Table of Contents

Last modified:

Tuesday, March 7, 2006

Software Requirements

PHP requirements

- PHP Version 4.1.0. Versions 4.3.x or higher are highly recommended.
- PHP IMAP extension (necessary even if only POP3 access is needed).
- PHP PCRE (Perl-Compatible Regular Expression)
- PHP Session support

Supported browsers

- Windows
  - Internet Explorer 4 or higher. 5.5 or higher highly recommended to take advantage of DHTML user interface enhancements.
  - Netscape 6.1 or higher
  - Mozilla 1.0 or higher
  - Firefox 1.0 or higher
  - Opera 6.0 or higher
- Linux
  - Netscape 6.2 or higher
  - Firefox 1.0 or higher
  - Opera 6.0 or higher

Installation

Installation Process

1. Extract the files from the V-webmail archive. Make sure the htdocs directory is accessible via your web server. If you extract the entire archive into a sub-directory of your website (eg. http://www.domain.com/v-webmail/) then this would make your url to V-webmail this: http://www.domain.com/v-webmail/htdocs/.

   Now this throws up problems since it exposes the other directories, such as includes and userdata, but this can be avoided by setting the document root of your V-webmail site to the htdocs directory. For example you could add this to your apache config (the path will obviously need altering to suit your setup):

```
DocumentRoot /www/vhosts/v-webmail/htdocs/
DirectoryIndex index.php
```

2. ***Permissions!*** Ensure that the webserver can write to the userdata, logs and tmp directories. Ideally only the webserver should be able to read/write these directories. See the security section for more information on permissions.

   You now need to edit the configuration file. First, make a copy of *config.xml* which is located in the *config* directory. The copy should be called *local.config.xml*. This copying is necessary for easier upgrading:

```
cp config/config.xml config/local.config.xml
chmod +644 config/local.config.xml
```

3. Now open up *local.config.xml* and follow the instructions in there; each configuration option is fully documented. This file contains all settings for V-webmail.

   Important: Do not turn off *config_checks* until you are certain that V-webmail is up and running. If there is a problem with the configuration and this option is off, you may experience unpredictable behavior.

If you only want to allow access to one or a certain set of servers or you wish to change the default options presented on the login page, you will need to edit *local.config.xml*, in between the *servers* element. The most important setting here is servers_strict. With this 'false' you can connect to any server you specify with the default login page, however with it turned on ('true') you can only connect to the servers specified in this file. If you turn this on the default login template will automatically change from presenting the server/port/folder prefix options to instead presenting a single dropdown listing available servers to connect to. The text visible in this drop down is taken from the server descriptions.

The part in between the *hostinfo* element should be used to define any virtual hosts you wish to use. If you only want to have a single vhosts, then this part simply serves as a further configuration file. Multiple virtual hosts should be used when you wish for different domains to have different settings. eg mail.customer1.com and mail.customer2.com could both have a different set of templates for branding purposes. Each option is documented in here.

Very Important: It is not possible to use certain characters in the config file. These illegal characters include < (smaller than) and > (greater than).

Should you also wish to customize the default user and addressbook, you can do so by making copies of the config*/default.user.php* and *config/default.addressbook.php* files and prepending their names with "local.", so you end up with *local.default.user.php* and *local.default.addressbook.php*. You can then modify these files to suit your needs. These are not ini files, they are actual PHP scripts, so be extra careful when modifying them, take backups, and don't use single quotes.

With regard to default special folders, you can enter them here in the *local.default.user.php* file, but you must prefix the names with the folder prefix that you're using. So if you're using INBOX. as the folder prefix, and you want users' trash folder to be "Trash", you would specify "INBOX.Trash".

4. You should now be able to login with a valid username/password combination for the server details you have specified.

Using MySQL to store user preferences

If you want to store your users' preferences and addressbook information in a MySQL database (useful if you have a large number of users, and/or multiple webservers) you need to perform the following steps.

1. Follow the instructions in *config/local.config.xml*, in between the *general/prefs_storage* element for changing the preferences driver.
2. Create a table named "*userprefs*" in your database. The following table definition can be used:

```
CREATE TABLE userprefs (
  userprefs_id mediumint(8) unsigned NOT NULL
auto_increment,
  userprefs_host varchar(128) NOT NULL default '',
  userprefs_user varchar(64) NOT NULL default '',
  userprefs_data text NOT NULL,
  PRIMARY KEY  (userprefs_id),
  UNIQUE KEY host_user (userprefs_host,userprefs_user)
) TYPE=MyISAM;
```

3. You will now be storing user preferences and addressbook information in the database. It is important to not delete the *config/default.user.php* or *config/default.addressbook.php* files however, as these are still used for the default preferences and addressbook.

Using PostgreSQL to store user preferences

If you want to store your users' preferences and addressbook information in a PostgreSQL database (useful if you have a large number of users, and/or multiple webservers) you need to perform the following steps.

1. Follow the instructions in *config/local.config.xml*, in between the *general/prefs_storage* element for changing the preferences driver.
2. Create a table in your database with the name specified. The following table definition can be used:

```
     CREATE SEQUENCE userprefs_userprefs_id_seq;
CREATE TABLE userprefs (
userprefs_id smallint DEFAULT nextval('userprefs_userprefs_id_seq')
NOT NULL,
userprefs_host varchar(128) DEFAULT '' NOT NULL,
userprefs_user varchar(64) DEFAULT '' NOT NULL,
userprefs_data text DEFAULT '' NOT NULL,
PRIMARY KEY (userprefs_id)
);
CREATE UNIQUE INDEX host_user ON userprefs
(userprefs_host, userprefs_user);
```

3. You will now be storing user preferences and addressbook information in the database. It is important to not delete the *config/default.user.php* or *config/default.addressbook.php* files however, as these are still used for the default preferences and addressbook.

Customisation

Templates

One of the areas where V-webmail really shines is the handling of templates. Templates are held in a sub-directory of the templates directory, with the name of the sub-directory denoting the set. Each set can be specific to a virtual host, or available to all virtual hosts, as defined in the *vhosts.php* configuration file (config directory). The usual header/footer combinations are used, each being in it's own file (header/footer.html).

The templates can be changed to suit your needs, though you should be careful to avoid removing necessary parts which construct the programs output. Most installations will probably only ever need to change the header/footer combinations and the stylesheets (htdocs/css/).

If you do need to alter one or more of the templates there are two ways. Method one is to create a new template set by creating a new directory in templates/ and giving it an appropriate name. Add this new set into your *vhosts.php* configuration file (templates directive, see the comments above it for details), and then copy and modify the templates you need changed. Don't worry about copying all the templates to the new set, as V-webmail will automatically fall back to the v-webmail set if it can't find one it needs in the new set. This method allows you to create a new set, and only change the ones you need to.

Method two involves copying the file you wish to change, to a new file with the same name but prefixed with "local.", eg *local.email.list.html*. This file will be used instead of the original. This method is advisable if the change you wish to make is minor, or you don't need a new template set.

Important: You should not delete or rename the v-webmail template set. This is used in case any other sets do not have all the required templates.

Hooks

The purpose of the *hooks.php* file in the includes directory is for user defined functions which are called at key points throughout the program. There follows a list of these hook functions and their uses. More documentation for individual functions is available in the hooks.php file. To use these functions, you should not edit the file directly. You should instead make a copy of the file, call it *local.hooks.php*, and edit that instead.

hook_usernameLookup()

This function is called during login. It takes an argument of the username/password and returns a two element array of username and password eg: array('username, 'password') Amongst other things, it can be used in particular for doing username aliasing. An example of this is given in the file. It's a basic example that could be extended to LDAP/database lookups.

hook_emailLookup()

This function is called during login. It is however only called if the users email address preference is blank, (it's therefore not called if the option allow_changeable_email is 'false' (*local.config.xml*)). It allows you to set a default email address for the user, so they aren't confused when the system won't allow them to send mail due to having no From address. The function should return a single string, which is the intended email address. The default function uses the "domain" option of the current server (from the *servers* section of the config file) and combines it with the username, to build the email address. You could, for example, use this function to lookup the email address in a database, or perhaps an LDAP directory.

hook_rewriteURL()

This function is called called every time when an URL which points to some V-Webmail component is built. Components include images and scripts. It takes an URL as argument and returns an altered URL. One possible use for this hook is to convert http-URLs to https-URLs, allowing for SLL access to V-Webmail via an SSL proxy (shown in the example in *hooks.php*).

Further hook functions will be added as and when is necessary or appropriate, suggestions for these are more than welcome.

Security

These security notes should be taken into consideration when installing and configuring V-webmail. It is by no means a comprehensive guide to system administration.

Directory Structure

The application was designed with virtual hosting firmly in mind. What this translates to is that the directory structure is thus:

```
/v-webmail
  |
  +-/config
  +-/help
  +-/htdocs   <--- Point your DocumentRoot here
  +-/includes
  +-/lang
  +-/logs
  +-/manual
  +-/templates
  +-/tmp
  +-/userdata
```

As pointed out, the DocumentRoot (the root folder of your website) should be pointed at the htdocs directory. This is so that the rest of the directory structure is hidden when accessing the website.

If you do not do this, or you don't have access to set up virtualhosts, then you can still use V-webmail, but the directories will be traversable. This means people could in theory gain access to the userdata folder, or the tmp folder (file uploads reside here).

All is not lost however, since V-webmail can quite happily accommodate directory name changes. So if you're forced to use this less secure method, then simply rename the userdata/tmp/logs/includes folders and update the *config.php* file (located in the htdocs directory) to reflect the changes.

If you have to use the more open method of installation, you can rename the htdocs directory, to something more friendly, eg webmail. No *config.php* changes are necessary if you do this.

Permissions

The userdata and tmp directories have to be webserver writable, so obviously all the inherent issues with webserver writable directories are present here. An example command to set the necessary permissions for these directories is:

```
chmod -R 777 userdata tmp
```

If you wanted to limit write access to the webserver only you could alternatively do this (assuming the webserver runs as the user www and the group www):

```
chown -R www:www userdata tmp logs
chmod -R 700 userdata tmp logs
```

Register Globals

"register_globals" is a PHP option which makes all input variables available in the "global scope". This is bad due to potential security risks, and hence the reason why it is off by default in recent versions of PHP (4.1.0+). Whilst there should be no risk in this software (all variables set in this manner are unset), you are strongly recommended to turn it off. As such, there is an Apache .htaccess file in the htdocs directory that sets it to off. Of course if you don't allow .htaccess files it won't have any effect.

Session Files

It is important to make sure your session data cannot be read by anyone except the webserver. This is because sensitive information is stored in the session. If people can read your session files, then they can read this data. On Unix or Linux with Apache, if it runs as the user/group www/www you could run the following commands:

```
mkdir -p /tmp/session-data
chown -R www:www /tmp/session-data
chmod -R 700 /tmp/session-data
```

You would of course have to set this (/tmp/session-data) to be the directory used for session data in your php.ini. By default, the directory is set to /tmp.

Support and questions

When there are problems installing V-webmail and you need some assistance, then please try the online FAQs, or post your problems in the V-webmail forums and we will try to assist you.

http://www.v-webmail.org/faq.html
http://forums.v-webmail.org