Executive Information System (EIS) is a Web based application for analyzing information about all College students.
Contents

1. Introduction ............................................................................................................................. 2
   1.1 Purpose ............................................................................................................................. 2
   1.2 Scope ................................................................................................................................ 2
   1.3 Definitions, Acronyms and Abbreviations ....................................................................... 2
2. System Requirements .............................................................................................................. 3
   2.1 Software Requirements .................................................................................................... 4
   2.2 Hardware Requirements ................................................................................................... 6
   2.3 Support Requirements ...................................................................................................... 7
3. Software Installation guidelines .............................................................................................. 8
   3.1 Installing Apache, MySQL, and PHP on Linux ............................................................... 8
   3.2 Get the source balls .......................................................................................................... 8
   3.3 Installation ........................................................................................................................ 8
   3.4 Compiling Apache ............................................................................................................ 9
   3.5 Compiling MySQL ......................................................................................................... 10
   3.6 Compiling PHP .............................................................................................................. 11
   3.7 Testing ............................................................................................................................ 12
   3.8 Starting Apache and MySQL Automatically ................................................................. 13
   3.9 Troubleshooting ............................................................................................................. 14
4. PHP Code in Detail................................................................................................................ 15
   4.1 Connecting to database and Executing Queries ............................................................. 17
   4.2 Migrating/Updating Code to work on a Oracle database ............................................. 18
5. Datatel Code in detail ............................................................................................................ 21
6. Useful URLs and References ............................................................................................... 22
1. Introduction

This document provides a high level overview and explains the whole process of using Executive Information Systems, EIS. It explains how a user can retrieve meaningful information by using this application.

The document provides a high-level description of the goals of the architecture, the use cases supported by the system and architectural styles and components that have been selected to best achieve the use cases. This framework then allows for the development of the design criteria and documents that define the technical and domain standards in detail.

1.1 Purpose

The System Architecture Document (SAD) provides a comprehensive software architectural overview of EIS. It also lists the hardware and software requirements to install and run the application. Architectural overview also includes database architecture and query logic.

1.2 Scope

The scope of this document is to depict the architecture of EIS web application created by Hartnell College TitleV team.

This document describes the aspects of EIS design that are considered to be significant; that is, those elements and behaviors that are most fundamental for guiding the construction of EIS and for understanding this project as a whole. Stakeholders who require a technical understanding of EIS are encouraged to start by reading this document, then reviewing the EIS UML model, and then by reviewing the source code.

1.3 Definitions, Acronyms and Abbreviations

- EIS – Executive Information System
- LAMP – Linux Apache MySQL PHP
- MySQL – Relational Database Management System (RDBMS)
- PHP – Hypertext Preprocessor scripting language
- HTTP – Hypertext Transfer Protocol
- WWW – World Wide Web
- Apache – Web Server
- ETL – Extract, Transform, Load
2. System Requirements

The following sections list the hardware, software and support requirements to install, run and maintain EIS application. This section will include software architecture requirements, hardware requirements, scheduled support jobs that must run, various batch processes required for using the application.

This document assumes the regular transactional database used by the college is Datatel. MySQL is used as a data warehouse for reporting and analysis.

High level data flow diagram:

In Figure.1 each component has associated software, hardware and support requirement, which will be found in the software requirements, hardware requirements and support requirements sections respectively.

Each section has more detail on data flow at each level, underlying code, interaction between each component and the internal components.
Some of the installations procedures are also covered in this document, which include Apache installation, MySQL installation and PHP installation, bundling them to work as a single component.

### 2.1 Software Requirements

All the EIS tools run on LAMP (Linux 2.6, Apache 2.2, MySQL 5.x, and PHP5.x) server. EIS is hosted on Apache Web Server and connecting to the College’s MySQL Database. The web server is listening on the web standard port 80 and it will accept all requests from the clients. All the communication with the client has to comply with public HTTP communication protocol standards.

EIS can be used as a web application hosted on College’s intranet or a stand-alone application ported to a user’s desktop. The support requirements to use EIS as a stand-alone application are also listed in this document under support requirements section.

Data flow diagram:

![Data flow diagram](Figure.2)
All EIS users can use any generic web browser to access the tools on the intranet. Each tool will be discussed in detail in following sections. The data flow is initiated from the user by opening the application via intranet or as a desktop application. This triggers the PHP code to execute, which in turn gets data from the MySQL database, apply business rules and transform the data as required. This data is presented back to the user in most meaningful way.

![Server Side Architecture](Server Side Architecture.png)

![Client Side Architecture](Client Side Architecture.png)

Figure 3

Figure 3 depicts the system architecture when EIS is used as a web application. As we are using http and not https as our protocol, the application must be hosted on a server on the college’s intranet, to ensure secured data access.

When used as a stand-alone application, the laptop/desktop hosting EIS must have LAMP installed on it. As the MySQL database is updated only once a semester after all the students are graded, every semester the laptop/desktop must be updated with the latest data.

We do not encounter this situation when using EIS as a web application because the database is updated on the server regularly by running various batch processes and scheduled tasks.

Various batch processes and scheduled tasks required to update the database are discussed later in this document.
2.2 Hardware Requirements

As mentioned in the previous section EIS runs on LAMP server.
2.3 Support Requirements

All the tables in MySQL must be updated regularly with data from the transactional database of the college, here it is Datatel. Various scheduled tasks and batch programs are executed as a part of support the system.

MySQL database must be updated every semester after the students are graded for that semester. A cron job is scheduled to update all the tables in MySQL from the transactional database i.e. Datatel.

MySQL has some additional reporting tables that must be populated after the database is updated every semester. There are separate set of cron jobs that run on the server to update the reporting tables.

When running EIS as a stand-alone application the laptop/desktop must be installed with LAMP server/WAMP server software. Database must be updated every semester manually.

Installation of LAMP server is discussed later in this document. WampServer is a windows based web development environment. WAMP server installation has been clearly listed at www.wampserver.com

More details on Datatel, ETL processes are covered under Datatel Extract Transform Load chapter.
3. Software Installation guidelines

3.1 Installing Apache, MySQL, and PHP on Linux

This tutorial is designed to guide you through the initial steps of setting up Apache, MySQL, and PHP on Linux. The Linux distribution being utilized for this tutorial is OpenSuse; however the steps should be very similar across most distributions. This tutorial makes the assumption that you have the required development tools loaded for compiling programs from source; these tools are beyond the scope of this document and will not be covered here. Also, it assumes you can use the vi text editor for basic editing tasks.

Apache, MySQL, and PHP have become one of the most utilized combinations for developing content driven websites. They are robust, flexible, provide a decent level of security, and they are available for many different platforms.

3.2 Get the source balls

The first thing you need to do is obtain the source balls for each package, we will be compiling each package from scratch here, and, while there are also binary packages available for some distributions, I find your end results are usually better when building each package for your machine. Make sure you get the source files.

Here are the links and the package versions available at the time this tutorial was written

Apache:
URL: http://httpd.apache.org/download.cgi

MySQL:
URL: http://www.mysql.com/downloads/mysql-4.0.html

PHP:
URL: http://www.php.net/downloads.php

3.3 Installation

The first thing we need to do is extract the source balls so we can work with the files included in them. Beginning now we will be working as root, so open a terminal window, change to the directory in which you saved your downloaded files and become root by issuing the su command, enter the root password and you should be good to go.

To extract the source balls type the following commands:

# tar -zxf httpd-2.2.10.tar.gz (enter)
# tar -zxf mysql-5.0.3.tar.gz (enter)
# tar -zxf php-5.2.8.tar.gz (enter)
The commands above will extract the source balls into their own separate directories. Now let’s move on to compiling the source into usable programs. We'll start with Apache.

### 3.4 Compiling Apache

Change into the directory created when you untarred the source ball as follows:

```
#cd httpd-2.2.10 (enter)
```

Follow this command by typing;

```
#./configure --prefix=/usr/local/apache2 --enable-mods-shared=most (enter)
```

This tells Apache to install in the /usr/local/apache2 directory, and to build most of the available loadable modules. There are a ton of options with Apache, but these should work for the most part. Once the configure is done and the system returns the prompt to you, issue the following command;

```
#make
```

This will take a few minutes, once the prompt comes back again issue the following command;

```
#make install
```

Wait for a few minutes and voila !, Apache is installed with the exception of a few minor changes we still need to make. They are as follows..

Issue the following command:

```
#vi /usr/local/apache2/conf/httpd.conf
```

Check the configuration file to make sure the following line is present in the file at the bottom of the LoadModule list, if it is not there add it.

- LoadModule php4_module modules/libphp4.so

Find the DirectoryIndex line and edit it so it looks like the following:

- DirectoryIndex index.html index.html.var index.php

Find the AddType application section and add the following line:

- AddType application/x-httpd-php .php

That’s it, save the file and we are done with Apache. Now, on to MySQL !
## 3.5 Compiling MySQL

If you are using rpm installation then also install MySQL-devel-community-5.1.30-0.sles10.i586.rpm also and the libraries are at /usr/include/mysql/

Change into the MySQL source directory as follows:

```sh
#cd mysql-5.0.3 (enter)
```

Follow this command by typing:

```sh
#./configure --prefix=/usr/local/mysql --localstatedir=/usr/local/mysql/data --disable-maintainer-mode --with-mysqld-user=mysql --enable-large-files-without-debug (enter)
```

Sit back and wait for a while configure does its thing, once the system returns the prompt to you issue the following command:

```sh
#make (enter)
```

Unless you have a very fast machine this will take some time, assuming the system has returned the prompt to you issue the following command:

```sh
#make install (enter)
```

Cool !, MySQL is installed, there are only a couple things left to do to get it working, first we need to create a group for MySQL as follows;

```sh
#/usr/sbin/groupadd mysql (enter)
```

Then we create a user called mysql which belongs to the mysql group;

```sh
#/usr/sbin/useradd -g mysql mysql (enter)
```

Now we install the database files as follows:

```sh
#./scripts/mysql_install_db (enter)
```

Then we make a couple minor ownership changes:

```sh
# chown -R root:mysql /usr/local/mysql (enter)
# chown -R mysql:mysql /usr/local/mysql/data (enter)
```

Last but not least, we use vi to add a line the ld.so.conf file as follows:

```sh
#vi /etc/ld.so.conf
```

And we add the following line:

```sh
/usr/local/mysql/lib/mysql
```
That’s it, MySQL is installed, and you can run it by issuing the following command:

```
#/usr/local/mysql/bin/mysqld_safe --user=mysql &
```

And as long as we're here we might as well set a root password for MySQL as follows:

```
#/usr/local/mysql/bin/mysqladmin -u root password new_password
```

Where new_password is the password you want to use.

3.6 Compiling PHP

Change into the PHP source directory as follows:

```
#cd php-5.2.8 (enter)
```

Follow this command by typing:

```
#.configure --prefix=/usr/local/php --with-apxs2=/usr/local/apache2/bin/apxs --with-
mysql=/usr/local/mysql (enter)
```

Once the prompt comes back to you issue the following command:

```
#make (enter)
```

Once you have the prompt back:

```
#make install (enter)
```

Once the install finishes and you have the prompt back issue the following command:

```
#cp php.ini-recommended /usr/local/php/lib/php.ini (enter)
```

Then edit that file:

```
#vi /usr/local/php/lib/php.ini (enter)
```

And change the following:

Find the doc_root section and enter the correct path for the directory which serves your web content, such as:

```
  doc_root= "/usr/local/apache2/htdocs/"
```

(this is default for apache2)

Then find the file_uploads section and change it to reflect the following:
file_uploads=Off (for security reasons)

That's if for PHP, now let's see if it all works..

3.7 Testing

Assuming your MySQL process is still running from earlier, let's start Apache by issuing the following command:

#/usr/local/apache2/bin/apachectl start (enter)

This starts the Apache web server, now change into the following directory:

If you get an error like httpd: Syntax error on line 54 of /usr/local/apache2/conf/httpd.conf:
Cannot load /usr/local/apache2/modules/libphp5.so into server:
/usr/local/apache2/modules/libphp5.so: undefined symbol: _efree

Do the following:
Make clean or gmake clean and compile php again. It works

If you get an error like

Linux-vaxr:/usr/local/apache2 # apache2ctl stop
[warn] module php5_module is already loaded, skipping

Then solution is

Obviously there are two LoadModule directives for php5_module in your httpd.conf file. Remove one of them and the message will disappear.

#cd /usr/local/apache2/htdocs (enter)

And using vi create a file called test.php:

#vi test.php

Add the following line to the file:

<?php phpinfo(); ?>

Save the file, then fire up your browser and point it to localhost/test.php. You should see a listing of all kinds of cool info about Apache, PHP, etc. If you do then your set !, if you don't, then take a look at your logs for Apache and MySql, and remember Google is your friend. But hopefully you do, and now you have a fully functioning setup.
Ok, one last step and we'll be done, you have everything running now, but you had to start Apache and MySql manually, that's something you don't want to have to remember to do every time you reboot your machine.

### 3.8 Starting Apache and MySQL Automatically

Let's start with MySQL, as root make your working directory that of the MySQL source directory you worked with earlier, something similar to:

```
#cd /home/xxxx/mysql-4.0.16 <enter>
```

Then, copy the file mysql.server to your /etc/init.d directory as follows:

```
#cp support-files/mysql.server /etc/init.d/mysql
```

Ok, let's create some links in the startup folders for run levels 3 and 5.

```
#cd /etc/rc3.d <enter>
#ln -s ../init.d/mysql S85mysql <enter>
#ln -s ../init.d/mysql K85mysql <enter>
#cd /etc/rc5.d <enter>
#ln -s ../init.d/mysql S85mysql <enter>
#ln -s ../init.d/mysql K85mysql <enter>
#cd ../init.d <enter>
#chmod 755 mysql <enter>
```

That's it for MySQL, it should start automatically now when you reboot your machine. Now let's do the same for Apache, still as root make your working directory that of the Apache binaries as follows:

```
#cd /usr/local/apache2/bin <enter>
```

Then, copy the file called apachectl as follows:

```
#cp apachectl /etc/init.d/httpd <enter>
```

Now, for some more links:

```
#cd /etc/rc3.d <enter>
#ln -s ../init.d/httpd S85httpd <enter>
```
#ln -s ../init.d/httpd K85httpd <enter>
#cd /etc/rc5.d <enter>
#ln -s ../init.d/httpd S85httpd <enter>
#ln -s ../init.d/httpd K85httpd <enter>

And that’s it for Apache !, it should start automatically along with MySQL the next time you boot your machine.

### 3.9 Troubleshooting

Problem:

A php test script to open in the Web Browser and if i type it as http://localhost/test.php then its fine i can see the phpinfo on the page but if i put it as http://localhost/test.phtml or http://localhost/test.html it does not show the php info.

Solution:

Edit httpd.conf file and add the line

```
AddType application/x-httpd-php .php .html .phtml .htm
```

Problem:

When you corrupt some table in your database

Solution:

On the Lamp server we have backup at C:\_hold_\backup_files which is collected every day at night (or time we set in the cron job) ex:

C:\_hold_\backup_files>dir

Directory of C:\_hold_\backup_files
02/03/2009  09:01 PM     1,931,115,265 eis_backup.sql
12/08/2004  05:36 PM       122,407,734 eis_backup.sql.gz
02/03/2009  07:16 PM        85,581,336 www_apache.tar.gz

Take the eis_backup.sql and login into mysql and do SOURCE eis_backup.sql; this will restore the database.
To install LAMP using Yast on OpenSuse refer to
http://en.opensuse.org/SDB:Linux_Apache_MySQL_PHP
A step by step installation guide is available.
4. PHP Code in Detail

Event-driven programming is a paradigm where the program flow is determined by user actions. This is in contrast to batch programming, where flow is determined when the program is written. Event-driven programming is used for interactive programs. The program usually has an event loop, which repeatedly checks for interesting events, such as a key press or mouse movement.

All EIS web applications are developed in an Event Driven Programming paradigm. All the tools are designed to have a similar design, creating a user friendly environment.

All the tools are in a tab format; similarly the underlying code is grouped by tabs. Following is a screen shot of one of the tools:

![Figure 4](image)

**EXECUTIVE INFORMATION SYSTEMS**

Welcome to Hartnell College's EIS

The Executive Information System (EIS) is a tool for accessing information about Hartnell College students and other areas. The information is summarized in tables on the web page. More info in the FAQ tab. FAQ tab has more details about how to use the tool etc. Data are extracted from one of two sources:

- **Datatel Data** - These data represent information currently stored in the Datatel system files. They are dynamic and change as modifications are made to the data in the system. Thus, summary tables produced from the data may vary slightly over time.

- **MIS Data** - These data represent summary information that has been approved through the California Community College Chancellor's Office Management Information System (MIS). They provide snapshots of the college at specific points in time that do not change for a specified semester. Thus, these data are best for making comparisons across time and with other California Community Colleges.

Because of the differences noted above, Datatel data and MIS data will not always match.

Every tool has the above information listed; this is coded in simple HTML. The background color for each tool is different for end-user convenience. Main header part of all the tools have Hartnell College logo and Google search. This can be customized to any other college by replacing the image tags to match their logos and images.

Replace hartnell.JPG, EIS1.jpg with respective logos and image as required. These images must be placed in the right path under /usr/local/apache2/htdocs/EIS folder i.e. web server main folder.
In the html code identify the following line to update your college’s logos and images.

```html
<img style="width: 104px; height: 96px;" alt="Hartnell" src="hartnell.JPG">
```

As mentioned earlier every tool has tabs as shown below:

![Persistence and Success](image.png)

Figure.5

Every tool has name/title, description about the functionality of the tool. Above figure is a screen print of Tool#1 for Persistence and Success analysis. All the front end code is simple HTML and Java Script. It also uses a style sheets to implement the tabs.

Basic architecture of the code is based on the front end design of the tabs. Each tab has php code that is incorporated between `<div>` tags. This php code takes the user inputs of Start and End Term and the Course names and queries the database for various calculations before the front end is updated with the information. Php code has many complex sql queries in it which run against the MySQL database to retrieve data. Returned data is formatted to a user friendly term wise tabular representation.

All the calculations are based on the initial cohort as the reference point. Following code snippet shows the actual code in all the tools. This is just the prototype of the underlying code.
Session variables are used to store user inputs and display back on the screen. We used POST method for security purposes. Information sent from a form with the POST method is invisible to others and has no limits on the amount of information to send.

### 4.1 Connecting to database and Executing Queries

Following code snippet shows the underlying code used to connect to mysql database that is currently being used in all the EIS tools.

```php
$conn = mysql_connect('127.0.0.1','eisuser','eisuser');
if(!$conn) {
    die(mysql_error()); } 
$db = mysql_select_db("tn_test") or die(mysql_error());
```
mysql_connect() is a pre-defined function in PHP to establish database connection to a MySQL database. mysql_select_db() is also a PHP function to select a database after a database connection is established.

Once the connection is established then the code is ready to execute SQL queries and retrieve data from the database. Following code snippet is to execute a SQL query.

```php
$data = mysql_query("Select DISTINCT STC_TERM from sac where STC_TERM IS NOT NULL and TERM_INDEX >= 20001 ORDER BY STC_TERM DESC") or die(mysql_error());
$i=0;
while($info = mysql_fetch_row($data))
{
    $Terms[$i] = $info[0];
    $i++;
}
```

mysql_query() is a PHP function used to send a SQL query to a currently active MySQL database returning an active dataset. This dataset is stored into a variable and used to read data in the next line of the code. The data from dataset can read in much number of ways. Some of them are

- mysql_fetch_row which reads row by row in the dataset
- mysql_fetch_array which reads the dataset into an array variable
- mysql_fetch_assoc which reads the dataset into an associated array and so on
- mysql_num_rows returns number of rows in the dataset

Many such pre-defined functions are available in PHP to read data from a dataset. Some of these functions are used in the code underlying the EIS tools. Please go through the code to find many calculations to display the data from datasets in a user friendly tabular data.

### 4.2 Migrating/Updating Code to work on a Oracle database

Oracle Database Connection Strings in PHP:

There are 3 ways to connect to Oracle database in PHP:

- tnsnames.ora file
- Full connection string
- Easy connect string

Using tnsnames.ora File

The tnsnames.ora file is a client sidefile that maps an alias used by client programs to a database service. It is used to connect to a non-default database. Here you have to have an entry in the tnsnames.ora file, and reference the alias to that entry in your connection code.
PHP code:
oci_connect($un, $pw, 'MYDB');

tnsnames.ora entry:
MYDB = (DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)
(HOST = mymachine.mydomain)(PORT = 1521))
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = MYDB.AU.ORACLE.COM)) )

Full Connection String:
The full connection string does not require the use of a tnsnames.ora file.
You need to enter the full connection string when you connect to the database in your code.

PHP code:
oci_connect($un, $pw,
'(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=mymachine.mydomain)(PORT=1521))
(CONNECT_DATA=(SERVER=DEDICATED)
(SERVICE_NAME = MYDB)))');

Easy Connect String:
This is the most useful way to connect to the database with least overhead.
The easy connect string does not require the use of a tnsnames.ora file, and is an abbreviated
version of the full connection string. you must have the Oracle 10g client-side libraries to use the
easy connect string.

PHP code:
oci_connect($un, $pw, '//mymachine.mydomain:port/MYDB');

1. This is similar to the mysql code that has been developed already. Just replacing
   mysql_connect() with oci_connect() will make the code work like a charm.
2. But before that all the oracle related functions calls must replace mysql_query()
   function calls. Function equivalent to mysql_query() for oracle in PHP is oci_parse().

Ex: Replace  $data = mysql_query("select DISTINCT(STC_PERSON_ID) from sac where
STC_TERM=$myTerm1 and STC_COURSE_NAME=$myCourse1");
With
   $data = oci_parse($conn,"select DISTINCT(STC_PERSON_ID) from sac where
STC_TERM=$myTerm1 and STC_COURSE_NAME=$myCourse1");

3. Assuming $conn as the connection variable which is
   $conn = oci_connect('eisuser','eisuser','127.0.0.1');
4. `mysql_select db ()` can be removed as it does not make sense for Oracle. Oracle has schema names to identify databases. Schema names are used as user names to connect to the database whereas in mysql same user name can be used to connect to different databases.
5. Replace `mysql_fetch_row` with `oci_fetch_row`
6. Replace `mysql_num_rows` with `oci_num_rows`
7. Replace `mysql_fetch_array` with `oci_fetch_array`
8. Delete `mysql_free_result` lines

Voila! Now the PHP code is all set to be used in Oracle environment assuming Apache Web Server, Oracle and PHP are setup right.
5. Datatel Code in detail
6. Useful URLs and References

1. Following URL has sample PHP code to connect to oracle database using PHP and retrieving data

http://st-curriculum.oracle.com/obe/db/11g/r2/prod/appdev/opensrclang/phphol2010_db/php_db.htm