Registry Installation - PHP

Versions

- For Registry v1.0.x and earlier, **PHP 5.2.8** or later is required.
    - ⚠ The GitHub Provisioning Plugin requires PHP 5.4. It may be necessary to delete this plugin to use an earlier PHP version.
    - PHP 7 is not supported.
- For Registry v2.0.0 through v4.0.2, **PHP 5.4** or later is required.
    - PHP 7 is supported. The use of PHP 7.0.10 or later is recommended, as earlier version have known bugs that may cause segmentation faults under Apache.
        - PHP 7.1 and 7.2 are supported starting with Registry v3.1.0.
        - PHP 8 is supported starting with Registry v4.0.2.
    - The Crowd Provisioning Plugin requires PHP 7.0 or later.
    - The LDAP Source Plugin requires PHP 5.6 or later.
    - The Password Authenticator Plugin requires PHP 7.0 or later (PHP 5.5 or later for Registry 3.1.x).
- For Registry v4.1.0 and later, **PHP 7** or later is required.
    - The GitHub Provisioning Plugin requires PHP 7.4.
- For Registry v5.0.0 and later, **PHP 8.0.2** or later will be required.

⚠ Regardless of the minimum version requirements, it is strongly recommended to use a version of PHP that is not EOL (currently 7.3 or later, though certain Linux distributions may provide security fixes for earlier versions).

ℹ Installations with more than a thousand or so CO Person records should consider using PHP 7, which features improved memory utilization.

Build Options

Depending on your requirements, you may need the following PHP options:

PHP Functionality	Required?	Build Option	Debian Package [1]	RHEL Package [2]
XSL	Yes	`--with-xsl`	php*-xsl	php-xsl
An appropriate database option	Yes	eg `--with-pgsql --with-pdo-pgsql` or `--with-mysql --with-pdo-mysql`[3]	php*-pgsql or php*-mysql	php-pgsql or php-mysql
OpenSSL	If COmanage will connect directly to an SMTP server to send mail, using SSL or TLS	`--with-openssl`	*Enabled by default*	php-openssl
mbstring	For multi-byte string support in data normalization	`--enable-mbstring`	*Enabled by default*	php-mbstring
LDAP	In order to use any LDAP based plugin (including the LDAP Provisioning Plugin, LDAP Identifier Validator Plugin, or LDAP Source Plugin)	`--with-ldap`	php*-ldap	php-ldap
GD2	To make use of QR codes exposed for enrollment flow starting points, the GD2 image library is required	--with-gd	php*-gd	php-gd

Notes

[1]Debian packages use the PHP version number (eg: 5 or 7.0) in place of the *.

[2]Not all packages are available in the default repos.

[3]Both regular and PDO are required, with the former used by ADOdB and the latter used by CakePHP.

PHP 5.4.0 Enables Strict Error Reporting

As of PHP 5.4.0, PHP ships by default with strict logging enabled. As of CakePHP 2.0.5, this will cause failures during setup. This can be disabled in `php.ini` by setting

```
error_reporting = E_ALL & ~E_STRICT
```

PCRE Bug May Cause Problems

There are known issues with earlier versions of the PCRE library that will cause COmanage Registry to be unable to set up its database tables. Version 6.6 and earlier are known to have problems, while versions 8.02 and later are known to work. You can check the version that PHP was built against by running this command:

`php -r 'phpinfo();' | grep PCRE`

If you are using an old version of PCRE, you'll first need to install a more recent version. Be sure to configure it with the `--enable-utf8 --enable-unicode-properties` flags. You'll then need to rebuild PHP against the newer version of the PCRE library.

Alternately, you may be able to rebuild PHP using its own internal copy of PCRE. As of PHP v5.3.3, PCRE 8.02 or later is included by default.

The database setup step may throw errors like `Error: Database connection "Mysql" is missing, or could not be created.` if the command line php client can't find the PDO libraries. One way to ensure their availability is to build PHP `--with-pdo-mysql` (or with the appropriate flag for the database you are using).

You can test for this with `php -r 'phpinfo();' | grep pdo`:

```
$ php -r 'phpinfo();' | grep pdo
pdo_mysql
pdo_mysql.default_socket => /var/lib/mysql/mysql.sock => /var/lib/mysql/mysql.sock
pdo_sqlite
```

If the PDO libraries show correctly but you are still getting this error, you may be trying to connect via the default socket (ie: "localhost") instead of TCP (ie: "127.0.0.1"), and the PDO libraries may not be able to find the default socket.

Testing PHP Database Connectivity

PHP will work with many different database servers. You may wish to test that PHP was built with support for MySQL or PostgreSQL (or whatever database you are using). You should have already installed and configured the database server and (in this example) have created a user named `registry_user`.

To test if PHP was built with support for MySQL create the file `mysql-test.php` with contents

```
<?php
mysql_connect("localhost", "registry_user", "a password goes here") or die(mysql_error());
echo "Connected to MySQL<br/>";
?>
```

To test if PHP was built with support for PostgreSQL create the file `postgreSQL-test.php` with contents

```
<?php
pg_connect("host=localhost dbname=test user=registry_user password=password") or die(pg_last_error());
echo "Connected to PostgreSQL<br/>";
?>
```

Either run the command line php tool on the file or serve it from your webserver and make sure that the script can connect to your database server.

Be sure to remove your test file after testing so it is not exposed on your web server.

Configuration Suggestions

As for all production PHP installations, `display_errors` should be set to `Off` in your php.ini.

Memory Considerations

Larger deployments may need to consider the amount of memory, in particular for Bulk Initial Data Loading. Error messages such as

```
2021-12-01 23:13:39 Notice: Undefined index: status in [/srv/comanage-registry/app/Console/Command
/BulkLoadShell.php, line 257]
Fatal Error Error: Allowed memory size of 134217728 bytes exhausted (tried to allocate 4096 bytes) in [/srv
/comanage-registry/lib/Cake/Console/ConsoleOutput.php, line 239]
```

are indicative of this situation. To increase the allowed memory, set `memory_limit` in your php.ini to a number larger than the size in the error.

⚠ This error can also be indicative of a bug in the code. If generously increasing the memory limit does not solve the problem, please file a bug report.

Next: Registry Installation - Web Server