

Integrating Moodle with Google Apps Education/Premier Edition

Administration Manual

by

Rashan Anushka - UCSC

{rap@ucsc.cmb.ac.lk}

V1.0

| | |
|---|---|
| Administration Manual..... | 1 |
| Rashan Anushka - UCSC..... | 1 |
| Introduction..... | 2 |
| Requirements..... | 3 |
| Current Status & Future Development..... | 4 |
| Installing and Configuring the code for Moodle..... | 5 |
| Configuring Google Apps..... | 7 |
| Appendix: Generating Certificates..... | 8 |

Introduction

Google Apps is a collection of collaboration and communication tools hosted by Google. Currently it has 3 editions; the Standard Edition, the Premier Edition and the Education Edition. The good thing about Google Apps, as far as the educational institutes are concern, is that the Education Edition, which is very similar to there Premier Edition, is given freely and it provides a wide range of services including some communication tools like email, gTalk and some collaborative tools like Calender, docs/spreadsheets etc... Also it has some well documented APIs which can be used to integrate all these services into institution's existing IT infrastructure.

On the other hand Moodle is a course management system (CMS) designed using sound pedagogical principles, which is distributed freely under GNU GPL license. It has a very large user base as well as a active developer base. According to <http://moodle.org/stats>, Moodle is being used by about 15 million users around the world.

This document's primary focus is on how to integrate these two very popular and widely used systems so that users can experience the excellent services provided by these two products under one roof.

Requirements

- Google Apps Education or Premier Edition
- Apache HTTP Server v2.2.3 or later
- PHP v5.2.1 or later with (php-xml extension)
- xmlsec v1.2.9 or later
- xmlsec-openssl (OpenSSL plugin for xmlsec)
- Moodle 1.8 or later

Note: Currently the code is tested on Linux environment with above configurations only.

Current Status & Future Development

Currently the following is possible via Moodle;

- For Administrators
 - Create new accounts
 - Suspend User Accounts
 - Delete User Accounts
 - Set User Passwords
- For Normal Users
 - Request new email account
 - Change password
 - Access gApps mailbox via SSO (no need to login again)
 - Set gApps email as the default Moodle email account

The following developments have been planed in future;

- Managing and using email lists, nick names
- Full integration with other services like Docs/Spreadsheets, Calendar, etc...

Installing and Configuring the code for Moodle

Install the pre-requisites

- In RedHat based linux;

```
yum install httpd php php-xml xmlsec1 xmlsec1-openssl
```

- Download the Moodle-GApps integration code package from the from the [project page](#) or download via CVS

Instructions on Downloading via CVS

Note: The following download the code into the current folder

```
cvs -d:pserver:anonymous@gapps-moodle.cvs.sourceforge.net:/cvsroot/gapps-moodle login
cvs -z3 -d:pserver:anonymous@gapps-moodle.cvs.sourceforge.net:/cvsroot/gapps-moodle co
-P .
```

- Now you can extract the code in the root of the moodle, but it is always safe to follow the following manual method;

- Extract the package (or download via CVS) in a temporary location- the package has the following structure;

```
<temp_location>
|__ admin
|   |__ gapps
|__ mail
|
|__ blocks
    |__ gapps
```

- Copy the **gapps** folder inside the **admin** folder into <YOUR_MOODLE_ROOT>/admin/

- Copy the **gapps** folder inside the **blocks** folder into <YOUR_MOODLE_ROOT>/blocks/

- Copy the **mail** folder into <YOUR_MOODLE_ROOT>/

- Using your favorite editor, open the **config.php** file inside the **gapps** directory and update your account settings;

```
define('GAPPS_USERNAME', 'your_googleApps_admin_username');
define('GAPPS_DOMAIN', 'your_googleApps_domain_name');
```

```
define('GAPPS_PASSWORD', 'your_googleApps_admin_password');
```

- Using your favorite editor, open the config.php file inside the mail directory and update some settings including your account settings (yes you have to do it again!);

```
define('GAPPS_USERNAME', 'your_googleApps_admin_username');  
define('GAPPS_DOMAIN', 'your_googleApps_domain_name');  
define('GAPPS_PASSWORD', 'your_googleApps_admin_password');  
define('XMLSEC', path_to_xmlsec_binary); // if the binary is in your PATH,  
then it is ususally 'xmlsec1'  
define('LD_LIBRARY_PATH', 'path_to_xmlsec-openssl_library'); // Usually it is  
'/usr/bin'
```

- Using your favorite editor, open the **misc.php** file inside the **<YOUR_MOODLE_ROOT>/admin/settings** directory and **add** following code at the end of the file;

```
$ADMIN->add('misc', new admin_externalpage('gapps', 'Google Apps', "$CFG->  
wwwroot/$CFG->admin/gapps/"));
```

- Now click on **notifications** on the Site administration block

Configuring Google Apps

- Log into your Google Apps control Panel; usually http://www.google.com/a/<YOUR_GApps_DOMAIN>
- Go to **User accounts** and click on **settings**
 - Mark the **Enable provisioning API** check box
 - Click on **Set up single sign-on (SSO)**; you will taken into the SSO settings page;
 - Mark the **Enable Single Sign-on** check box
 - Update the settings as follows;
 - Sign-in page URL**
`http://<YOUR_MOODLE_PATH>/mail/index.php`
 - Sign-out page URL**
`http://<YOUR_MOODLE_PATH>/index.php`
 - Change password URL**
`http://<YOUR_MOODLE_PATH>/mail/index.php`
 - For testing upload the verification certificate located at `<THE_MOODLE_ROOT/mail/keys/DSAPublicKey01.key`

Note: For production use, please follow the instruction given bellow on generating certificates.

Appendix: Generating Certificates

This section will briefly explain how to generate your own certificate and PKI keys by using OpenSSL to be use in a production environment. The keys and certificates can be based on either RSA or DSA.

- **Generating the private key - Note: Use ONLY RSA or DSA**

- For RSA based keys**

- ```
openssl genrsa -out myrsaprivkey.pem 1024
```

- Since the code expect the keys in DER format, the following is necessary;

- ```
openssl rsa -in myrsaprivkey.pem -pubout -outform DER -out myrsapubkey.der
openssl pkcs8 -topk8 -inform PEM -outform DER -in myrsaprivkey.pem -out \
myrsaprivkey.der -nocrypt
```

- For DSA based keys (Recommended)**

- ```
openssl dsaparam -out mydsaparam.pem 1024
```

- ```
openssl gendsa -out mydsaprivkey.pem mydsaparam.pem
```

- Since the code expect the keys in DER format, the following is necessary;

- ```
openssl dsa -in mydsaprivkey.pem -outform DER -pubout -out mydsapubkey.der
openssl pkcs8 -topk8 -inform PEM -outform DER -in mydsaprivkey.pem -out \
mydsaprivkey.der -nocrypt
```

- **Generate the certificate**

- ```
openssl req -new -x509 -key mydsaprivkey.pem -out mycert.pem
```

- Note:** In the case of RSA keys, you should replace the private key above with myrsaprivkey.pem

- **Install the keys - Note: Use ONLY RSA or DSA**

- Upload the mycert.pem into Google Apps via the Google Apps administration panel
 - Copy the mydsaprivkey.pem and mydsapubkey.der into "keys" directory inside "mail" folder