If you find Xdebug useful, please consider **supporting the project**.

# Installation Wizard

## Summary

- **Xdebug installed:** no
- **Server API:** Apache 2.0 Handler
- **Windows:** no
- **Zend Server:** no
- **PHP Version:** 8.2.4
- **Zend API nr:** 420220829
- **PHP API nr:** 20220829
- **Debug Build:** no
- **Thread Safe Build:** no
- **OPcache Loaded:** no
- **Configuration File Path:** /Applications/XAMPP/xamppfiles/etc
- **Configuration File:** /Applications/XAMPP/xamppfiles/etc/php.ini
- **Extensions directory:** /Applications/XAMPP/xamppfiles/lib/php/extensions/no-debug-non-zts-20220829

## Instructions

1. Download xdebug-3.3.2.tgz
2. Install the pre-requisites for compiling PHP extensions.
   On your Mac, we only support installations with 'homebrew', and `brew install php && brew install autoconf` should pull in the right packages.
3. Unpack the downloaded file with `tar -xvzf xdebug-3.3.2.tgz`
4. Run: `cd xdebug-3.3.2`
5. Run: `phpize` (See the **FAQ** if you don't have `phpize`).

   As part of its output it should show:

   ```
   Configuring for:
   ...
   Zend Module Api No:      20220829
   Zend Extension Api No:   420220829
   ```

   If it does not, you are using the wrong `phpize`. Please follow **this FAQ entry** and skip the next step.

6. Run: `./configure`
7. Run: `make`
8. Run: `mkdir -p /Applications/XAMPP/xamppfiles/lib/php/extensions/no-debug-non-zts-20220829`
9. Run: `cp modules/xdebug.so /Applications/XAMPP/xamppfiles/lib/php/extensions/no-debug-non-zts-20220829/`
10. Update `/Applications/XAMPP/xamppfiles/etc/php.ini` and add the line:
    `zend_extension = xdebug`
11. Restart the Apache Webserver

## Enabling Features

Now Xdebug is installed, you can enable its features. Please refer to the dedicated sections in the documentation about information on how to enable and configure these Xdebug features. Where these sections refer to `php.ini` or similar, please remember to use `/Applications/XAMPP/xamppfiles/etc/php.ini`:

- **Development Helpers** — help you get better error messages and obtain better information from PHP's built-in functions.
- **Step Debugging** — allows you to interactively walk through your code to debug control flow and examine data structures.
- **Profiling** — allows you to find bottlenecks in your script and visualize those with an external tool.