

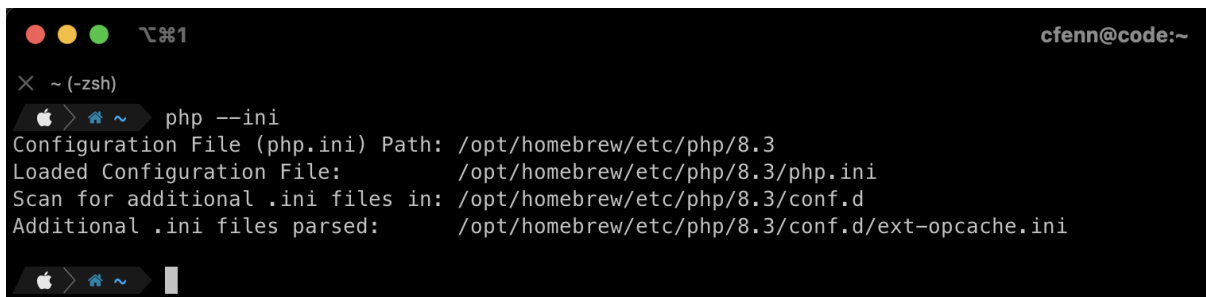
Installing and using Xdebug will allow for much more efficient and effective software development in the world of PHP.

Follow the install instructions on the following page: <https://xdebug.org/docs/install>

Background information:

The **php.ini** file is where extensions can be managed. In this case we are configuring PHP to have Xdebug loaded, but we will see later in the course that this is also where we manage database driver extensions that act as a bridge between PHP and a database such as MySQL or Postgres.

You can find your php.ini file by running **php --ini** and looking at the “Loaded Configuration File” line:



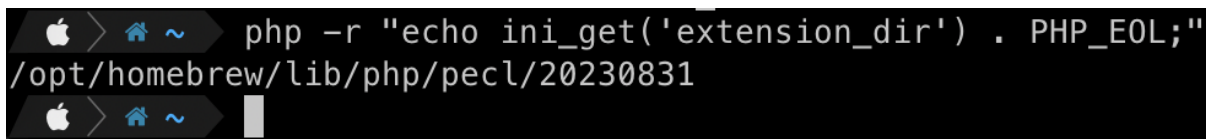
```
cfenn@code:~  
X ~ (-zsh)  
~ > php --ini  
Configuration File (php.ini) Path: /opt/homebrew/etc/php/8.3  
Loaded Configuration File:      /opt/homebrew/etc/php/8.3/php.ini  
Scan for additional .ini files in: /opt/homebrew/etc/php/8.3/conf.d  
Additional .ini files parsed:    /opt/homebrew/etc/php/8.3/conf.d/ext-opcache.ini
```

In the above screenshot my php.ini file is at **/opt/homebrew/etc/php/8.3/php.ini**, but your file will likely be at a different location depending on your OS, how PHP has been installed, and the version of PHP in use.

Finding your PHP extensions directory:

Run: **php -r "echo ini_get('extension_dir') . PHP_EOL;"**

For example:



```
~ > php -r "echo ini_get('extension_dir') . PHP_EOL;"  
/opt/homebrew/lib/php/pecl/20230831
```

This is the directory where your Xdebug executable file will need to be (the directory will be different on your machine).

If you’ve installed Xdebug using PECL on MacOS or Linux, you likely won’t need to manually move the file. It’ll automatically put it into the right folder for you. If you’re on Windows, manually move the Xdebug .dll file to the directory that the above command gives you.

php.ini modifications:

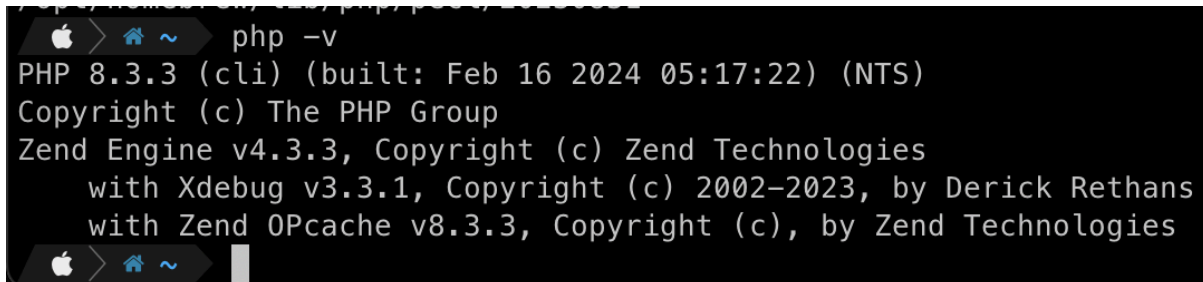
At the bottom of my php.ini file I've added:

```
[xdebug]
zend_extension=xdebug
xdebug.mode=debug
xdebug.client_host=localhost
xdebug.client_port=9003
xdebug.discover_client_host=true
xdebug.start_with_request=yes
xdebug.log_level=0
```

Note #1 that the client_port is the port which Xdebug will use. It is not the port that your PHP application needs to be served from.

Note #2 that the zend_extension can simply be "xdebug" because PHP will look for the extension in the folder that the **extension_dir** value is. You could also put the fully qualified path to the Xdebug file here if you wish.

At this stage you can test that PHP has Xdebug installed by running **php -v**.



```
Apple > php -v
PHP 8.3.3 (cli) (built: Feb 16 2024 05:17:22) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.3, Copyright (c) Zend Technologies
    with Xdebug v3.3.1, Copyright (c) 2002-2023, by Derick Rethans
    with Zend OPcache v8.3.3, Copyright (c), by Zend Technologies
Apple >
```

The "with Xdebug..." line confirms things are looking good!

Common issues:

1. If you see a message such as "Cannot load Xdebug - it was already loaded", open up your php.ini file and do a ctrl+f search for "xdebug" you likely have two "zend_extension=xdebug" lines. During the Xdebug
2. "Cannot load Xdebug" if you are on Windows you will need to download one of the executables from the downloads page. See: <https://xdebug.org/download>. In order to select the correct Xdebug executable file, run **php -v**: if it prints out "NTS" (meaning "Not Thread Safe") as part of the output, ensure you select an executable without "TS" (meaning "Thread Safe") in the file name.