

# Open Administration for Schools 9.0+

## Administrator Documentation

Les Richardson

June, 2016

## Contents

<b>Installation</b>	<b>3</b>
Overview for Newbies . . . . .	3
File Locations . . . . .	3
The Simple Install - Debian/Ubuntu . . . . .	4
Other Installation/Update Scripts . . . . .	6
Program Requirements . . . . .	6
Importing Student / Staff Data . . . . .	7
Configuration . . . . .	7
Getting Started on the Browser Side... . . . .	8
Install Appendix B - Setting Up Webname Resolution . . . . .	9
<b>How to Upgrade Open Admin for Schools</b>	<b>10</b>
<b>Customize OA - Adding New Fields to Students and Staff</b>	<b>12</b>
Overview . . . . .	12
Extending Student Demographics . . . . .	12
<b>Security</b>	<b>15</b>
Web Security . . . . .	15
<b>External User Management</b>	<b>16</b>
Add External Server User Accounts . . . . .	16
<b>Maintenance</b>	<b>18</b>
Multiple Schools (Multiple Directories in /opt/openadmin) . . . . .	18
<b>Moodle 3.0 Setup</b>	<b>19</b>
Authorization . . . . .	19
Enrolment . . . . .	19

---

# Installation

## Version Information

- OA 9.00 – June 2016
- OA 8.50 – December 2015
- OA 7+ – January 2012 and onward.
- OA 6.00 – June 2011
- OA 5.75 – March 2011
- OA 5.25 – January 2011
- OA 4.75 – July 2010
- OA 3.50 – December 2008
- OA 3.25 – August 2008
- OA 2.50 – December, 2007

**Please read** this over before trying to do too much!

## Overview for Newbies

Each school that is running Open Admin would normally have three virtual websites...one for the school office, one for teachers, and one for parents/students. They will live in different directories and be accessible by different people with different rights of access. If installing 2 schools (by running the install-3 script twice), then you would have 6 virtual sites, 3 for each school, etc.

The required files (html files, cgi scripts, configuration settings, and templates for all sites for each school are located in a common folder, normally under the school name. /opt/openadmin/YOURS

The administration or office site has it's files in **admin** for the HTML files and **cgi** for the matching cgi scripts. Similarly the teacher site directories are called **tadmin** and **tcgi**. So also, **padmin** and **pcgi** for the parent site.

If you are limited to having **one** site only (as some Australian schools are), it can also be configured to run on this single site.

## File Locations

Everything is normally arranged:

---

/opt/openadmin/andrews/admin – for St Andrew’s School main admin site  
/opt/openadmin/andrews/cgi  
/opt/openadmin/andrews/tadmin – for St Andrew’s School teacher site  
/opt/openadmin/andrews/tcgi  
/opt/openadmin/andrews/padmin – for St Andrew’s School parent site  
/opt/openadmin/andrews/pcgi

As well as these 6 folders, there are 3 additional folders for:

- **Configuration** - these values are stored in the **etc** folder as separate files (admin.conf, repcard.conf). The new configuration system stores values in a conf\_system table in the database and will rewrite these configuration files for backwards compatibility.
- **Templates** - these templates for reports and editing are stored in the **templates** folder. A subfolder, called **html** contains the templates used to generate the html files of OA (in admin,tadmin,padmin), and is used by the translation system to do so.
- **Libraries** - function libraries used by scripts in cgi (admin), tcgi (teacher), and pcgi (parent) are located in **lib** directory.

Together, these 9 folders make up the Open Admin files for a school.

There is also a **global.conf** configuration file listing all of the schools in a school division, consortium, etc. if more than one installation. If not, then only the version of this file found in the school’s etc directory need be used. It contains enrollment codes as well as read only access credentials to read other schools databases to check for withdrawn student info (during enrollment) to simplify data entry.

If more than one school installation is done, the global.conf file should be placed in

/opt/openadmin/global

so that all school installations may read it, and share data. This is so that withdrawn students from one school can be easily enrolled in another (ie. feeder schools). Even single schools have to add their database name to this file. This is one of the installation problems people have had in the past, and which is now more easily done with the new installation scripts. The location of this directory is set in the admin.conf file and would normally be in your schools etc folder by default.

If you are running the IEP (Special Education) site, you can install those files in:

/opt/openadmin/iep2016/admin – for the special needs website.  
/opt/openadmin/iep2016/cgi

## The Simple Install - Debian/Ubuntu

1. Install Linux (Debian). The current version with the 9.00 release is for Debian 8.1 and Ubuntu Server 14.04.3LTS. Older versions of Linux may not work correctly with the new version of the Apache web server and it’s method of session management, so upgrade if

---

necessary. For a minimal secure server installation in Debian, turn off the desktop, and print server choice, leaving only the 'Standard System utilities' choice to minimize running software and installation issues. Open Admin installs the software it needs with the install scripts 1 and 2 (outlined below).

2. Download Open Admin. Untar the download (`tar xvzf openadmin-X.XX.tar.gz`) and change to the utility/install-debian directory.
3. Run the scripts in numeric order from 1 to 3. (You should be running as root and prepend the scripts with `./` (ie. `./install-1.sh`) Script 1 will install Apache and MySQL/MariaDB and other required apps (via `apt-get`). Script 2 will install perl modules not available as Debian packages.

Script 3 does the main OA site installation including the MySQL database creation. Script 3 can be run again to install other schools on the same server. In order to use script 3 you should decide on/enter the following:

- **School Name** The single word (no spaces, no capitals) for your school that identifies it. We use names like 'cws', 'cbme' or 'willowcree'. This becomes the directory that your Open Admin (OA) files are stored in. It will also be the name of the database storing your school information.
- **Database ROOT Password** This is the master password for the database used when you installed it. This is required in order to create the school database.
- **SCHOOL Database Password** This is the password used for database access for both reading and writing data for that school only.
- **Domain Name for ADMIN site** This the URL/Website name to provide access to the school administrative virtual site. We typically use 'a','t','p' (for admin, teacher, parent) suffix onto a base school name (perhaps as used above). If we use a base of 'cws', then 'cwsa' would be the admin site, 'cwst' would be the teacher site, and 'cwsp' would be the parent/student site. Adding these to your main domain name would give something like 'cwsa.openadmin.ca' for the admin site, 'cwst.openadmin.ca' for teacher site, and 'cwsp.openadmin.ca' for the parent site. Note no `http://` prefix.
- **Domain Name for TEACHER site** As above, like 'cwst.openadmin.ca'.
- **Domain Name for PARENT site** As above, like 'cwsp.openadmin.ca'.
- **Admin Website User Id** This will be a userid (along with a password next) to give access to the admin site. Other userids and passwords may be added using the `htpasswd` program (part of Apache tools).
- **Admin Website User Password** The matching password for admin site access.
- **Database Local Read Only User Id** This is a userid (internal to OA configuration) that along with a matching password is used to access the school database with a read only access (ie. no ability to make changes).
- **Database Local Read Only Password** This is the matching read only password to give read only database access.

At this point you should be able to log into your new admin site installation, once you have added this website address to your local hosts file (on the workstation used to access

---

the site) or on your DNS server that provides the mapping between the website names and the IP address of the server. You **cannot** just use the IP address of the server to access the virtual sites on the server, since this will only give you the default website.

You can then add staff members on the Start/End of Year (EOY) tab, by clicking on 'Add' under the staff management section. By using the staff members userid and password you can then access the teacher site (as soon as you have added the teacher site also, to the hosts file or DNS server).

## Other Installation/Update Scripts

1. The 'removesite' script can be used to remove sites (the reverse of script 3). Install-iep.pl will do the installation of the special education site, also.
2. If **updating** your installation, then use the update script. This will update your installation from as far back as OA 5.25. If you have a version older than that, then manually update the database using files in utility/sql\_older to bring it up to the 5.25 level first. Then use the update script.
3. OA is now installed and accessible using those web addresses (URL's) specified during installation. If you don't yet have these web addresses in your DNS server, you can edit your workstation computer 'hosts' file to map the web addresses to the IP address of the server. Look up the hosts file entry in Wikipedia for more information about the location for your operating system.

You can use 'ifconfig' program to find the IP address of your server if you do not already know it. If the server address was 192.168.1.248 (a local LAN address) and the web addresses you specified in the install-3 script were robin.openadmin.ca, etc. you can then put that into your hosts file:

```
192.168.1.248 robin.openadmin.ca
192.168.1.248 robintch.openadmin.ca
192.168.1.248 robinpar.openadmin.ca
```

**Note:** Using the URL 192.168.1.248/admin in your web browser **will not work**, since virtual sites need the website name (ie. robin.openadmin.ca) passed in the HTTP header in order to deliver you the correct site. Using any variant of the 192.168.1.248 IP address will only give you the default page for the web server.

## Program Requirements

One of the design goals of OA was to make it relatively easy to install and lightweight to run both in hardware and bandwidth demands so that even satellite connections with long latency will work. This has already been demonstrated in many remote schools.

The install 1 and 2 scripts will install all of the required software listed below. Feel free to skim over this material, but it is only for reference purposes. The install 3 script does the installation for a school (to setup your 3 virtual sites).

---

The size of your `logo.gif` should be approximately 150-200 pixels high or so (and a reasonable width). Excessive width (more than 300 pixels or so) may cause layout problems on smaller screens. Pick a size that is pleasing when displayed on the main page. This should give a good starting point. A smaller version of the school logo (about 100 pixels high) should be saved and renamed to `logotn.gif` (Logo Thumbnail) This is the logo visible on the attendance, discipline, report card pages, etc. To conserve bandwidth, these should be as small as possible. The other 2 small image files `tabactive.gif`, `tabmain.gif` (total 2k in size) implement the menu system on the admin site. There is now a script to upload these into position on the Start/End of Year page.

## Importing Student / Staff Data

The student information can be uploaded from a CSV spreadsheet file using the import script on the Export page of the main admin site called 'Import Student Records - CSV'. This is the easiest approach. Simply save your spreadsheet student information as a CSV file, first. Make sure that your birthdates are in standard ISO format: yyyy-mm-dd.

There is a matching script that can also import staff information.

## Configuration

You use the 'Configure Open Admin' button on the Start/End of Year Page to configure Open Admin (OA).

This system stores configuration data in a database table 'conf\_system' and also writes those values into various files in the `/opt/openadmin/YOURSCHOOL/etc` folder to be read by scripts. We are gradually moving away from the use of those files to using the database table directly.

The main configuration file is called **admin.conf** and is read by all scripts from main admin (cgi), teacher (tcgi), and parent (pcgi). Setting correct values is very important since it affects all program operations.

There is also a global configuration file (nominally located in the school etc folder and called `global.conf`) that stores information about all of the schools on the local server (normally, that would be all participating schools in the division). This information is used to enable global reporting. This is used by central office for reporting as well as used by the schools to enable school transfers of demographic information. Once a secretary or admin in a school finds the student number of a withdrawn student, they can simply put that number into the normal `entry/withdraw` input box on the main page to register that student and have all demographic information transferred.

I've located the global config file in the school etc folder, and set the `admin.conf` configuration file to point to its location there. This is because most installations are for single schools, not large division installs.

If you are doing a multiple school installation then move this file to a *global* folder at top level (ie. `/opt/openadmin/global`). Add all databases to it. Then edit all school `admin.conf` files so that their `$globaldir` setting points to this folder.

---

The *Look and Feel* of all school sites can be changed by using the 'Set Background Color' and 'Set Logo Files' buttons on the EOY page.

The HTML pages are basically *separate* from the scripting, so that the look of all school sites can be changed extensively if desired by simply changing the html files and the CSS files. The only issue will be to maintain those elements in the CSS files used by the scripts to control report rendering, etc.

## Getting Started on the Browser Side...

From this point on, secretaries may take over, if desired. The following functions are on the Start/End of Year page (There is more about this in the User Documentation under *Getting Started*).

1. **Add your teachers and other staff members.** For a teacher in a split situation (teaching 2 grades in the same classroom), add those 2 grades into the grade area, but with the same homeroom value. For jobsharing teachers, add each one separately with the same homeroom.

In a middle years or high school (or with any subject specialists who only teach certain subjects such as music or foreign languages), the homeroom and grade fields are normally left blank. The teacher is grouped with students by their subject enrollment in his/her class.

2. **Set School Year** See the section on setting the school year including terms and track and their meanings within OA.
3. **Enter school dates** that school is not in session, Monday to Friday. This is located on the Start/End of Year page also.

This includes any school holidays within the year and teacher inservices, etc. (The system automatically assumes that school is not in session on Saturday/Sunday). This should now correctly do attendance calculations, etc. The DayInCycle setting is used to mark those days that actually *count* in schools using different types of cycles (ie. 6 day cycles) for classes, even though the students may not be in class (ie. due to teacher professional development).

You are now ready to enter attendance, discipline, etc.

---

## Install Appendix B - Setting Up Webname Resolution

You can set these website locations in individual computers in their *hosts* files so that they can access these sites without any DNS (Domain Name) records being installed on your DNS server (that maps webnames to IP addresses). This can be quite useful in initial testing before any DNS updates have been done. On Linux workstations these are located in */etc/hosts*. On Windows workstations (XP) these are found in:

```
c:\windows\system32\drivers\etc
```

Look on the wikipedia website ([wikipedia.org](http://wikipedia.org)) under *Hosts file*.

The entries in the hosts files look like this for Clifford Wuttunee School:

```
cwsa.mysd.ca 142.165.5.25 # admin site  
cwst.mysd.ca 142.165.5.25 # teacher site  
cwsp.mysd.ca 142.165.5.25 # parent site
```

Using hosts files will help *hide* the virtual sites to some degree, since there will not be DNS records for them. However, they have to individually maintained. Also, in terms of security, avoid placing links to admin sites on school web pages. (security by obscurity). The next step up in security is the use of SSL to give encrypted connections to/from websites.



---

# How to Upgrade Open Admin for Schools

Installation for OA 5.25 - January 2011

There have not been any large changes to the installation process in this release. The usual process is:

1. **Move the old version out of the way and insert the new version.** I would suggest the excellent file management tool called 'Midnight Commander' (mc), based on a much earlier piece of software called 'Norton Commander'. You can mark files with the 'insert' key and move/copy files with F6/F5 keys between alternate panes of files.
  - Move/Rename the old version to, for example, *andrews.old*.
  - Create a new directory of the same name and copy the new folders of OA 5.25 **school** into this *andrews* folder. ( normally /opt/openadmin/andrews ) Directly below this we should have the admin and cgi directories (ie. /opt/openadmin/andrews/html, /opt/openadmin/andrews/cgi ), etc.
  - Change the ownership of the new *andrews* folder so that it matches that of the UID/GID that the web server is running as. This is 'www-data' for both user and group in Debian.
2. **Copy the configuration files** from the old etc folder into the new etc folder. Before doing so, copy the *admin.conf* and rename it to *admin.conf.525*. Edit them both and add any new values at the top of the 5.25 *admin.conf* to the older *admin.conf* from your existing installation.

Copy the *studentnumber* file (which contains the next available student number), the *term* file (which contains the current term) and the *receiptnumber* file (if present),(which contains the next receipt number for the fees system).

Copy the *repcard.conf* file (report card system) to the new etc folder from the old etc folder if changed, while renaming the new ones with a .525 ending. Do the same for any other .conf files that have been changed in previous install. (Look at file dates)
3. Edit the (old) *admin.conf* and the *admin.conf.525*. Copy the changes at the top of the new 5.25 file and make any other necessary changes in file locations, etc. Do the same with the *repcard.conf* and the new *repcard.conf.525* and update changes at the top of the file.
4. Copy the previous 2 school logo files (*logo.gif* and *logotn.gif*) in *admin/images* into their new location. Do the same for the *admin.css* file which contains the color information you may have modified previously (in /*admin*).
5. **Backup your data** using your backup tool of choice. Mysqldump works well too, creating a portable .sql if it's output is captured into a backup file.
6. Update the database table structure using the provided update scripts ( in the *utility/sql/475to525* folder of the download). The sql files should be used for earlier versions of OA first, if updating from a much earlier version.

---

Now update the meta data by using the metadata update script on the eoy (end of year) page. It is the Update Metadata button. You could also empty the meta table and use the meta475.sql script also. However, I would suggest the the button method, with minimal changes.

7. If you would like to use the *iep* (special education) application, please read the installation notes. It is a separate website with it's own html and cgi folders. It would normally be installed into /opt/openadmin/iep. Use the install script to automate this process.
8. **TaDa.** Sit down and gloat. Gloating is **Important**. Tell your Spouse or Significant Other how wonderfully talented you are, etc.

---

# Customize OA - Adding New Fields to Students and Staff

## Overview

Open Admin can have both student demographics and staff information extended with additional fields with very little effort. The staff table already contains several additional fields for use by international schools for storing staff certification, etc.

The only negative to this process is that a report must be created to actually display this information. The Student Roster reports are templated so that they can be changed and new templates created with the template generator. Staff reports are also template based.

This is made possible by:

1. **Templates** – files located in a *template* directory hold layouts for student addition and editing, student search, and student viewing. These templates are loaded by scripts to enable editing of the student table(s): student, studentwd(withdrawn students) and prereg (preregistered students).
2. A **Metadata** table – a table storing information about the structure of the staff and student tables in OA. There are additional fields in each record that are used to control form element length and type, default values, etc. This information can be modified by using the meta scripts (edit and update) on the Eoy page.

## Extending Student Demographics

As a result of this, additional fields may be added to the end of the student tables. All 3 student tables must remain identical since records are moved directly from one table to another during withdrawal, re-enrollment, etc. Fields should not be removed from the current student table format since older scripts may depend on this organization.

The templates are then updated to add these additional field(s) to the templates. The final part of the process would include, possibly, some additional custom reports that support these new fields. OA really requires a report generator, but that must lie in the future.

The following scripts are now *template aware* and will display the additional fields added to the student tables and templates:

1. **Student Search** - cgi/studsearch.pl (uses the studsearch.tpl template).
2. **Student Enrollment** - cgi/entry/sentryX.pl scripts (use the student.tpl template).
3. **Student Editing** - cgi/studed.pl (and studeled.pl now supports editing of any of the 3 student tables)(also uses the student.tpl template)

The templating *technology* is currently making use of simple regex matching and replacement. Future use of templating modules is being considered as other needs develop and the pros and cons of each module are weighed.

---

The templates themselves are simply HTML markup blocks with the following tags that are replaced during script execution:

1. `<*desc*>` tags that are replaced by a fieldname descriptor from the meta table. These descriptors will be replaced during system configuration with alternate language versions stored in the meta table.
2. `<@value@>` tags that are replaced with an entry form with features controlled by the meta table: input type, field length, maxlength, etc when editing. These replaced with actual values from tables during search and viewing.

The meta table currently has 2 scripts:

1. `metaupdate.pl` – a script to read the current database and update the metatable to match any additional fields added to or fields removed from the underlying tables.
2. `metaedit.pl` – a script to allow edits to any of the field settings in the meta table. Currently this only applies to the student and staff tables. (studentwd and prereg tables are ignored and the settings apply to all 3 tables, since they are identical structurally).

This script allows you to:

- Change the field name descriptor if a `<* *>` tag exists in the template for it.
- Default values – these will be the values available if the form entry type is a selection list. The following rules apply for select entries:
  - Each entry must be separated from the other by a space.
  - Multiword entries they must be joined together with an underscore (`_`). It will be removed before displaying.
  - Initial tilde characters (`~`) force a blank at the start of a selection box.
- Formtypes – text input boxes, selection lists, checkboxes, textareas are allowed. Checkboxes in edit screens become text input boxes, since they are impossible to use to clear values.
- View size – controls the size of the box for text input form types.
- Required – controls whether this is a required field when enrolling students.

The **process of customization / table extension** consists of the following steps:

1. Create additional field definitions in a file (`update.sql`) . Use this file to extend the student table (`mysql -ppassword database < update.sql`). Copy and edit this file so that you also extend the studentwd and prereg tables in the same way. (`updatewd.sql`, `updatepre.sql`)
2. Edit the template folder to add the new field(s) to the templates at a desired location (or use the template creator). This will allow for addition and editing of the new fields during enrollment and demographic updates as well as searching and viewing of the field(s). They can also be printed out on the student viewing page.

- 
3. Run the `metaupdate.pl` script to add those new field definitions to the meta table. Each new field in the student table will become a new record in the meta table. Now run the `metaedit.pl` script, choose to edit the student table and then add your new field characteristics.
  4. You now have a customized version of OA with more student fields. As new releases of OA come out, your custom templates will have to be placed in the new versions's template directory to match your extended student table(s). The meta data will not be altered so this may be left.
  5. **If you are using MySQL version 4.x**, the *studentall* table must be rebuilt due to the extended student table . MySQL version 5.x will not require any change since it uses a view. The simplest approach is to first get the data from the current student table and put into a text file:

```
mysqldump -d -ppassword -u user database student > student.sql
```

(where *password* is your mysql user password, *user* is your mysql user, and *database* is your database.)

Now do the same thing to create a data dump of the studentall table:

```
mysqldump -d -ppassword -u user database student > student.sql
```

Edit the student.sql file and change the name from *student* to *studentall*. At the bottom of this file copy the identical section of the studentall.sql file replacing the MyISAM section with the merge table information.

Now drop the studentall table from your database (drop table studentall), and create the new one:

```
mysqldump -d -ppassword -u user database student < student.sql
```

(the student.sql file, now changed, is actually the file used to create the new version of studentall)

You are now done.

---

## Security

This section deals with the issues of making sure that your school data remains intact and accessible only to authorized individuals.

### Web Security

Currently, both the Teacher and Admin sites are protected by basic authentication (as outlined in the installation section). This method of authentication sends passwords as clear text over the 'wire'. The parent site is freely accessible, but protected by passwords passed to the running script.

This could be improved by moving to *Digest Authentication* which is a step up in security and doesn't send clear text over the wire. The configuration setup for this is outlined in the Apache documentation.

The next step is to add full https (SSL) support to the server and use certificates to provide banking level protection to the transactions. These instructions are now provided by the current installation and are written by Andy Figueroa.

Another entire area of concern is the use of passwords. Access to the teacher site is by means of a single shared password (via a single password file). Other apache modules could be used to store separate passwords for each teacher in a mysql database (and in fact use the teacher table for these passwords). Apache 2.x has some advanced authentication modules to allow this. Thus the admin site could be used to directly manage teacher access to the website (as well as the attendance entry, marks entry, etc.)

Since passwords are the first line of defense, they should be carefully chosen so that people can remember them, but not be so simple that they can easily be guessed or cracked by a brute force attack.

Some other ideas:

1. Don't provide links on any publicly visible web sites to your admin or teacher sites. This is only security by obscurity, but it can't hurt.
2. For the truly paranoid, don't place the virtual admin and teacher sites into your DNS records. Instead, configure client workstations by placing these records in your 'hosts' file (either in /etc/hosts or c:\windows\hosts). The hosts file would contain the IP address and then the domain name of the school/division server.

---

## External User Management

Since the secretary and/or administrators maintain a list of enrolled students, it makes sense to use this information to manage users on external servers. This would allow a student, once enrolled, to have his/her login to workstations, email accounts, and other school based resources added from this information.

### Add External Server User Accounts

This system consists of 2 parts:

1. Scripts to export the user information (ie. name, student number and password) and/or account changes (reset password, lock or unlock account) to an 'openadmin' account on an external server.
2. A script, run periodically by cron on the external server, to take this user information and add missing user accounts on the external server or reset password or lock/unlock the account.

The steps in the process are:

1. On the external server, create an 'openadmin' user account. This will be the account into whose home directory the OA user file will be copied. Also create a 'students' group to which the added user accounts will be added.
2. On this server, place the 'adduser\_cron5.pl' script into a location such as /opt/openadmin/usermanage and make sure it is executable. Then create a cron job to execute this script periodically once every 10 or 15 minutes (or as desired). This script will check for the presence of the user file or reset file and if found add new users and reset values. Typically the cron job is added into the 'root' file in /var/spool/cron, as a line like:  

```
0,10,20,30,40,50 /opt/openadmin/usermanage/adduser_cron5.pl
```

Restart the cron service so it will read and check every 10 minutes (in this example).
3. On the Open Admin server, edit the /opt/openadmin/YOURSCHOOL/etc/usermanage.conf and change the IP address to that of the external server, and the password to that of the 'openadmin' account (or whatever account created) on the external server. This will allow the script to copy a created user file and/or reset file (in CSV format) to that external computer.

In order for the scripts to run you will need to add perl modules Text::CSV\_XS and Expect on the external server (typically with the CPAN method command: perl -MCPAN -e 'install Text::CSV\_XS') or via apt-get for libtext-csv-perl library. You will also have to edit the external 'adduser\_cron5.pl' if your version of Unix/Linux uses different commands/syntax to add users and passwords.

---

On the Open Admin server the new scripts also require the `Net::SCP::Expect` perl module (`apt-get install libnet-scp-expect-perl` ) that moves the files to the external server. This requirement has been added to the updated documentation.

There are also additional settings in the `adduser_cron5.pl` script on the external server (and initially located in the download in `utility/usermanage`). These settings can be used to chain user updates to multiple servers, etc.



---

# Maintenance

## Multiple Schools (Multiple Directories in /opt/openadmin)

In order to maintain multiple schools running Open Admin, I have done the following:

1. In the */opt/openadmin* folder (where all schools are located), I have created an *updates* folder.
2. In this folder place a copy of the *oafscopy.pl* perl script (from the utility directory of the download) and modify them to copy files into all of your school folders (add your school directories to the list).

An upcoming version will have to take into account modifications to a file that are specific to a school. (ie. customization). If those changed files are placed in a directory at the school level (ie. same level as *etc* and *cgi*) instead of where they normally run and then a symbolic link added to the normal folder pointing to the custom folder, a new *oafscopy.pl* script could *detect the presence of the symbolic link* and not copy the update for this school (although it should warn the user of this event).

3. When updating scripts, you would copy the update(s) to this update folder and use *oacopy* to update all of the other schools on the same server. If there are multiple scripts, they can then be tarred/zipped into a single archive file. This file is then copied to the other server using *scp*. I use date archive names (i.e. *OA20101010a.tgz* for the first archive of October 10th, 2010).
4. On the other server, move the archive file into the */opt/openadmin/updates* folder, uncompress it, and then use *oacopy* to update all of the other schools. This can be done quite quickly for small numbers of servers.

In the Notes folder, mentioned above, I place notes to myself about particular settings in each school, any customizations for any school, etc. This is particularly important for student numbers since I assign each school a particular range of student numbers (which is stored in the */cgi/entry/studentnumber* file). I use a large range to that this setting would last at least 6 years or better. At that point, a centralized (per division) routine can be used to assign division student numbers. For now, this method of storing the next available student number in a file is simple and reliable.

---

## Moodle 3.0 Setup

OA can integrate with Moodle in several ways:

- It can authenticate students for login using their OA credentials (student number and password).
- It can automatically enrol students into their courses in Moodle when enrolled in the matching course in OA.

It can do this by using the 'External Database' option for both authorization and enrollment. The authorization process can use the **student** table directly. The eval table which holds student report card marks indicates their subject-section enrollment however cannot be used since it has a record for every term and no 'role' field. A new table called **ext\_moodle** is used for the external program Moodle to read. It is maintained by the enrolment script in OA. The next section outlines how to set this up in Moodle.

### Authorization

Authorization is setup in Moodle by:

- Login as administrator.
- Under Site Administration, choose Users, Authentication, Manage Authentication.
- Turn on the Eyecon for 'External Database', and move it up as far as possible using the up arrow beside the eyecon.
- Configure the external database by clicking on 'Settings'.
- Set DB Name, DB User, and Password to the values used by your school's OA database.
- Set Table to 'student', Username field to 'studnum', Password field to 'password', Password format to 'Plain Text' (default).
- In the lower 'Data Mapping' section, more fields may be set also.
- Set First Name to 'firstname', Surname to 'lastname', Email to 'email', City/town to 'city1', ID Number to 'studnum'. Others may be set also, as desired.
- Click on 'Save Changes', to save these values.

### Enrolment

Enrolment is setup in Moodle by:

- Login as administrator.

- 
- Under Site Administration, click Courses, then Enrolments.
  - Click 'Enable' for the External Database.
  - Configure the external database by clicking on 'Edit' in the settings column.
  - In the first section called External Database Server Settings, set the following to allow Moodle to connect to your school database and the ext\_moodle table.
    - enrol\_dbtype: mysql
    - enrol\_dbhost: localhost
    - enrol\_dbuser: Your Server user
    - enrol\_dbpass: Your Server password
    - enrol\_dbname: Your School Database name
    - enrol\_dbtable: ext\_moodle
  - In the next section called Enrolment (remote) Database Fields, set the following values:
    - enrol\_localcoursefield:idnumber
    - enrol\_localuserfield:username
    - enrol\_db\_localrolefield:shortname
    - enrol\_remotecoursefield:subjsec
    - enrol\_remoteuserfield:studnum
    - enrol\_db\_remoterolefield:role
  - The rest of the values may remain as is (or as desired).
  - Click on 'Save Changes', to save these values.