

HowTo: Implement TLS 1.2 Support with the cURL PHP Extension in Zend Server

Applies To

Zend Server 6.3, 7

GNU / Linux, OS X

Issue

Older Zend Server versions doesn't support TLS 1.2 with the cURL PHP extension.

Technical

We ship OpenSSL 0.9.8 with older Zend Server releases and hence no support for TLS 1.2. Also it is not possible to compile cURL against the latest OpenSSL (1.0.1 or 1.0.2) for TLS 1.2 support -everything would need to be re-compiled and it would be a very complex procedure, of course. We ship latest **OpenSSL 1.0.2 (hence TLS 1.2)** with the **latest** Zend Server in all compatible Linux versions. Ideally, you should consider upgrading to PHP 7 and Zend Server 9.0 (or later).

Workaround

To implement TLS 1.2 support with the cURL PHP extension in Zend Server, Instead of OpenSSL, you can easily compile cURL against another TLS implementation - Network Security Services (NSS). To (re)compile cURL against NSS, execute the following commands in the terminal as 'root' or using 'sudo':

Note: If you are using Zend Server 7 with PHP 5.4, replace *php-5.3-source-zend-server* with *php-5.4-source-zend-server* in the YUM command.

```
# cp /usr/local/zend/lib/php_extensions/curl.so /usr/local/zend/lib/php_extensions/curl.so.BAK
# yum install php-5.3-source-zend-server autoconf automake libtool make gcc libcurl-devel cyrus-sasl-devel openssl-devel
# cd /usr/local/zend/share/php-source/ext/curl/
# /usr/local/zend/bin/phpize
# ./configure --with-php-config=/usr/local/zend/bin/php-config
# make
# make install
# /usr/local/zend/bin/zendctl.sh restart
```

You can then verify NSS implementation with cURL using *phpinfo()*.

For more details please see this Knowledge Base article - [HowTo: PHP Extensions Compilation for Zend Server](#).

After the compilation you should see that cURL is working with TLS greater than 1.0. Simple test code:



Unknown macro: 'divbox'

Note

In case the workaround doesn't work, please try to update your Operating System to the newer version.