Xdebug: Documentation

XDEBUG EXTENSION FOR PHP | DOCUMENTATION
» Documentation for: **Xdebug 2**
» Feature: **Installation**

This section describes on how to install Xdebug.

Precompiled Windows Modules

There are a few precompiled modules for Windows, they are all for the non-debug version of PHP. You can get those at the download [1] page. Follow these instructions [2] to get Xdebug installed.

PECL Installation

As of Xdebug 0.9.0 you can install Xdebug through PEAR/PECL. This only works with with PEAR version 0.9.1-dev or higher and some UNIX.

Installing with PEAR/PECL is as easy as:

```
# pecl install xdebug
```

but you still need to add the correct line to your php.ini: (don't forget to change the path and filename to the correct one — but make sure you use the **full path**)

```
zend_extension="/usr/local/php/modules/xdebug.so"
```

Note: You should ignore any prompts to add "extension=xdebug.so" to php.ini — this **will** cause problems.

Installation From Source

You can download the source of the latest **stable** release 2.2.3. Alternatively you can obtain Xdebug from GIT:

```
git clone git://github.com/derickr/xdebug.git
```

This will checkout the latest development version which is currently 2.2.3. You can also browse the source at https://github.com/derickr/xdebug [3].

Compiling

There is a wizard [4] available that provides you with the correct file to download, and which paths to use.

You compile Xdebug separately from the rest of PHP. Note, however, that you need access to the scripts 'phpize' and 'php-config'. If your system does not have 'phpize' and 'php-config', you will need to compile and install PHP from a source tarball first, as these script are by-products of the PHP compilation and installation processes. (Debian users can install the required tools with apt-get install php5-dev). It is important that the source version matches the installed version as there are slight, but important, differences between PHP versions. Once you have access to 'phpize' and 'php-config', do the following:

1. Unpack the tarball: tar -xzf xdebug-2.2.3.tgz. Note that you do not need to unpack the tarball inside the PHP source code tree. Xdebug is compiled separately, all by itself, as stated above.
2. cd xdebug-2.2.3
3. Run phpize: phpize (or /path/to/phpize if phpize is not in your path). Make sure you use the phpize that belongs to the PHP version that you want to use Xdebug with. See this FAQ entry if you're having some issues with finding which phpize to use.
4. ./configure --enable-xdebug
5. make install

Configure PHP to Use Xdebug

1. add the following line to php.ini: zend_extension="/wherever/you/put/it/xdebug.so" (for non-threaded use of PHP, for example the CLI, CGI or Apache 1.3 module) or: zend_extension_ts="/wherever/you/put/it/xdebug.so" (for threaded usage of PHP, for example the Apache 2 work MPM or the the ISAPI module). **Note:** In case you compiled PHP yourself and used --enable-debug you would have to use zend_extension_debug=. **From PHP 5.3 onwards, you always need to**

use the zend_extension PHP.ini setting name, and not zend_extension_ts, nor zend_extension_debug. However, your compile options (ZTS/normal build; debug/non-debug) still need to match with what PHP is using.

2. Restart your webserver.
3. Write a PHP page that calls '*phpinfo()*' Load it in a browser and look for the info on the Xdebug module. If you see it next to the Zend logo, you have been successful! You can also use 'php -m' if you have a command line version of PHP, it lists all loaded modules. Xdebug should appear twice there (once under 'PHP Modules' and once under 'Zend Modules').

Compatibility

Xdebug does not work together with the Zend Optimizer or any other extension that deals with PHP's internals (DBG, APD, ioncube etc). This is due to compatibility problems with those modules.

Debugclient Installation

Unpack the Xdebug source tarball and issue the following commands:

```
$ cd debugclient
$ ./configure --with-libedit
$ make
# make install
```

This will install the debugclient binary in /usr/local/bin unless you don't have libedit installed on your system. You can either install it, or leave out the '--with-libedit' option to configure. Debian 'unstable' users can install the library with `apt-get install libedit-dev libedit2`.

If the configure script can not find libedit and you are sure you have (and it's headers) installed correctly and you get link errors like the following in your configure.log:

```
/usr/lib64/libedit.so: undefined reference to `tgetnum'
/usr/lib64/libedit.so: undefined reference to `tgoto'
/usr/lib64/libedit.so: undefined reference to `tgetflag'
/usr/lib64/libedit.so: undefined reference to `tputs'
/usr/lib64/libedit.so: undefined reference to `tgetent'
```

```
/usr/lib64/libedit.so: undefined reference to `tgetstr'
collect2: ld returned 1 exit status
```

you need to change your configure command to:

```
$ LDFLAGS=-lncurses ./configure --with-libedit
```

---

1. http://xdebug.org/download.php
2. http://xdebug.org/wizard.php
3. https://github.com/derickr/xdebug
4. http://xdebug.org/wizard.php