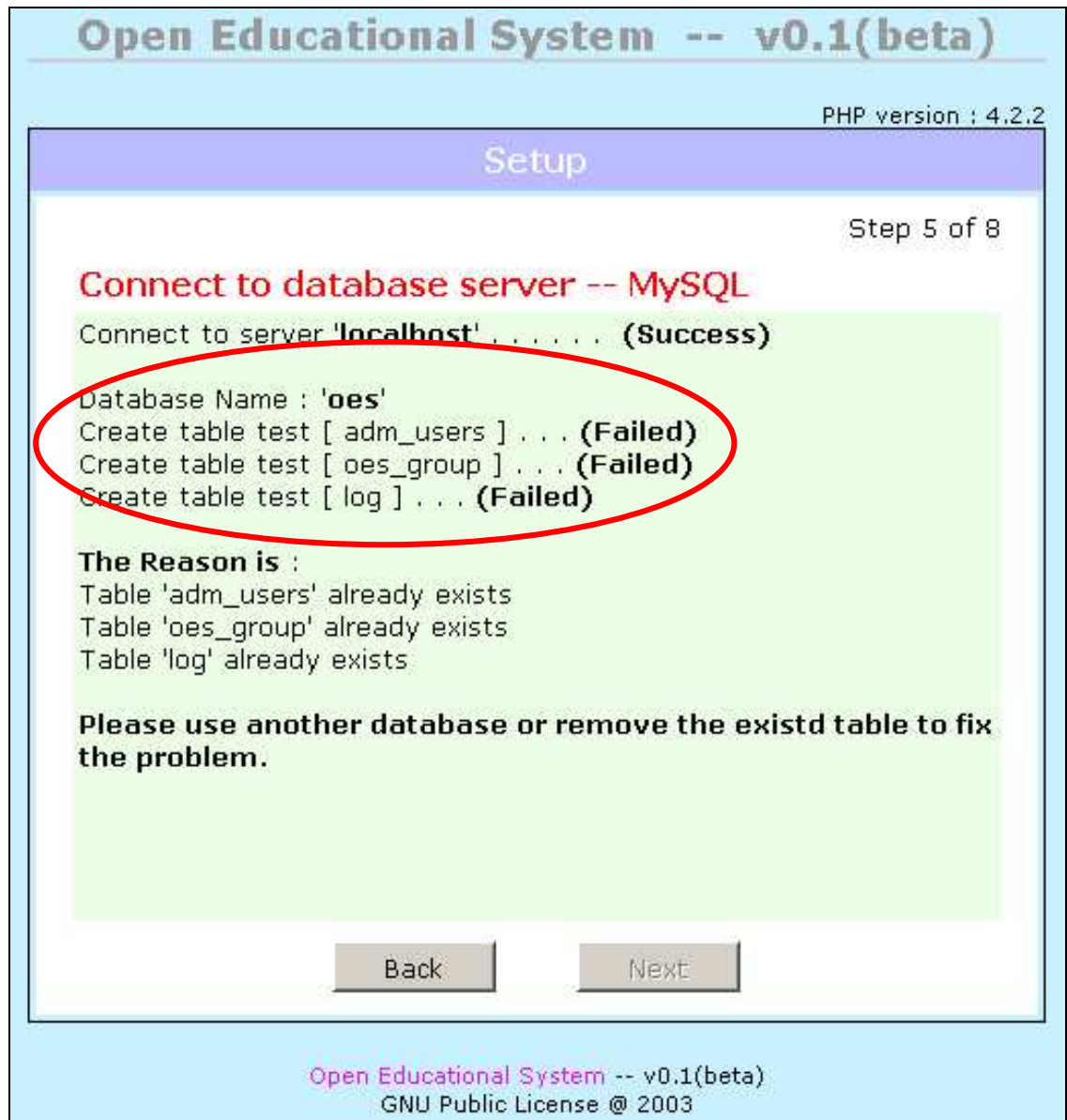


Fig 4.7 – Error Screen of creating database

Open Educational System -- v0.1(beta)

PHP version : 4.2.2

Setup

Step 6 of 8

System Environment

OES require the absolutely path which OES was installed on. By default, the setup utility will try to retrieval the system path, but may not be accurate. So you should change the following field if needed.

Install path *

The following is the OES internal paths, only change it if needed. Normally no need to change.

Config file path *

Include path *

User Themes path *

User Modules path *

Admin path *

Admin Themes path *

Admin Modules path *

Session path *

Session path [Admin] *

The following is the OES URL setting. If you provide wrong URL, the OES will not be work. Please read the documentation for details. Here is the examples.

OES URL = http://your.domain

Administration URL = http://your.domain/admin

OES URL *

Administration URL *

Open Educational System -- v0.1(beta)
 GNU Public License @ 2003

Fig 4.8 –Display the system information

Step 6: Checking the PATH. The path is check by the wizard

automatically, but there may not be correct.

Open Educational System -- v0.1(beta)

PHP version : 4.2.2

Setup

Step 7 of 8

Administrator Account

Now, please complete the administrator information form. This person will be the root administrator. ONLY he/she can FULL control the OES. So don't forget the login id and password. If you was lost the password, you can use your provided e-mail account to receive the new password generated by OES. Otherwise, please read the documenations 'How to active the default administrator account'.

Admin Login ID	<input type="text" value="admin"/>	*
Password	<input type="password" value="****"/>	*
Password Again	<input type="password" value="****"/>	*
Language	English (Default)	
E-Mail Address	<input type="text" value="admin@test.com"/>	*

Open Educational System -- v0.1(beta)
GNU Public License @ 2003

Fig 4.9 –Create admin account

Step 7: Enter the ROOT administrator login ID and password.

Open Educational System -- v0.1(beta)

PHP version : 4.2.2

Setup

Step 8 of 8

Confirmation

Please check the following. When you found anything was wrong, you should go back and make the correction. If all the things are correct, you should click to complete the setup process. This utility will generate the configuration file automatically.

Database Setup

Database Type	= MySQL
Server IP or Domain	= Default
Server Port	= Default
Database Name	= oes
Account	= oes
Password	= oes

System Environment

Install path	= /home/system/var/www/hd2_project
Config file path	= /conf
Include path	= /include
User Themes path	= /users/themes
User Modules path	= /users/modules
Admin path	= /admin
Admin Themes path	= /admin/themes
Admin Modules path	= /admin/modules
Session path	= /session
Session path [Admin]	= /session/admin
OES URL	= http://oes.eie.polyu.edu.hk
Administration URL	= http://oes.eie.polyu.edu.hk/admin

Administrator Account

Admin Login ID	= admin
Password	= Encrypted
E-Mail Address	= admin@test.com

Click to generate the configuration file.

Back
Next

Open Educational System -- v0.1(beta)
 GNU Public License @ 2003

Fig 4.10 –Confirmation the information

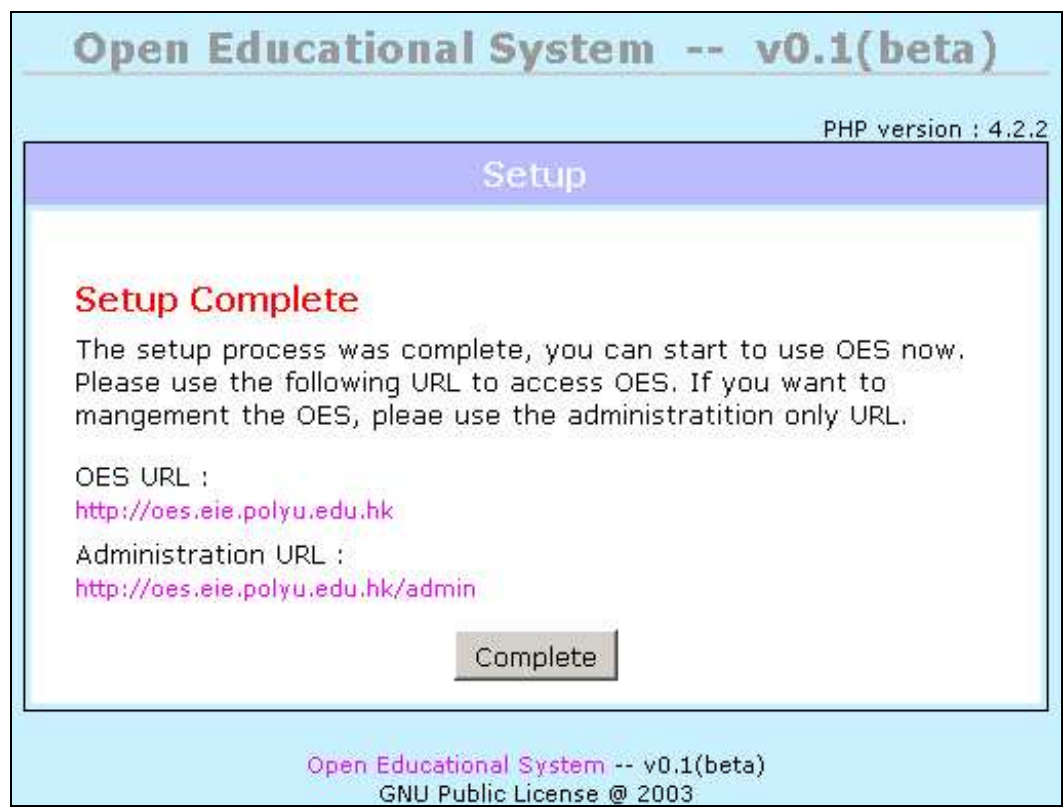
Step 8: Confirm the setting.

Fig 4.11 – Setup complete

Step 9: Finish.

4.2 Administration Guide

Go to the URL <http://oes.eie.polyu.edu.hk/admin> in Browser and log in to the administrator control page using the account and password created as before.



The screenshot shows the login interface for the Open Educational System (v0.1). The title "Open Educational System" is at the top right, with "v 0.1" below it. The main heading is "管理員登入" (Administrator Login). Below this, there are two input fields: "帳號" (Username) and "密碼" (Password), each followed by a colon and a text box. A "GO" button is positioned to the right of the password field. At the bottom, it says "Open Educational System -- v0.1(beta)" and "GNU Public License @ 2003".

Fig 4.12 – Interface of the admin’s login page

4.2.1 *Change System Default Theme*

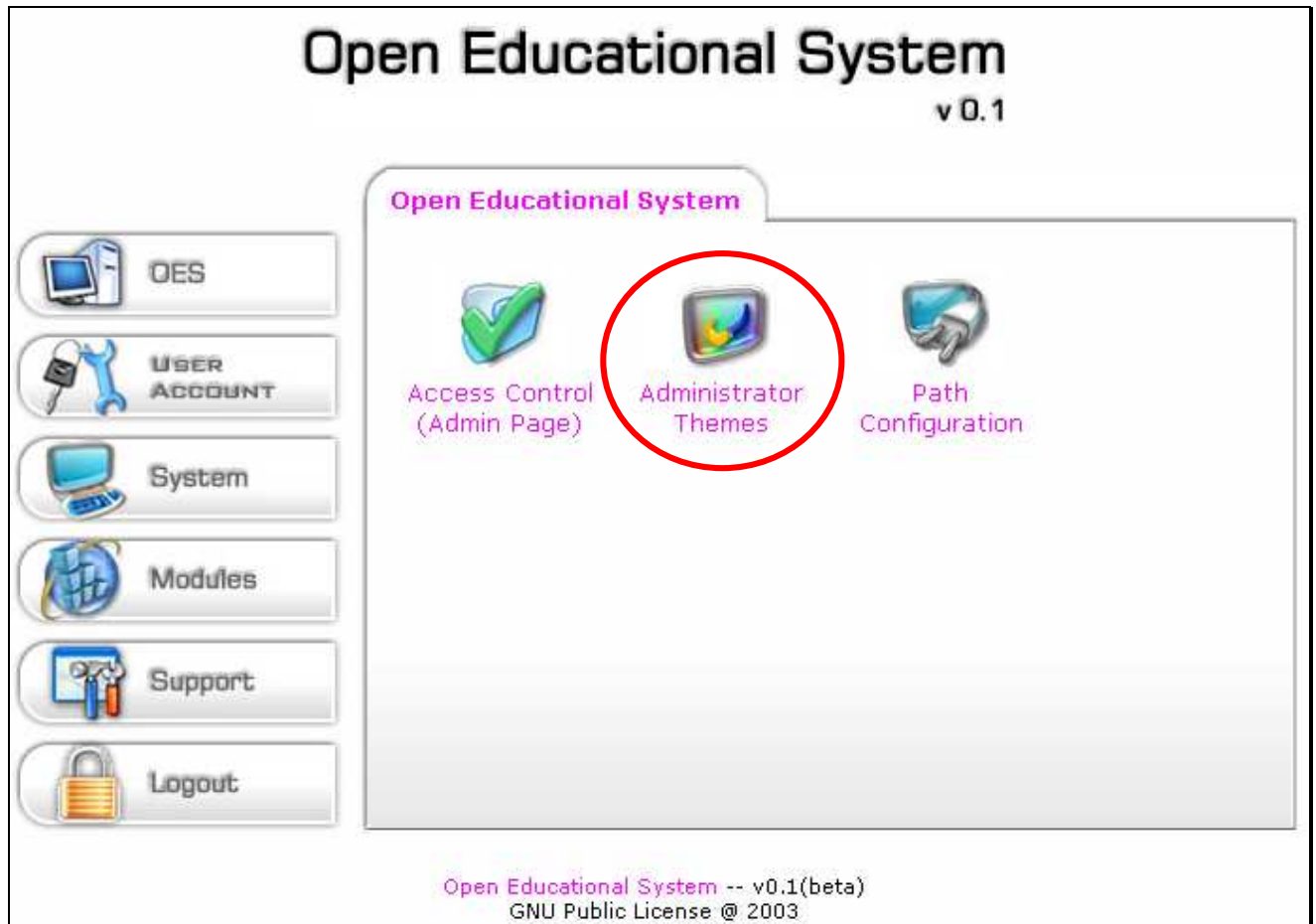



Fig 4.13 – Interface of the admin's page

Click the Theme Setting button , to change the default system Theme.

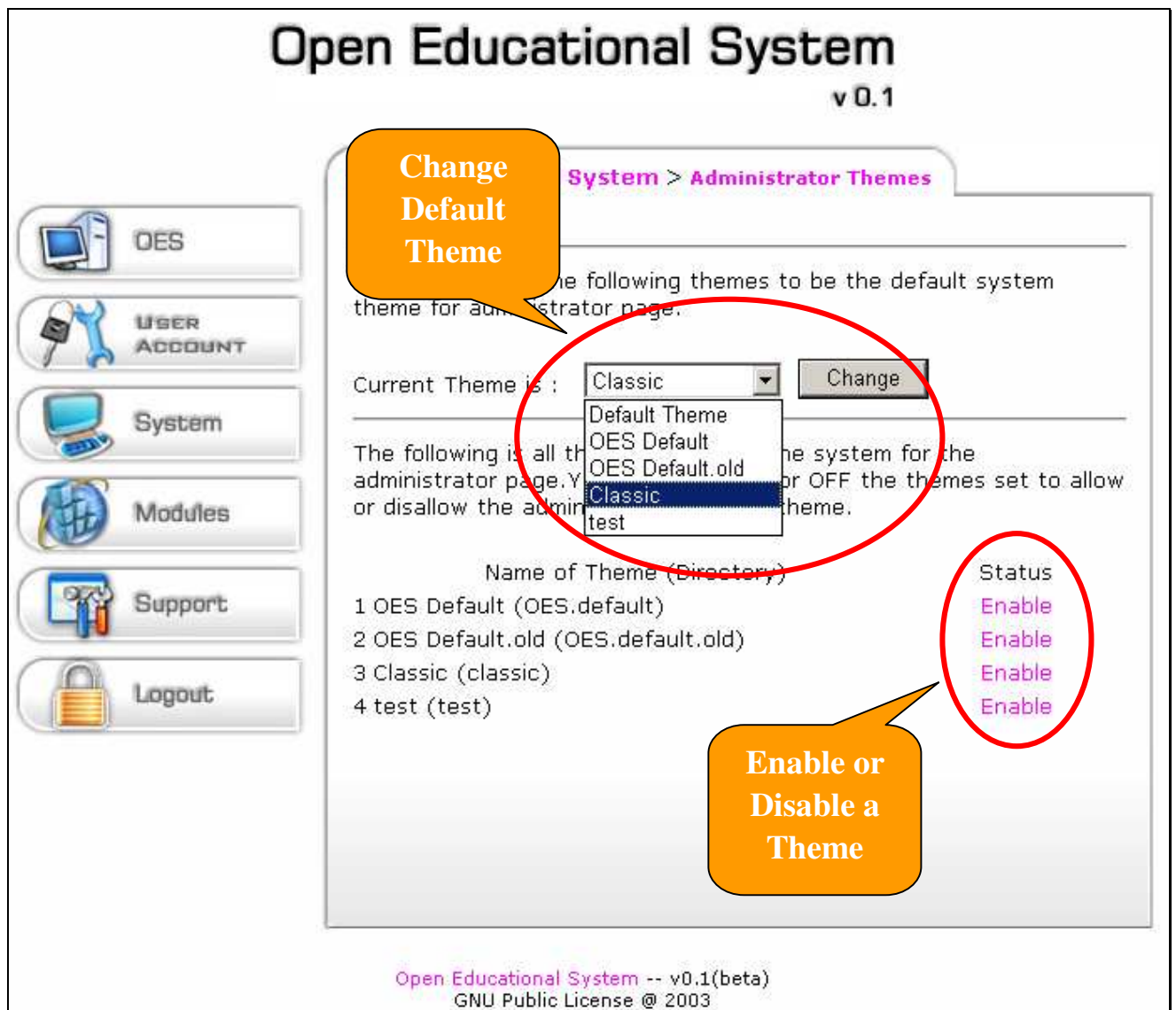


Fig 4.14 – Choosing the Current Theme

To change the default theme, by select a new default and click the button '**Change**'.

And also the administrator can '**Enable**' or '**Disable**' a Theme installed on the system.

4.2.2 Update the system LINK

Enter the Path Configuration button by click



Open Educational System

v 0.1

-  OES
-  USER ACCOUNT
-  System
-  Modules
-  Support
-  Logout

Open Educational System > 路徑設定

請注意!! 在更改以下路徑前，請務必確定路徑是正確無誤的。因為其中一個路徑的錯誤，都可能會影響整個系統的運作，嚴重時整個系統將會癱瘓，直至你把錯誤全部更正為止。

如果你有需要自行更改這個系統設定檔，請到下面的聯結：

/var/www/hd2_project/conf/config.inc.php

安裝路徑 *

系統設定檔路徑	<input type="text" value="/conf"/>	*
程式集路徑	<input type="text" value="/include"/>	*
使用者樣板路徑	<input type="text" value="/users/themes"/>	*
使用者模組路徑	<input type="text" value="/users/modules"/>	*
管理員路徑	<input type="text" value="/admin"/>	*
管理員樣板路徑	<input type="text" value="/admin/themes"/>	*
管理員模組路徑	<input type="text" value="/admin/modules"/>	*
Session path	<input type="text" value="/session"/>	*
Session path [Admin]	<input type="text" value="/session/admin"/>	*

OES 網址 = <http://your.domain>
 管理員網址 = <http://your.domain/admin>

OES 網址 *

管理員網址 *

Open Educational System -- v0.1(beta)
 GNU Public License © 2003

Fig 4.15 – Edit/View the configuration information

4.2.3 Create / Modifier the administrator Account

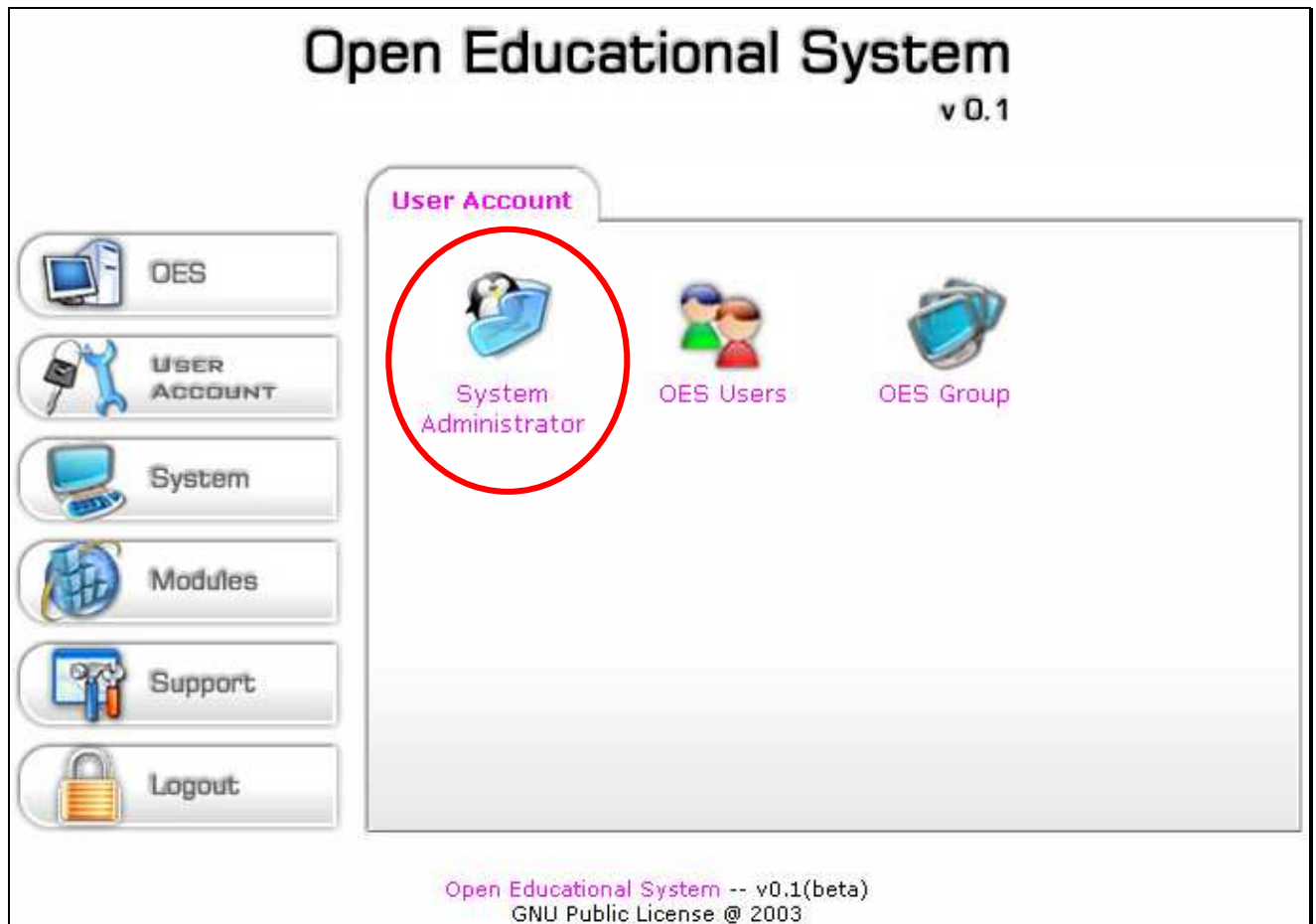



Fig 4.16 – Interface of user account page

Click the System Administration button , to create the administrator account or list the account installed on the system.

To create a new administrator account click



OES

USER ACCOUNT

System

Modules

Support

Logout

Open Educational System

v 0.1

User Account > System Administrator > Create Admin Account

You are writing the form to create a administrator account. Please reminded that any of you provided information will limit the activites of this account.

Status

☒ Allow ☐ Deny

Expire Date

9999-12-31 23:59:59 (YY-MM-DD hh:mm:ss)

Login ID*

New Password*

Password again*

Account Name

E-Mail

Description

Group*

EIE-42075-1 (4)
EIE-42075-2 (8)
EIE-71202-1 (6)
EIE-71202-2 (7)
NO_SPECIAL (Disabled)
NO_SPECIAL2 (Disabled)

AddRemove

Theme

Language

IP Access List

Modules Allow

== OES ==
☐ Access Control (Admin)
☐ Admin Page Theme
☐ Path Configuration

== USER_ACCOUNT ==
☐ Admin User
☐ OES Users
☐ OES Group

== SYSTEM ==
☐ Access Control
☐ Database Configuration
☐ Log Record
☐ Session
☐ User Theme

== MODULES ==
☐ Internal Messenger
☐ News System
☐
☐ News System
☐ News System
☐ News System
☐ News System
☐ Forum Config
☐ Plot Graph
☐ News System
☐ News System


== SUPPORT ==
None of modules!

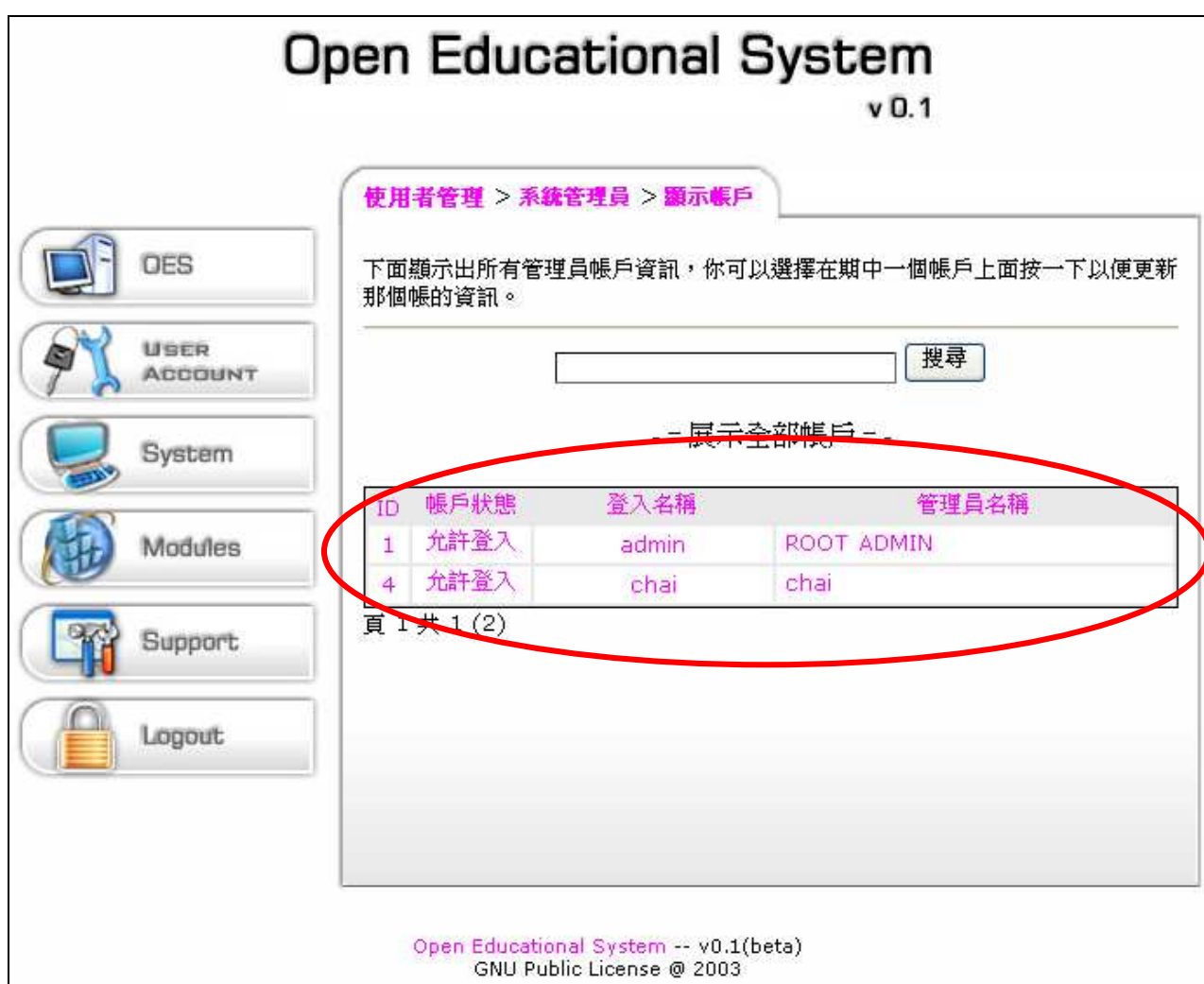
Create a new account

Cancel

Open Educational System -- v0.1(beta)
GNU Public License © 2003

Fig 4.17 – Create admin account

Or modifier the administrator account click  . The administration can edit or delete the account by select the account list on below.



Open Educational System
v 0.1

使用者管理 > 系統管理員 > 顯示帳戶

下面顯示出所有管理員帳戶資訊，你可以選擇在期中一個帳戶上面按一下以便更新那個帳的資訊。

搜尋

- 顯示全部帳戶 -

ID	帳戶狀態	登入名稱	管理員名稱
1	允許登入	admin	ROOT ADMIN
4	允許登入	chai	chai

頁 1 共 1 (2)

Open Educational System -- v0.1(beta)
GNU Public License @ 2003

Fig 4.18 – View/ Edit account

4.2.4 Another Theme on admin control page

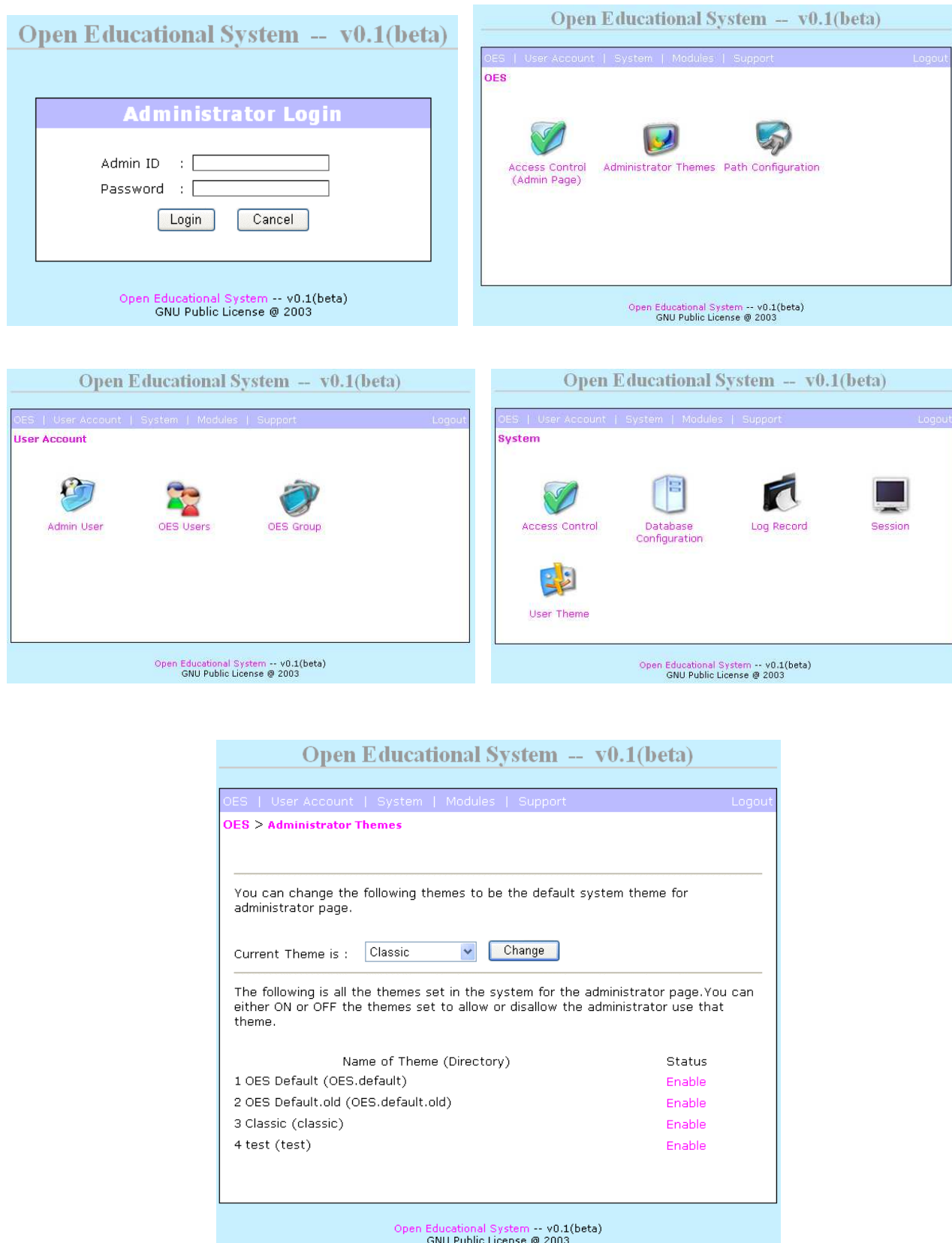


Fig 4.19 – Another set of theme

4.3 Adding Module

After downloading the modules (for admin page) from the official website, extract it and then put all the file in the

`/admin/modules/modules/`. After that login to the administrator control page again, and the new module should be found on 'Modules'.

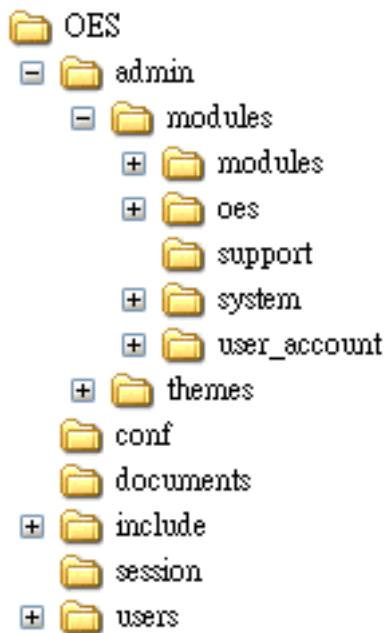


Fig 4.20 – File Structure of OES

Chapter 5

Further Improvement

This chapter will focus on the further develop and improvement of OES. Up to now, OES is the beta version, there are many functions can be add and improve on further. So, at this state, it is a good time to layout what will be the further improvement, and expected in the next generation of OES. So this chapter can give a new idea of those people who are interesting in the continue development of OES.

5.1 Conceptually module

Up to now, reader wouldn't have any problem on writing modules on OES, so at this point, there will only point out what modules are going to do in OES.

First of all, the main aim of developing the OES is major for educational purpose, so it must get some modules that can be helped for e-learning. At the foremost is the “Online Test/Exam System”. The idea of it is basically to perform an online quiz for student. There are some features expected in this module. First of all, this module must available is a real time system, and it is capable to court the time of spending on the exam. Second, the module must available to record the users IP address, such that to prevent tricky students. Third, the module can create a multi-choice and question-answers mixed question paper; it is only depend on what is the examiner want. And also, the module can display the result after the test, so that the examiner and student can see the result clearly.

Besides the “Online Test/Examination System”, a “Teaching Material Center” will be introduced into OES. This module will providing a desired space for the lecturer to upload their teaching

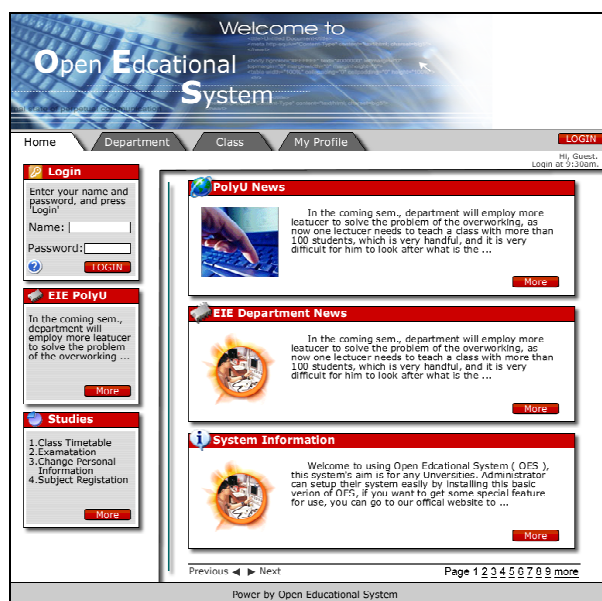
materials, such as the lecture notes, assignment, solution, video tape, etc. So that the student can download the teaching materials or view the steaming video over the net. The aim of this system can centralize all the notes from different subject into one page, so that the student can get their materials more convenience.

Getting an idea of the module just the beginning. There is still a long way to finish the module. This chapter may give some new idea to the module programmer design new modules.

5.2 Implementation on the user side

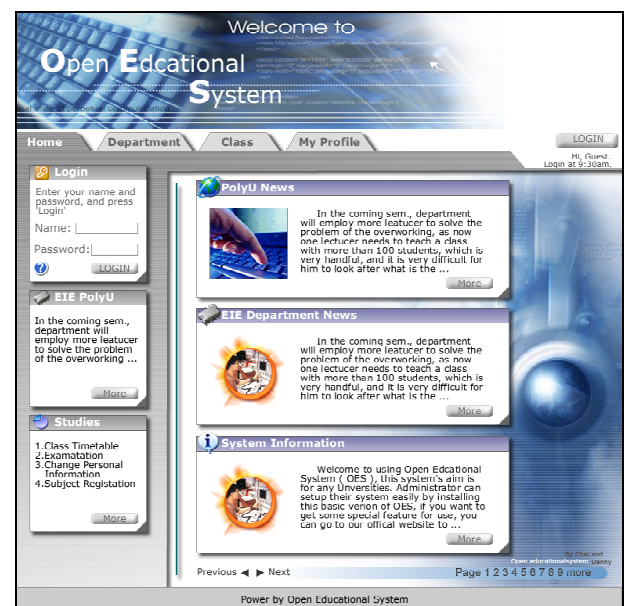
The design on user side is simulator to the design on administrator control page, which have been done. The only require to do is, create a new Login Processor for user; modifier the Modules Loader / Themes Loader and Template Loader to work on the user side. Here is the picture of the design on user side.

Fig 5.1 – User Page Interface 1



Design 1

Fig 5.2 – User Page Interface 2



Design 2

Chapter 6

Conclusions

This chapter contains the conclusions of the project writing by the students Ho Wing Chuen and Hui Ka Chai.

6.1 Conclusion

OES is a complex system, this require a clear concept and the understanding to starting with. It consumes many times to construct the structure and programming it. So it is a time consuming project, but we feel happy in the process. Because we wish this system could help the student easier to study on the Internet and we hope it can replace the Web-CT, since we feel the design of Web-CT is use not user-friendly.

The core of the system is the Kernel. We need to design it at the beginning before continue to design the modules. So it is the most important part to have a good design on it. Because a good design of Kernel can minimize the time usage on further developments.

In the system design, we was consider many of things, such as the security of the system, the extensions, the APIs, multi-platform support, multi database system support, multi-language support, control of the user right, and etc. There are many of things must be consider and we have faced many problem, so the try an error technique is very important. Lucky we can solve most the problems until now.

In this project, we need to be advance in PHP script. We need to design much CLASS for the programmer to follow, so a good programming technique is required. Also, the OES has an ability to work on most common database system, but the query syntax is little bit different to each other. So we were spend much of time to check the differences from the net. To archive the multi-database system support, we were found many CLASS can help us to implement it, but finally we were choose ADODB. Because it is easier to use compare with each other and it support many database system including the Windows platform.

Furthermore, in considering is the template changing technique. We have using some similar technique from Fast template, which is calling Template Loader in our system. The operation concept of Template Loader is simulator to Fast Template, but we have fully program the code by our self. Because we found that our system is a special one, the Fast Template is not suitable to be use in our system. Also, we have using the Flash software to create some dynamic effect, such as the Flash Draw and Plot Graph. So we have to learn the Action Script to program with Flash.

After this project, we have learned a lot of thing and more

familiar to PHP, ADODB, Action Script, HTML, JavaScript, etc. The more important is, we were learned to handle a big project, and the thinking process. We think the knowledge can help us to fulfill the further career.

Finally, although this project is not finish yet, we will keep track in the development. Because we are enjoying in the developing OES, and hope it can provide a wonderful environment to the student, allow them studies more efficiency and convenience in the Web environment.

Bibliography

This is the Bibliography. It will list all useful Links and the reference books for this project.

< 1 > **PHP** : <http://www.php.net>

PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. If you are new to PHP and want to get some idea of how it works, try the introductory tutorial. After that, check out the online manual, and the example archive sites and some of the other resources available in the links section.

< 2 > **MySQL** : <http://www.mysql.com>

MySQL is the world's most popular Open Source Database, designed for speed, power and precision in mission critical, heavy load use. MySQL AB is the company owned by the MySQL founders.

< 3 > **Linux** : <http://www.linux.org>

Linux is a free Unix-type operating system originally created by Linus Torvalds with the assistance of developers around the world. Developed under the GNU General Public License , the source code for Linux is freely available to everyone.

< 4 > **RedHat Linux** : <http://www.redhat.com>

Red Hat Linux has become the market-leading, most trusted Linux distribution. And it's why so many companies take advantage of our expertise for services like support, training, and consulting. We bring the benefits of Linux to the enterprise

< 5 > Apache Server : <http://www.apache.org>

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows NT. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

Apache has been the most popular web server on the Internet since April of 1996. The August 2002 Netcraft Web Server Survey found that 63% of the web sites on the Internet are using Apache, thus making it more widely used than all other web servers combined.

< 6 > Webmin : <http://www.webmin.com>

Webmin is a web-based interface for system administration for Unix. Using any browser that supports tables and forms (and Java for the File Manager module), you can setup user accounts, Apache, DNS, file sharing and so on.

< 7 > WebCT : <http://www.webct.com>

WebCT's e-learning systems deliver the most advanced educational technology available to support a full range of teaching and learning styles, while making the most intelligent and efficient use of an institution's intellectual and technical resources. Using WebCT allows institutions to provide more convenient and personalized learning options for students, to expand access to academic programs, and to continually improve course and degree program quality.

< 8 > **PHP-Nuke** : <http://www.phpnuke.org>

PHP-Nuke is a news automated system specially designed to be used in Intranets and Internet. The Administrator has total control of his web site, registered users, and he will have in the hand a powerful assembly of tools to maintain an active and 100% interactive web site using databases.

< 9 > **ADODB** : <http://php.weblogs.com/ADODB>

ADODB is an object oriented library written in PHP that abstracts database operations for portability. It is modelled on Microsoft's ADO, but has many improvements that make it unique (eg. pivot tables, generating HTML for paging recordsets with next and previous links, cached recordsets, HTML menu generation, etc).

< 10 > **Marocomedia** : <http://www.marcomedia.com/>

Marocomedia Flash, Fireworks, Dreamweaver, .

< 11 > **JASC** : <http://www.jasc.com/>

Paint Shop Pro .

< 12 > **Adobe** : <http://www.adobe/>

Adobe PhotoShop.

< 13 > EditPlus : <http://www.editplus.com>

EditPlus editor.

< 14 > Professional PHP4 (ISBN 1-861006-91-8)

Published by Wrox Press Ltd. , Arden House, 1102 Warwick Road, Acobs Green,
Birmingham, B27 6BH, UK.

Printed in the United States

Author (Luis Argerich, Wankyu Choi, John Coggeshall, Ken Egervari, Matrin
Geisier ...etc)

Website : <http://www.worx.com/>

Appendix

This is the Appendix. It will separate into three parts. Appendix (A) is the list of API for OES. The programmer must read this. Appendix (B) is the source code of OES. For any person who interest in the coding or he/she want to programming with OES. Appendix (C) is the Tools used to help develop the OES.

Appendix (A)

List of OES Library (API)

A.1 *lib-library.inc.php*

```
// File      : /include/library.inc.php
//
// Description : This is the general library for OES

// ----- //
// Function - chk_browser(); //
// //
// Detect the client browser, support CSS v1.0 or not //
// Return TRUE or FALSE //
// //
// *** Modifier this function if more browser support CSS v1.0 //
// ----- //

// ----- //
// Function - crypt_str( $inStr , $outSize ); //
// //
// Encrypt a string //
// $inStr = Input String to be encrypt //
// $outSize = Limit the encrypted string size //
// ----- //

// ----- //
// Function - detect_browser( $show = 1, &$type, &$ver ); //
// //
// Detect the client browser type //
// return $type and $ver //
// ----- //

// ----- //
// Function - arrayNoDuplicate ( $inArray ); //
// //
// Return a removed duplicate rows from an array //
// ----- //

// ----- //
// Function - chkSQLName ( $inSQLType ); //
// //
// Return the sql server Name //
// ----- //

// ----- //
// Function - chkDBIsODBC ( $inSQLType ); //
// //
// Return TRUE if the sql server is use ODBC connect //
// ----- //

// ----- //
// Function - chkDBSupport ( $inSQLType ); //
// //
// Return TRUE if the database is support in current PHP //
// ----- //

// ----- //
// Function - db2adodb ( $inSQLType ); //
// //
// Change the database type into ADOdb driver name. //
// ----- //
```



```

// ----- //
// Function - find_net($host,$mask); //
// //
// Function to determine network characteristics //
// $host = IP address or hostname of target host (string) //
// $mask = Subnet mask of host in dotted decimal (string) //
// returns array with //
// "cidr" => host and mask in CIDR notation //
// "network" => network address //
// "broadcast" => broadcast address //
// //
// Example: find_net("192.168.37.215","255.255.255.224") //
// returns: //
// "cidr" => 192.168.37.215/27 //
// "network" => 192.168.37.192 //
// "broadcast" => 192.168.37.223 //
// //
// ----- //

// ----- //
// Function - validEmail ( $inputEmail ); //
// //
// Return TRUE if the input string is a valid email string. //
// ----- //

// ----- //
// Function - noDoubleSpace ( $inStr ); //
// //
// Remove the double space in a string. (mainly for PostgreSQL) //
// ----- //

// ----- //
// Function - noComment ( $inPreg ); //
// //
// Remove the comment from input contents. The comment block //
// look like /*.....*/ or a single line comment //
// //
// Usage : $inPreg enter a block of string and return //
// un-commented content //
// ----- //

// ----- //
// Function - getFileVar ( $inStr ); //
// //
// Return an array of variables which found in the input TEXT //
// file. //
// ----- //

// ----- //
// Function - randomPasswd($size); //
// //
// Return a random number, the size can be limited. //
// ----- //

// ----- //
// Function - _make_seed($size); //
// //
// Private function, return the random seed for any function, //
// which require the seed. //
// ----- //

```

A.2 *lib-time.inc.php*

```
// File      : /include/lib-time.inc.php
//
// Description : This library contain the class for time function.

////////////////////////////////////
// CLASS - c_Timer();           //
// Calc. time between two actions //
//                               //
////////////////////////////////////
// Usage                       //
// $timer = new c_Timer;        //
// $timer->start();              //
// $timer->stop();               //
// echo $timer->elapsed();       //
////////////////////////////////////
```

A.3 *lib-file.inc.php*

```
// File      : /include/lib-file.inc.php
//
// Description : This library contain all the class or function for file handling.
//
// -----
// Class :: myFile
// -----
// Description : This class contain the method to access file system indirectly.
//              This is vary useful to protect the file in system and only allow
//              the authorise person to download the file.
// -----
// Variable   : $filename;           // the file will be sent
//              $error=0;             // set to 1 to enable error mode
//              $trafControl=0;       // set to 1 to enable traffic control
//              $noCache=0;           // set to 1 to disable using cache
//              $trafSpeed=10240;     // speed byte per second (byte/s)
//                                   // default is 10240 = 10KB/s
//              $_trafIntrval=100;    // default interval time is 100us
// -----
// Method      : myFile( $inFileName = '' );
// (Public)    : setFile( $inFileName = '' );
//              : getFile();
//              : loadFile();
// -----
// Examples    : (Create a myfile object)
//              $myfile = new myFile();
//
//              (other method to create myfile object)
//              $myfile = new myFile();
//              $myfile->setFile( $file );
//
//              (force browser to download file)
//              $myfile->getFile();
//
//              (force browser to display the content of file)
//              $myfile->loadFile();
//
//              (Limit the bandwidth)
//              $myFile->trafControl=1; // enable bandwidth limit
//              $myFile->trafBuffer=1024; // buffer size
//              $myFile->trafInterval=30000; // interval time
// -----
```


A.4 *lib-account.inc.php*

```
// File      : /include/lib-account.inc.php
//
// Description : This library is for administrator and normal user account mangement.

// -----
// Class :: Admin_Account
// -----
// Description : This class contain the method to access the database to check,
//               the administrator account for login process. And then can get
//               the account information from database
// -----
// Variable   : $debug=0;           // Change to 1 to enable debug mode
// (Public)   $loginID;             // login id
//            $loginPwd;            // login password
//            $admin_info;          // login success, return information
//            $error;               // Error message
//            $autoLog=1;           // '1'=Auto save LOG data, 0=Off
//
//            $readArray=array();   // All simular record in admin table
//            $selectLimit=0;        // Selecting limit (only work for find())
//            // -- 0 mean no selection limit
//            $recordNum;           // Total number of the record are founded.
//            $searchRegular='AND'; // Search regular accept ('AND','OR' only)
//            $searchMode='LIKE';   // Search mode accept ('LIKE', '=' only)
//            $orderBy='id,adm_login_id'; // Order by which column
//            $orderMode='DESC';    // DESC = Descending Order
//            // ASC = Ascending Order
//
// (Private)  $_DB_TBL='adm_users'; // Default table for admin account
//            $_DB_TYPE;            // database information
//            $_DB_ADDR;
//            $_DB_PORT;
//            $_DB_NAME;
//            $_DB_USER;
//            $_DB_PWD;
//            $_LAST_QUERY;         // The last query string
//            $_LAST_RECORD=array(); // Temp. var store the last finded record
//            $_selectOffset=0;     // Offset (only work for find())
// -----
// Method     : Admin_Account( $ID='', $PWD='' );
// (Public)   setAdmin( $ID, $PWD );
//            error();
//            getData();
//            read( $admID );
//            find( $parent, $admLoginID, $admName, $admDesc, $admEmail, $admGroup,
//            //      $createDateTime, $expireDateTime, $orderBy );
//            newAdmin( $record=array() );
//            update( $ID='', $recordInfo=array() );
//            delAccount( $ID='' );
//            chkLogin( $ID, $PWD );
//
// (Private)  _chkLogin();
//            _getRecord( $admID, $parent, $admLoginID, $admName, $admDesc,
//            //      $admEmail, $admGroup, $createDateTime, $expireDateTime,
//            //      $orderBy );
//            Record( $mode=1, $admRecord=array() );
```

```
// ----- //  
// Examples : (Check administrator login) //  
//           $admAccount = new Admin_Account(); //  
//           $admAccount->setAdmin( $ID, $PWD ); //  
//           $result      = $admAccount->chkLogin(); //  
//           // //  
//           (other method to check login) //  
//           $admAccount = new Admin_Account( $ID, $PWD ); //  
//           $result      = $admAccount->chkLogin(); //  
//           // //  
//           (other method to check login) //  
//           $admAccount = new Admin_Account; //  
//           $result      = $admAccount->chkLogin( $ID, $PWD ); //  
// ----- //
```


A.5 *lib-group.inc.php*

```
// File      : /include/lib-group.inc.php
//
// Description : This library is used for create/edit/view the group info.
//
// -----
// Class :: OESgroup
// -----
// Description : This class contain the method to access the database to get/edit/create,
//               the group for the administrator and normal user account.
// -----
// Variable    : $debug=0;           // Change to 1 to enable debug mode
// (Public)    $loginID;             // login id
//             $loginPwd;            // login password
//             $admin_info;          // login success, return information
//             $error;               // Error message
//             $autoLog=1;           // '1'=Auto save LOG data, 0=Off
//
//             $readArray=array();   // All similar record in admin table
//             $selectLimit=0;       // Selecting limit (only work for find())
//                                   // -- 0 mean no selection limit
//             $recordNum;           // Total number of the record are founded.
//             $searchRegular='AND'; // Search regular accept ('AND','OR' only)
//             $searchMode='LIKE';   // Search mode accept ('LIKE', '=' only)
//             $orderBy='id,adm_login_id'; // Order by which column
//             $orderMode='DESC';    // DESC = Descending Order
//                                   // ASC = Ascending Order
//
// (Private)   $_DB_TBL='adm_users'; // Default table for logging system
//             $_DB_TYPE;            // database information
//             $_DB_ADDR;
//             $_DB_PORT;
//             $_DB_NAME;
//             $_DB_USER;
//             $_DB_PWD;
//             $_LAST_QUERY;         // The last query string
//             $_LAST_RECORD=array(); // Temp. var store the last finded record
//             $_selectOffset=0;     // Offset (only work for find())
// -----
// Method      : Admin_Account( $ID='', $PWD='' );
// (Public)    setAdmin( $ID, $PWD );
//             error();
//             getData();
//             read( $admID );
//             find( $parent, $admLoginID, $admName, $admDesc, $admEmail, $admGroup,
//                 $createDateTime, $expireDateTime, $orderBy );
//             newGroup( $group_name='', $group_describe='', $group_enable='' );
//             update( $GID='', $group_name='', $group_describe='', $group_enable='' );
//             delGroup( $GID='' );
// (Private)   _getRecord( $admID, $parent, $admLoginID, $admName, $admDesc,
//                 $admEmail, $admGroup, $createDateTime, $expireDateTime,
//                 $orderBy );
//             _Record( $mode=1, $GID='', $group_name='', $group_describe='',
//                 $group_enable='N' );
// -----
// Examples    : (Getting all group information)
//               $myGroup = new OESGroup;
// -----
```

A.6 *lib-log.inc.php*

```
// File      : /include/lib-log.inc.php
//
// Description : This is a CLASS of Logging system.
//
// -----
// Class :: Log
// -----
// Description : This is a class of logging system. This contain a set of method to
//               access the database table 'log'.
//               The purpose of this class is used to put or read the log event into
//               or from the database. If you have any problem of using this class, or
//               don't know how this work, you can enable the DEBUG mode and this will
//               display all the debug messages on the screen (english only).
// -----
// Variable   : $debug=0;           // Change to 1 to enable debug mode
// (Public)   $SID;                 // Session ID
//           $error;                // Error message
//           $readArray;            // All similar event from log
//           $selectLimit=0;        // Selecting limit (only work for find())
//           $recordNum;            // Total number of the record are founded.
//           $orderMode='DESC';     // DESC = Descending Order
//                                   // ASC  = Ascending Order
//
// (Private)  $_selectOffset=0;     // Offset (only work for find())
//           $_DB_TBL='log';        // Default table for logging system
//           $_DB_TYPE;             // database information
//           $_DB_ADDR;
//           $_DB_PORT;
//           $_DB_NAME;
//           $_DB_USER;
//           $_DB_PWD;
//           $_FIND_CODE;           // Temp. var used for check the selection
//           $_FIND_DATETIME;      // is the the same record
//           $_FIND_USERID;        //
//           $_LAST_QUERY;         // Store the last time of query string
// -----
// Method     : Log();
// (Public)   save( $logCode='', $logMsg='' );
//           read( $logID );
//           find( $logCode, $logDateTime, $logUserID );
//           error();
// (Private)  _saveLog( $logID, $logMsg );
//           _getLog( $logID='', $logCode='', $logDateTime='', $logUserID='');
// -----
// Examples   : (Save Log)
//           $myLog      = new Log();
//           $result      = $myLog->save( $logID='', $logMsg='' );
//
//           (Find all similar log event)
//           $myLog      = new Log();
//           $result      = $myLog->find( $logCode, $logDateTime, $logUserID );
//
//           (get unique log record)
//           $myLog      = new Log();
//           $result      = $myLog->read( $logID );
// -----
```


A.7 *lib-mime.inc.php*

```
// File      : /include/lib-mime.inc.php
//
// Description : This is file define all the MIME type.

// #####
// # Function - getMIME( $inFileName );
// #
// # Enter the file name and return the MIME type.
// #
// #####

// MIME type use for Content-Type header
$mimetypes = array(
    'ez'      => 'application/andrew-inset',
    'hqx'     => 'application/mac-binhex40',
    'cpt'     => 'application/mac-compactpro',
    'doc'     => 'application/msword',
    'bin'     => 'application/octet-stream',
    'dms'     => 'application/octet-stream',
    'lha'     => 'application/octet-stream',
    'lzh'     => 'application/octet-stream',
    'exe'     => 'application/octet-stream',
    'class'   => 'application/octet-stream',
    'so'      => 'application/octet-stream',
    'dll'     => 'application/octet-stream',
    'oda'     => 'application/oda',
    'pdf'     => 'application/pdf',
    'ai'      => 'application/postscript',
    'eps'     => 'application/postscript',
    'ps'      => 'application/postscript',
    'smi'     => 'application/smil',
    'smil'    => 'application/smil',
    'mif'     => 'application/vnd.mif',
    'xls'     => 'application/vnd.ms-excel',
    'ppt'     => 'application/vnd.ms-powerpoint',
    'wbxml'   => 'application/vnd.wap.wbxml',
    'wmlc'    => 'application/vnd.wap.wmlc',
    'wmlsc'   => 'application/vnd.wap.wmlscriptc',
    'bcpio'   => 'application/x-bcpio',
    'vcd'     => 'application/x-cdlink',
    'pgn'     => 'application/x-chess-pgn',
    'cpio'    => 'application/x-cpio',
    'csh'     => 'application/x-csh',
    'dcr'     => 'application/x-director',
    'dir'     => 'application/x-director',
    'dvr'     => 'application/x-director',
    'dvi'     => 'application/x-dvi',
    'spl'     => 'application/x-futuresplash',
    'gtar'    => 'application/x-gtar',
    'hdf'     => 'application/x-hdf',
    'js'      => 'application/x-javascript',
    'skp'     => 'application/x-koan',
    'skd'     => 'application/x-koan',
    'skt'     => 'application/x-koan',
    'skm'     => 'application/x-koan',
    'latex'   => 'application/x-latex',
    'nc'      => 'application/x-netcdf',
    'cdf'     => 'application/x-netcdf',
    'sh'      => 'application/x-sh',
    'shar'    => 'application/x-shar',
    'swf'     => 'application/x-shockwave-flash',
    'sit'     => 'application/x-stuffit',
    'sv4cpio' => 'application/x-sv4cpio',
    'sv4crc'  => 'application/x-sv4crc',
    'tar'     => 'application/x-tar',
    'tcl'     => 'application/x-tcl',
    'tex'     => 'application/x-tex',
    'texinfo' => 'application/x-texinfo',
    'texi'    => 'application/x-texinfo',
    't'       => 'application/x-troff',
    'tr'      => 'application/x-troff',
    'roff'    => 'application/x-troff',
    'man'     => 'application/x-troff-man',
```



```

'me'      => 'application/x-troff-me',
'ms'      => 'application/x-troff-ms',
'ustar'   => 'application/x-ustar',
'src'     => 'application/x-wais-source',
'xhtml'   => 'application/xhtml+xml',
'xht'     => 'application/xhtml+xml',
'zip'     => 'application/zip',
'au'      => 'audio/basic',
'snd'     => 'audio/basic',
'mid'     => 'audio/midi',
'midi'    => 'audio/midi',
'kar'     => 'audio/midi',
'mpga'    => 'audio/mpeg',
'mp2'     => 'audio/mpeg',
'mp3'     => 'audio/mpeg',
'aif'     => 'audio/x-aiff',
'aiff'    => 'audio/x-aiff',
'aifc'    => 'audio/x-aiff',
'm3u'     => 'audio/x-mpegurl',
'ram'     => 'audio/x-pn-realaudio',
'rm'      => 'audio/x-pn-realaudio',
'rpm'     => 'audio/x-pn-realaudio-plugin',
'ra'      => 'audio/x-realaudio',
'wav'     => 'audio/x-wav',
'pdb'     => 'chemical/x-pdb',
'xyz'     => 'chemical/x-xyz',
'bmp'     => 'image/bmp',
'gif'     => 'image/gif',
'ief'     => 'image/ief',
'jpeg'    => 'image/jpeg',
'jpg'     => 'image/jpeg',
'jpe'     => 'image/jpeg',
'png'     => 'image/png',
'tiff'    => 'image/tiff',
'tif'     => 'image/tiff',
'djvu'    => 'image/vnd.djvu',
'djv'     => 'image/vnd.djvu',
'wbmp'    => 'image/vnd.wap.wbmp',
'ras'     => 'image/x-cmu-raster',
'pnm'     => 'image/x-portable-anymap',
'pbm'     => 'image/x-portable-bitmap',
'pgm'     => 'image/x-portable-graymap',
'ppm'     => 'image/x-portable-pixmap',
'rgb'     => 'image/x-rgb',
'xbm'     => 'image/x-xbitmap',
'xpm'     => 'image/x-xpixmap',
'xwd'     => 'image/x-xwindowdump',
'igs'     => 'model/iges',
'iges'    => 'model/iges',
'msh'     => 'model/mesh',
'mesh'    => 'model/mesh',
'silo'    => 'model/mesh',
'vrl'     => 'model/vrml',
'vrml'    => 'model/vrml',
'css'     => 'text/css',
'html'    => 'text/html',
'htm'     => 'text/html',
'asc'     => 'text/plain',
'txt'     => 'text/plain',
'rtx'     => 'text/richtext',
'rtf'     => 'text/rtf',
'sgml'    => 'text/sgml',
'sgm'     => 'text/sgml',
'tsv'     => 'text/tab-separated-values',
'wml'     => 'text/vnd.wap.wml',
'wmls'    => 'text/vnd.wap.wmlscript',
'etx'     => 'text/x-setext',
'xsl'     => 'text/xml',
'xml'     => 'text/xml',
'mpeg'    => 'video/mpeg',
'mpg'     => 'video/mpeg',
'mpe'     => 'video/mpeg',
'qt'      => 'video/quicktime',
'mov'     => 'video/quicktime',
'mxu'     => 'video/vnd.mpegurl',
'avi'     => 'video/x-msvideo',
'movie'   => 'video/x-sgi-movie',
'ice'     => 'x-conference/x-cooltalk',

```

}:

A.8 *lib-mydb.inc.php*

```
// File      : /include/lib-mydb.inc.php
//
// Description : This class contain the method is used to access the database
//
// -----
// Class :: myDB
// -----
// Description : This class contain all the method is used to help the programmer access
//               the database without mine the changes of database system. The variable,
//               function and usage is listed on below.
// -----
// Variable    : $debug=0;           // Change to 1 to enable debug mode
// (Public)    $error;               // Error message
//             $readArray=array();   // All simular record in admin table
//             $selectLimit=0;       // Selecting limit (only work for find())
//                                     // -- 0 mean no selection limit
//             $recordNum;           // Total number of the record are founded.
//             $searchRegular='AND'; // Search regular accept ('AND','OR only)
//             $searchMode='LIKE';   // Search mode accept ('LIKE', '=' only)
//             $orderBy='id,adm_login_id'; // Order by which column
//             $orderMode='DESC';    // DESC = Descending Order
//                                     // ASC = Ascending Order
//             $keyField='id';       // The name of the key field in a table
//
// (Private)   $_DB_TBL='';         // Default table used
//             $_DB_TYPE;           // database information
//             $_DB_ADDR;
//             $_DB_PORT;
//             $_DB_NAME;
//             $_DB_USER;
//             $_DB_PWD;
//             $_LAST_QUERY;        // The last query string
//             $_LAST_RECORD=array(); // Temp. var store the last finded record
//             $_selectOffset=0;    // Offset (only work for find())
// -----
// Method      : changeTable( $table='' );
// (Public)    error();
//             getRecord( $inData='' );
//             update( $inData=array() );
//             insert( $inData=array() );
//             delRecord( $inData='' );
//             query( $query );
//
// (Private)   _Query( $mode=0, $inData='' );
// -----
// Examples    : (Select all record in a table, without limit the result on each time)
//               $mydb = new myDB('TABLE');
//               $mydb->selectLimit=0;
//               $result = $mydb->getRecord();
```



```

//      (Select the record with the conditions, the following command will the //
//      same of this query: [ SELECT * FROM myTABLE WHERE name LIKE '%Test%' //
//      AND favor LIKE '%apple%' ORDER BY id, name ASC ] ) //
//      $mydb = new myDB('myTABLE'); //
//      $mydb->searchRegular = 'AND'; //
//      $mydb->searchMode = 'LIKE'; //
//      $mydb->orderMode = 'ASC'; //
//      $mydb->orderBy = 'id, name'; <= Assume the table have the //
//      $array['name'] = 'Test'; field (id, name) //
//      $array['favor'] = 'apple'; //
//      $result = $mydb->getRecord ( $array ); //
//
//      (Update a ROW in the table, the following program will provide the same //
//      as the query : //
//      [ UPDATE myTABLE SET name='Test', favot='apple' WHERE id=10 ] //
//      Remember you must provide the name of KEY FIELD inorder to update the //
//      unique row in the table.) //
//      $mydb = new myDB('myTABLE'); //
//      $mydb->keyField = 'id'; //
//      $array['id'] = '10'; //
//      $array['name'] = 'Test'; //
//      $array['favor'] = 'apple'; //
//      $mydb->update ( $array ); //
//
//      (Insert a ROW into the table, the operation is simulator to update. But //
//      you must remind that this insert will automatically increase the KEY //
//      field by 1. So you must confirm that the table must have a unique field//
//      and the propriety of that field is a numeric value. The query will like//
//      [INSERT INTO myTABLE (id, name, favor) VALUES (??, 'Test', 'apple')] //
//      $mydb = new myDB('myTABLE'); //
//      $mydb->keyField = 'id'; //
//      $array['id'] = '10'; <= This field is the key field, //
//      $array['name'] = 'Test'; so it will be ignore and the //
//      $array['favor'] = 'apple'; number will be assign by program //
//      $mydb->insert ( $array ); //
//
//      (Delete a ROW in a table. The ROW will be delete is 'id'=25.) //
//      $mydb = new myDB('myTABLE'); //
//      $mydb->keyField = 'id'; //
//      $mydb->delRecord( 25 ); //
//
//      (Execute a user defined SQL query command) //
//      $queryString='CREATE TABLE XYZ (id INT,name CHAR(20),favor CHAR(10))' //
//      $mydb->query( $queryString ); //
//      ----- //

```

A.9 *lib-template-mod.inc.php*

```
// File      : /include/lib-template-mod.inc.php
//
// Description : This library contain all the class or function for themes.
//
// -----
// Class :: myModTmp
// -----
// Description : This class contain the method provided for the modules developer more
//              easier to create their template for the modules.
// -----
// Variable   : $debug=0; // Set to 1 to enable debug mode
// (Public)   : $template=''; // the modules's template file
//            : $lang=''; // the language name (e.g 'en')
//            : $selfPath=0; // 1=loading the template from any where
//                      // (within the OES)
//            : // 0=only load the template for modules
//            : $isAdmin=0; // 1=loading the admin modules template
//                      // 0=loading the user modules template
// (Private)  : $_result=''; // the final result
//            : $_langPath=''; // the language file path
//            : $_category=''; // which category does this modules in
//            : $_tempResult=array(); // the temporary result
//                      // (array of repeat ROWs)
//            : $_modules=''; // default to find the current
//                      // modules is using
// -----
// Method     : myModTmp( $template='', $lang = 'en' );
// (Public)   : getResult();
//            : getLangVAR( $langStr );
//            : getTemplate();
//            : assign( $pattern='', $replace='' );
//            : repeatRow( $rowname );
//            : removeRow( $rowName );
// -----
// Examples   : (Create a myModTmp object, and get the template for 'abc.template'.
//              which is the administrator modules template and the language is 'EN'
//              (i.e. English), the base path of the template is :
//              "$CONF_ADMIN_MODULES_PATH.'/'. $this->_category.'/'. $this->_modules"
//              and then get the final result to the screen.)
//              $myMod = new myModTmp( 'abc.template', 'en' );
//              $myMod->debug=0;
//              $myMod->isAdmin=true;
//              $myMod->getTemplate();
//              echo $myMod->getResult();
//
//              (Dynamic Assign a value to the template, assume the template have a
//              dynamic variable {DYNAMIC_VAR})
//              $myMod = new myModTmp( 'abc.template', 'en' );
//              $myMod->debug=0;
//              $myMod->getTemplate();
//              $myMod->assign ( 'DYNAMIC_VAR', 'My Dinamic assignment.!!' );
//              echo $myMod->getResult();
//
//              (Repeat a row with a dynamic assignment, assume the ROW name is 'ABC')
//              $myMod->getTemplate();
//              $myMod->assign ( 'DYNAMIC_VAR', 'My Dinamic assignment 1.!!' );
//              $myMod->repeatRow( 'ABC' );
//              $myMod->assign ( 'DYNAMIC_VAR', 'My Dinamic assignment 2.!!' );
//              echo $myMod->getResult();
//
//              (Doubly call a repeat row)
//              for ( $i=0; $i<10; $i++ ) {
//                  for ( $j=0; $j<5; $j++ ) {
//                      $myMod->assign ( 'DYNAMIC_VAR', $i );
//                      $myMod->repeatRow( 'inRow' );
//                  }
//                  $myMod->removeRow( 'inRow' );
//                  $myMod->repeatRow( 'BIG_ROW' );
//              }
//              echo $myMod->getResult();
// -----
```


A.10 *lib-themes.inc.php*

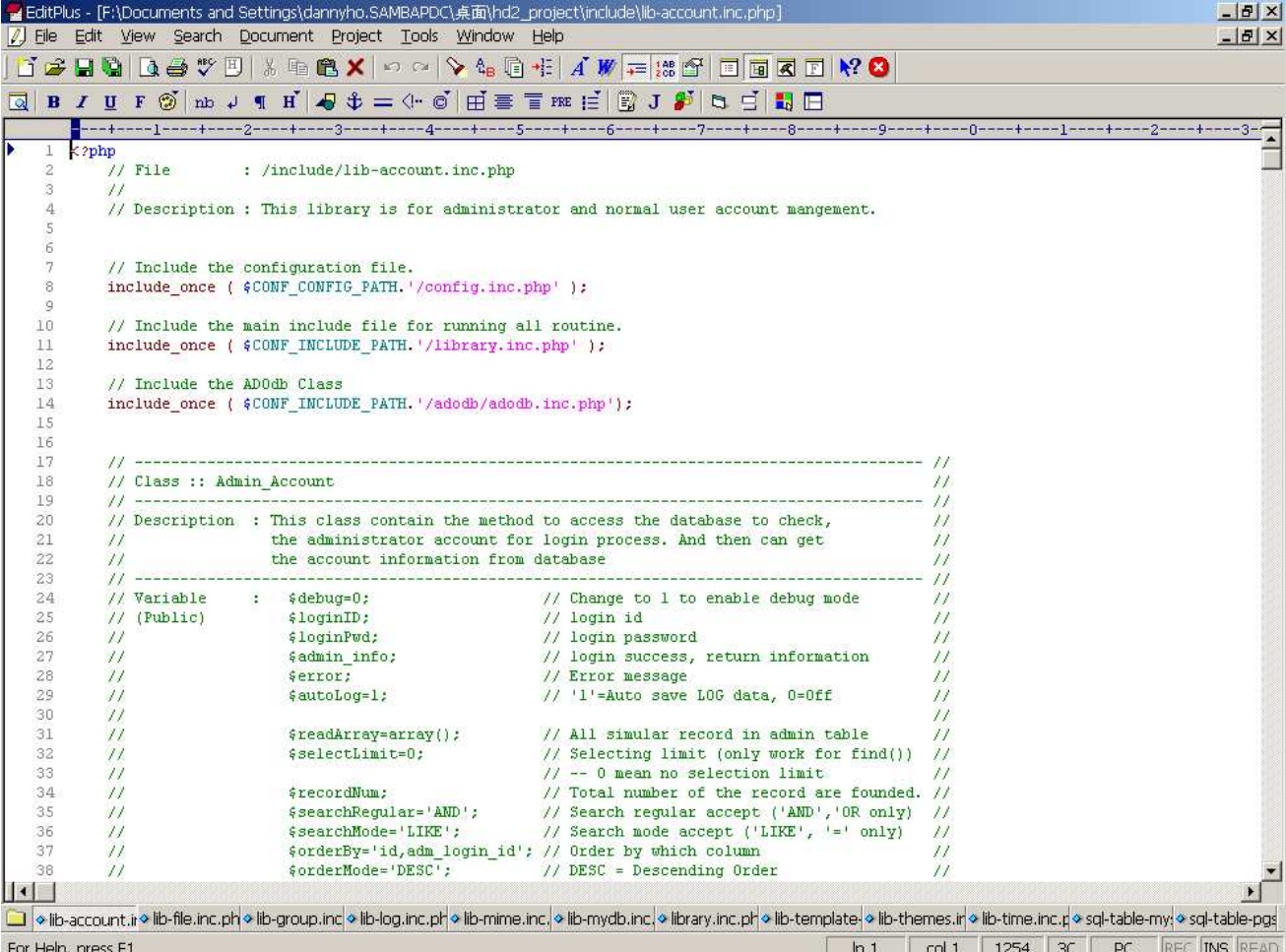
```
// File      : /include/lib-themes.inc.php
//
// Description : This library contain all the class or function for themes.
//
// -----
// Class :: myTheme
// -----
// Description : This class contain the method to access themes and template files.
//              This is vary important classes in OES. Because all the HTML output
//              will related on this class. Without the help of this class, the OES
//              can't display anything on the client browser.
// -----
// Variable   : $debug=0;           // Set to 1 to enable debug mode
//              $themeName='OES.default'; // The current themes is using
//              $themeLang='en';       // The default output language
//              $speedLimit=0;         // 1=limit the download speed
//              $isAdmin=0;            // 1=loading the admin themes
//                                     // 0=loading the user themes
//
//              $_result='';          // the final result
//              $_tempResult=array();  // the temporary result
//                                     // (array of repeat ROWs)
//              $_langPath='';         // the language file path
// -----
// Method      : myFile( $inFileName = '' );
// (Public)    : setFile( $inFileName = '' );
//              getFile();
//              loadFile();
//              getHeader();
//
// (Private)   : _checkTheme();
// -----
// Examples    : (Create a myfile object)
//              $myfile      = new myFile();
// -----
```

Appendix (B)

Helpful Tools

B.1 EditPlus v2.11

(<http://www.editplus.com/>)



```

1  k?php
2  // File      : /include/lib-account.inc.php
3  //
4  // Description : This library is for administrator and normal user account mangement.
5
6
7  // Include the configuration file.
8  include_once ( $CONF_CONFIG_PATH.'/config.inc.php' );
9
10 // Include the main include file for running all routine.
11 include_once ( $CONF_INCLUDE_PATH.'/library.inc.php' );
12
13 // Include the ADODB Class
14 include_once ( $CONF_INCLUDE_PATH.'/adodb/adodb.inc.php' );
15
16
17 // -----
18 // Class :: Admin_Account
19 // -----
20 // Description : This class contain the method to access the database to check,
21 //              the administrator account for login process. And then can get
22 //              the account information from database
23 // -----
24 // Variable   : $debug=0;           // Change to 1 to enable debug mode
25 // (Public)   $loginID;             // login id
26 //           $loginPwd;             // login password
27 //           $admin_info;           // login success, return information
28 //           $error;                // Error message
29 //           $autoLog=1;            // '1'=Auto save LOG data, 0=Off
30 //
31 //           $readArray=array();    // All simular record in admin table
32 //           $selectLimit=0;        // Selecting limit (only work for find())
33 //           // -- 0 mean no selection limit
34 //           $recordNum;            // Total number of the record are founded.
35 //           $searchRegular='AND';  // Search regular accept ('AND','OR only)
36 //           $searchMode='LIKE';    // Search mode accept ('LIKE', '=' only)
37 //           $orderBy='id,adm_login_id'; // Order by which column
38 //           $orderMode='DESC';     // DESC = Descending Order
  
```

lib-account.inc.php lib-file.inc.php lib-group.inc.php lib-log.inc.php lib-mime.inc.php lib-mydb.inc.php library.inc.php lib-template lib-themes.ir lib-time.inc.php sql-table-my sql-table-pgs

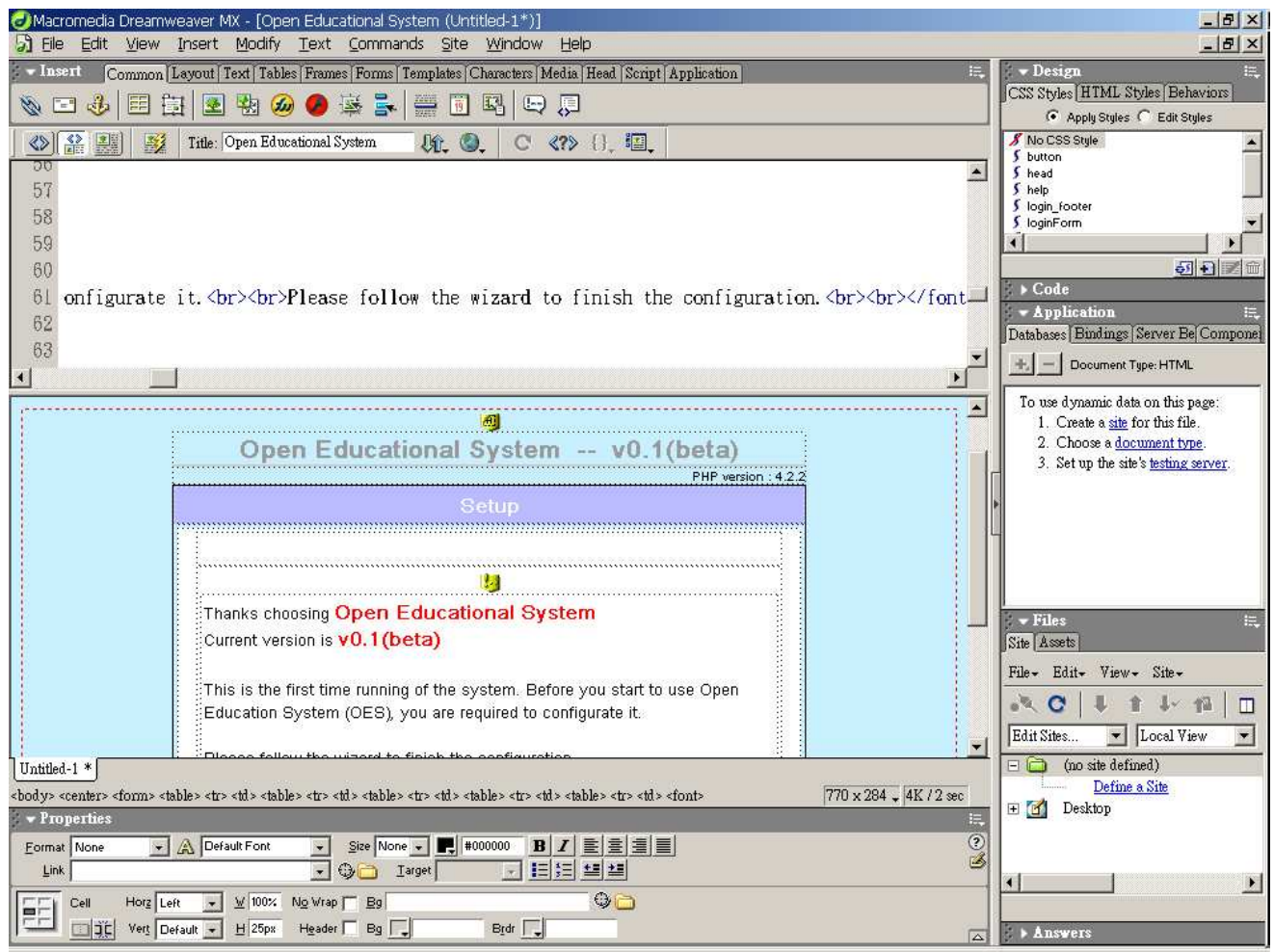
For Help, press F1

Ln 1 col 1 1254 3C PC REC INS READ

App-B Fig 1 – EditPlus v2.11

B.2 Marcomedia Dreamweaver MX

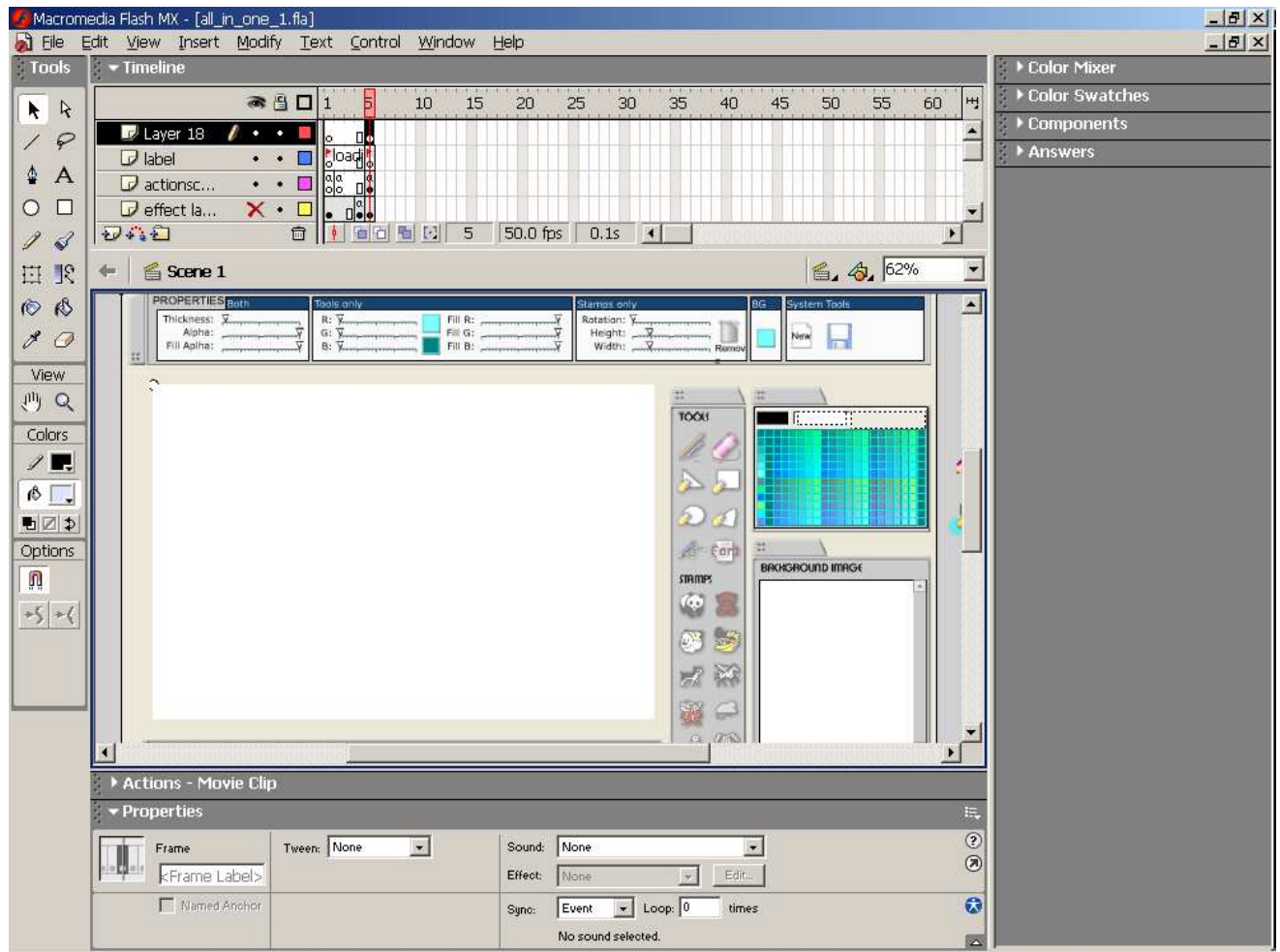
(<http://www.macromedia.com/>)



App-B Fig 2 – Dreamweaver MX

B.3 Marcomedia Flash MX

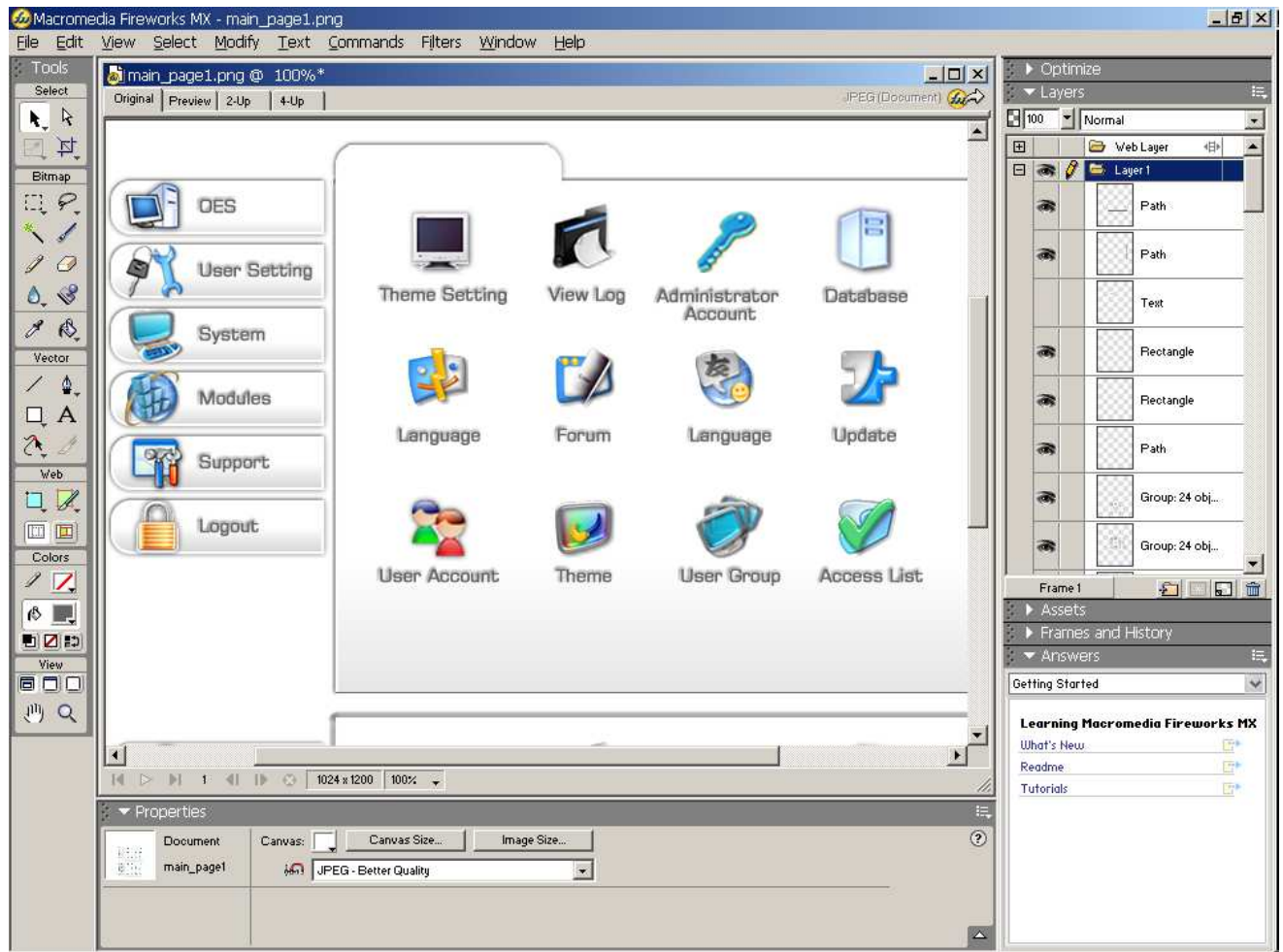
(<http://www.macromedia.com/>)



App-B Fig 3 – Flash MX

B.4 Marcomedia Firework MX

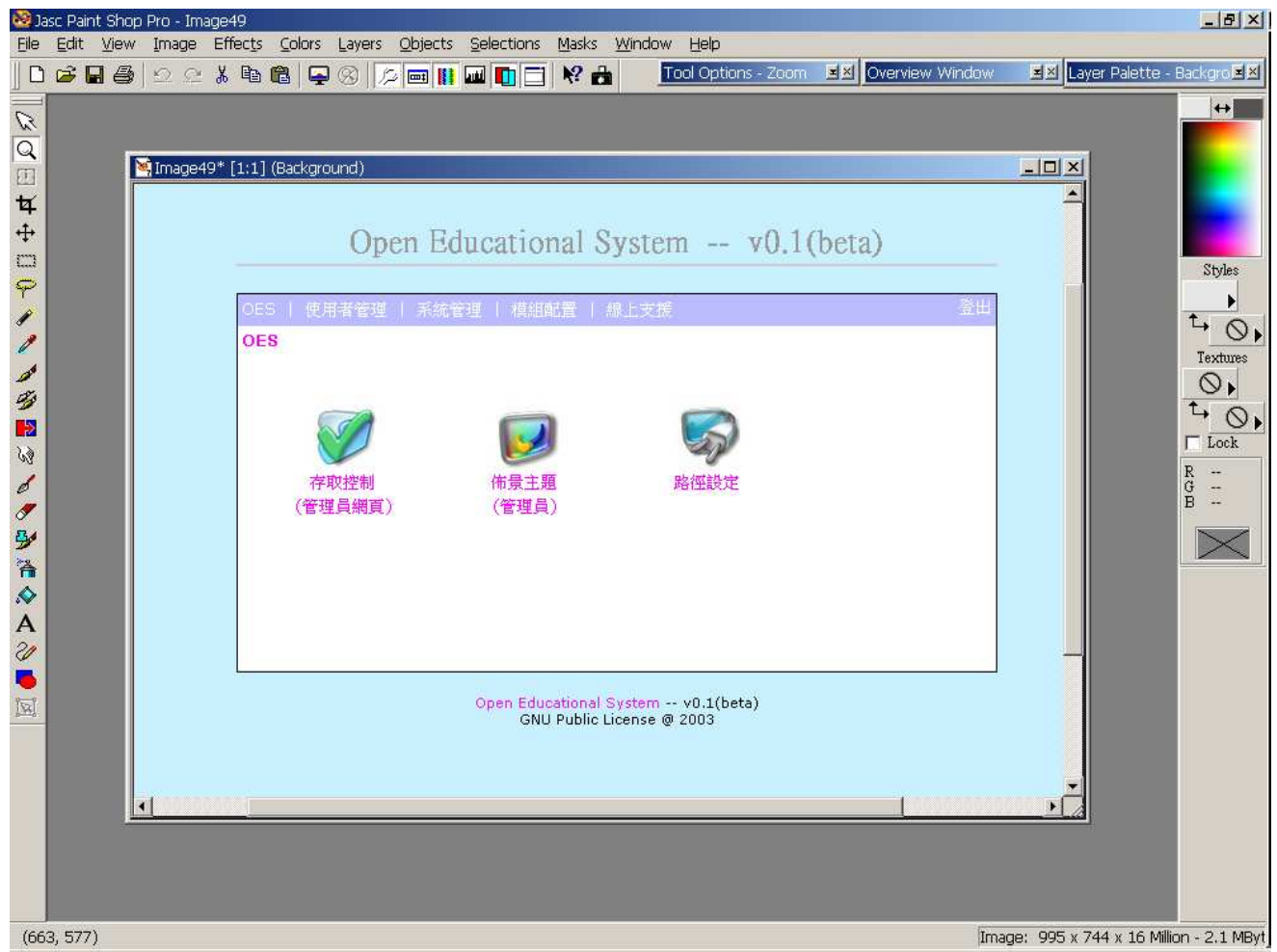
(<http://www.macromedia.com/>)



App-B Fig 4 – Fireworks MX

B.5 JASC Paint Shop Pro 7.0

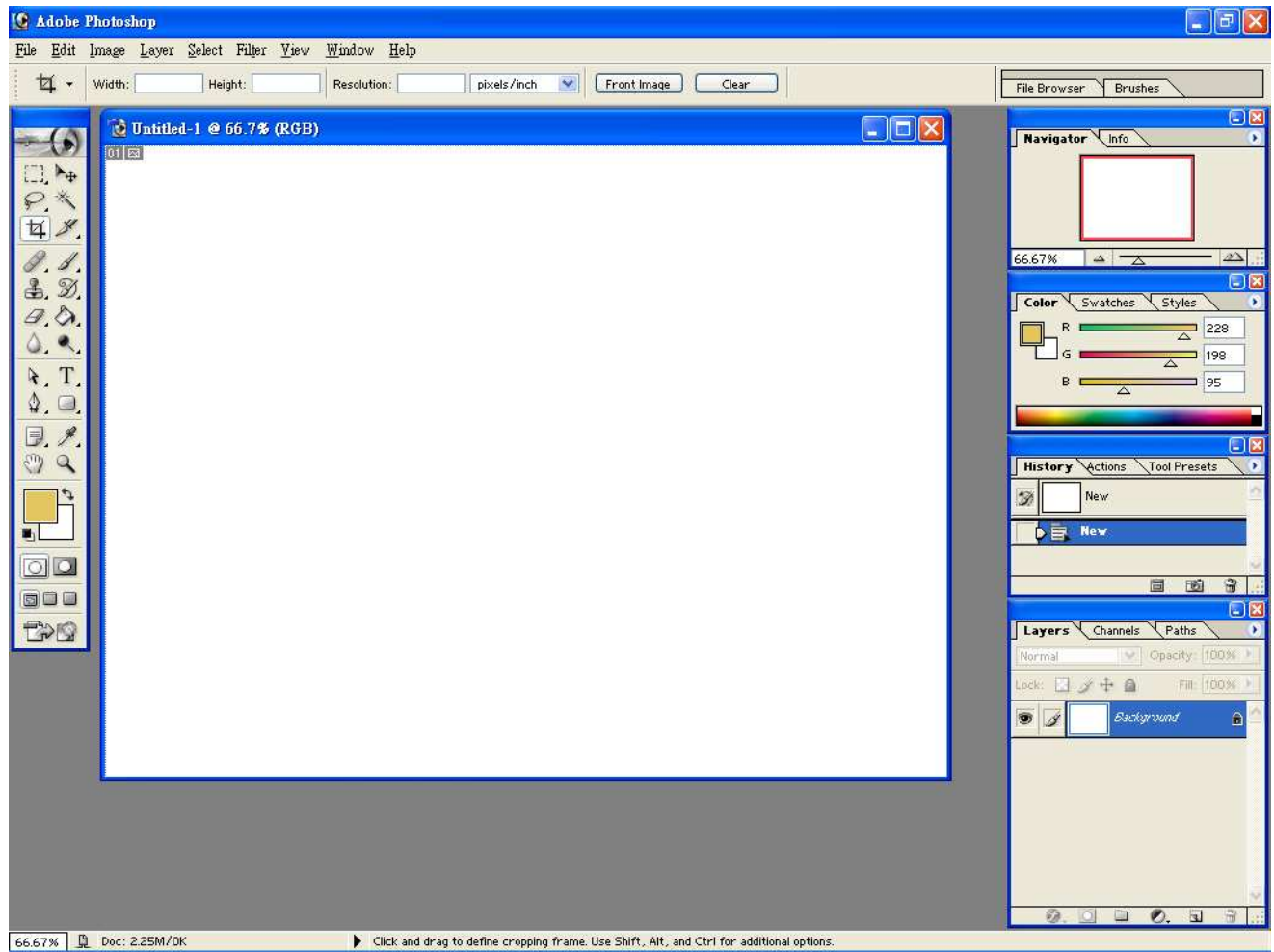
(<http://www.jasc.com/>)



App-B Fig 5 – Paint Shop Pro

B.6 Adobe PhotoShop 7.0

(<http://www.adobe.com/products/photoshop/>)



App-B Fig 6 – Photoshop 7.0

Appendix (C)

List of the Figures and Tables

Fig 2.1 – Scope of OES	(P. 18)
Fig 2.2 – Access to OES	(P. 20)
Fig 2.3 – Operation of OES	(P. 22)
Table 2.1 – Structure of URL link	(P. 27)
Fig 2.4 – Example of URL link	(P. 28)
Fig 2.5 – File Loader	(P. 31)
Table 2.2 – Structure of file loader	(P. 31)
Fig 2.6 – Download a file (GET mode)	(P. 32)
Fig 2.7 – Disable Cache	(P. 33)
Fig 2.8 – Display the file content in browser	(P. 35)
Fig 2.9 – Force browser to download file	(P. 35)
Fig 2.10 – Without Traffic	(P. 36)
Fig 2.11 – With Traffic	(P. 36)
Fig 2.12 – Flow chart of Speed Limit	(P. 38)
Fig 2.13 – Code of Traffic Limit	(P. 39)
Table 2.3 – Directory Structure of OES	(P. 40)
Fig 2.14 – Setup Wizard	(P. 42)
Fig 2.15 – Code to check Configuration file	(P. 43)
Table 2.4 – List of library file	(P. 44)
Fig 2.16 – DB table : Administrator Account (MySQL)	(P. 48)
Fig 2.17 – DB table : Group Account (MySQL)	(P. 48)

Fig 2.18 – DB table : Log	(P. 49)
Fig 2.19 – DB table : LOG (PostgreSQL)	(P. 49)
Fig 2.20 – Generate a new ID	(P. 50)
Fig 2.21 – Example on Update ROW on Table	(P. 50)
Fig 2.22 – Example on Insert ROW into Table	(P. 51)
Fig 2.23 – Example on Delete ROW from Table	(P. 51)
Fig 2.24 – Example of Configuration file	(P. 52)
Fig 2.25 – Default Theme	(P. 53)
Fig 2.26 – Usage of Theme	(P. 53)
Fig 2.27 – Automatic select language	(P. 54)
Fig 2.28 – Load the library	(P. 55)
Fig 2.29 – Login flow chart	(P. 58)
Fig 2.30 – Logout flow chart	(P. 58)
Fig 2.31 – Prepare the SESSION Environment	(P. 59)
Fig 2.32 – Create SESSION (Login)	(P. 59)
Fig 2.33 – Destroy the SESSION (Logout)	(P. 60)
Fig 2.34 – Start SESSION	(P. 61)
Fig 2.35 – Register Session	(P. 62)
Fig 2.36 – Module Configuration file 'module.php'	(P. 65)
Table 2.5 – Define Module Configuration File	(P. 65)
Fig 2.37 – Optional variable in module configuration file	(P. 66)
Fig 2.38 – Modules shows on client browser	(P. 67)
Fig 2.39 – Check the modules is available (Enable)	(P. 68)
Fig 2.40 – Display the Modules Icon on the screen	(P. 69)
Fig 2.41 – Include a Modules (Execute)	(P. 70)
Fig 2.42 – Full code to Execute a module	(P. 71)
Fig 2.43 – Language file (English)	(P. 72)
Fig 2.44 – Language file (Chinese – BIG5)	(P. 73)
Fig 2.45 – Display on English	(P. 74)
Fig 2.46 – Display on Chinese (BIG5)	(P. 74)
Fig 2.47 – Template file example	(P. 75)
Fig 2.48 – Replace Static Variable (Template Loader)	(P. 77)
Fig 2.49 – Step 1 : Original HTML Code 'mod_group_template'	(P. 78)
Fig 2.50 – Step 2: Define the template in configuration file 'module.php'	(P. 78)
Fig 2.51 – Step 3: Program codes, dynamic assign the variable	(P. 78)
Fig 2.52 – Step 4 : HTML Result	(P. 79)
Fig 2.53 – Code to do dynamic assignment the value (Template Loader) ...	(P. 79)
Fig 2.54 – Repeat a part of HTML codes	(P. 80)

Fig 2.55 – Step 1: Original HTML code ‘mod_group_template’	(P. 81)
Fig 2.56 – Step 2: Program code to Dynamic Assign and Dynamic Repeat ROWs (Segment)	(P. 82)
Fig 2.57 – Adding a Initialize BLOCK to the End	(P. 82)
Fig 3.1. – File structure of flash broad	(P. 84)
Fig 3.2 – Interface of the Flash Broad	(P. 85)
Fig 3.3 – Architecture of the flash broad before encode	(P. 86)
Fig 3.4 – Architecture of the flash broad after encode	(P. 87)
Fig 3.5 – Define mouse up/down function	(P. 88)
Fig 3.6 – Set the region of drawing plane	(P. 88)
Fig 3.7 – The structure of the draw tools	(P. 88)
Fig 3.8 – The rectangle movie clip code	(P. 89)
Fig 3.9 – The rectangle button code	(P. 90)
Fig 3.10 – Simple flow chart of rectangle draw tool	(P. 91)
Fig 3.11 – Structure of encoded data	(P. 91)
Table 3.1 – list of the parameter	(P. 92)
Fig 3.12 – Decoding code	(P. 92)
Fig 3.13 – Interface of plot graph module	(P. 94)
Fig 3.14 – Arch texture of plot graph	(P. 95)
Fig 3.15 – Define the math function	(P. 96)
Fig 3.16 – Define of create object function	(P. 96)
Fig 3.17 – Code of calculate the sum	(P. 97)
Fig 3.18 – Code of calculate the math function	(P. 97)
Fig 3.19 – Interface of the polar graph	(P. 99)
Fig 4.1 – Installation of OES	(P. 105)
Fig 4.2 – Choose language	(P. 106)
Fig 4.3 – License of OES	(P. 106)
Fig 4.4 – Choose Database	(P. 107)
Fig 4.5 – Enter Database information	(P. 108)
Fig 4.6 – Database create success	(P. 109)
Fig 4.7 – Error Screen of creating database	(P. 110)
Fig 4.8 –Display the system information	(P. 111)
Fig 4.9 –Create admin account	(P. 112)
Fig 4.10 –Confirmation the information	(P. 113)
Fig 4.11 – Setup complete	(P. 114)
Fig 4.12 – Interface of the admin’s login page	(P. 115)
Fig 4.13 – Interface of the admin’s page	(P. 116)
Fig 4.14 – Choosing the Current Theme	(P. 117)

Fig 4.15 – Edit/View the configuration information	(P. 118)
Fig 4.16 – Interface of user account page	(P. 119)
Fig 4.17 – Create admin account	(P. 120)
Fig 4.18 – View/ Edit account	(P. 121)
Fig 4.19 – Another set of theme	(P. 122)
Fig 4.20 – File Structure of OES	(P. 123)
Fig 5.1 – User Page Interface 1	(P. 127)
Fig 5.2 – User Page Interface 2	(P. 127)
App-B Fig 1 – EditPlus v2.11	(P. 154)
App-B Fig 2 – Dreamweaver MX	(P. 155)
App-B Fig 3 – Flash MX	(P. 156)
App-B Fig 4 – Fireworks MX	(P. 157)
App-B Fig 5 – Paint Shop Pro	(P. 158)
App-B Fig 6 – Photoshop 7.0	(P. 159)

Appendix (D)

Source Code