How To Add PHP YAML Extension Using The PECL Tool

Summary

This is an example recipe, which can be used as-is if needed, for adding PECL extensions to Zend Server, using Linux (CentOS) build tools and the Zend Server PHP tools (Sources, PECL, PHP Config, PHPIZE etc.).

The resulting module (PHP extension yaml.so) is **binary compatible** with matching RHEL version (7.9 in this example), so you can build on a CentOS machine and copy the PHP module to another CentOS / RHEL machine.

You will also need to **install any system LIBRARY** required by the PHP module(s) you add with the PECL tool - in this case you will need '**libyaml**' package from the distro (see the LDD command on step 4 to verify dependencies).

You can of course modify the instructions for other Zend Server and PHP version combinations.

If you need to build a Debian / Ubuntu module, please be aware that the Zend and Linux package names may vary, but the same procedure applies.

Adding PECL-YAML to Zend Server 2019.0.7 PHP 7.3.27 under CentOS 7.9

1. Add Linux and Zend packages for building the PHP YAML module from PECL sources:
   ```
   # yum install gcc autoconf libyaml libyaml-devel php-dev-zend-server
   ```
2. IMPORTANT: Add /usr/local/zend/bin to your $PATH, to use Zend Server binaries (PECL, PHP)
3. Run PECL commands. When prompted for libyaml location, click enter for autodetect.
   ```
   # pecl channel-update pecl.php.net
   # pecl install yaml
   ```
4. Check module dependencies:
   ```
   # ldd /usr/local/zend/php/7.3/lib/ext/yaml.so
   ```
5. Add ini for loading YAML into PHP:
   ```
   # echo "extension=yaml.so" > /usr/local/zend/php/7.3/etc/conf.d/yaml.ini
   ```
6. Check PHP module loaded in CLI:
   ```
   # php -m | grep yaml
   ```
7. Verify that yaml_parse (and other YAML functions) is added:
   ```
   # php -r 'print_r(get_extension_funcs("yaml"));'
   ```
8. Restart web server to load YAML into your Application PHP.
9. Check YAML functionality in your app (run unit test, load web page etc.).

Notes:

Installing Linux packages requires elevated credentials (root / sudo) - see the commands above with a preceding hash (#) suggesting that.

The commands in steps 6+7 can also be executed without root / sudo (regular user with /usr/local/zend/bin in the $PATH).

You will NOT see YAML in the PHP extensions under the Admin UI, as it is not in the blueprint (it is not in the original list of parsed modules).

You should see YAML section in the phpinfo() output.

Related articles

How to compile Imagick PHP extension with WebP support