

# HowTo: PHP Extensions Compilation for Zend Server

## Applies to

Zend Server 6.x or later  
GNU / Linux

## Summary

Zend Server includes [most of the most popular PHP extensions](#). Of course, in some cases our users may need additional extensions for their applications. This article has two examples of PHP extensions compilation for Zend Server - the YAML extension and the SNMP extension.

There is also a more detailed explanation of the process in on-line documentation:

[UNIX: Compiling PHP Extensions](#)



### Disclaimer:

Zend can provide only limited support for extensions compiled in such way.

## Preparing the Build Environment

The machine where the extensions are to be compiled, obviously, must have Zend Server installed. We will also need the PHP source packages for the corresponding PHP version (5.6 in all examples below) and the following common build tools: automake, autoconf, libtool, make and gcc.

RHEL / CentOS / OEL:

```
# yum install php-5.6-source-zend-server autoconf automake libtool make gcc
```

Ubuntu / Debian:

```
# aptitude install php-5.6-source-zend-server autoconf automake libtool make gcc
```

### Note:

Make sure to change the PHP version accordingly in the above commands. For instance, if you're using Zend Server 9.1.x, you should install the package `php-7.1-source-zend-server`

The system is now ready.

## Compiling a PECL Extension - YAML

First, we need to install the dependencies for this specific PHP extension - the 'libyaml' package and the corresponding development package:

RHEL / CentOS / OEL:

```
# yum install libyaml-devel libyaml
```

Ubuntu / Debian:

```
# aptitude install libyaml-dev libyaml-0-2
```

### Note:

Obviously, each extension has its own set of dependencies and prerequisites - make sure that you carefully read the text of error and warning messages that may be displayed during configuration and compilation. In some cases you may need to perform additional internet research to satisfy such dependencies and prerequisites.

After the package installation is finished, we can compile the extension. First, we should verify that the extension can be obtained from [PECL](#):

```
$ /usr/local/zend/bin/pecl search yaml
```

Once confirmed, we can do the actual configuration and compilation:

```
# /usr/local/zend/bin/pecl install yaml
```

The 'pecl' script will also place the compiled extension into the correct location:

```
$ ls -l /usr/local/zend/lib/php_extensions/yaml.so
```

## Compiling a Built-In PHP Extension - SNMP

The first step is the same - dependencies installation:

RHEL / CentOS / OEL:

```
# yum install net-snmp-devel net-snmp
```

Ubuntu / Debian:

```
# aptitude install libsnmp-dev snmp
```

### Note:

Obviously, each extension has its own set of dependencies and prerequisites - make sure that you carefully read the text of error and warning messages that may be displayed during configuration and compilation. In some cases you may need to perform additional internet research to satisfy such dependencies and prerequisites.

The extension source code is installed in the 'share' directory of Zend Server's installation tree. We need to make this our working directory:

RHEL / CentOS / OEL:

```
# cd /usr/local/zend/share/php-source/ext/snmp/
```

Ubuntu / Debian (path is different for different PHP versions):

```
# cd /usr/local/zend/share/php-source/php-5.6.15/ext/snmp/
```

To create the configuration that will match Zend Server, we need to execute 'phpize' in the source code directory:

```
# /usr/local/zend/bin/phpize
```

Then we use the generated configuration file with the 'configure' script (in the same source code directory):

```
# ./configure --with-php-config=/usr/local/zend/bin/php-config
```

Finally, we can build the extension and move the ready binary to the correct location:

```
# make  
# make install
```

To verify:

```
$ ls -l /usr/local/zend/lib/php_extensions/snmp.so
```

## Post-Compilation

To be able to use the compiled extensions, we need to modify PHP configuration (load the extensions):

```
# echo "extension=yaml.so" > /usr/local/zend/etc/conf.d/yaml.ini  
# echo "extension=snmp.so" > /usr/local/zend/etc/conf.d/snmp.ini
```

And to be able to manage these extensions from Zend Server UI, we need to make sure that the configuration files' ownership is correct:

```
# chown zend:zend /usr/local/zend/etc/conf.d/yaml.ini  
# chown zend:zend /usr/local/zend/etc/conf.d/snmp.ini
```

All that's left is Zend Server restart:

```
# /usr/local/zend/bin/zendctl.sh restart
```

After the restart you should be able to see the new extensions in 'phpinfo()' output and use these extensions' functions in your code.