Activate and Use Xdebug

Xdebug is a powerful open source debugger and profiler for PHP. It is included with XAMPP and can be used to display stack traces, analyze code coverage and profile your PHP code.

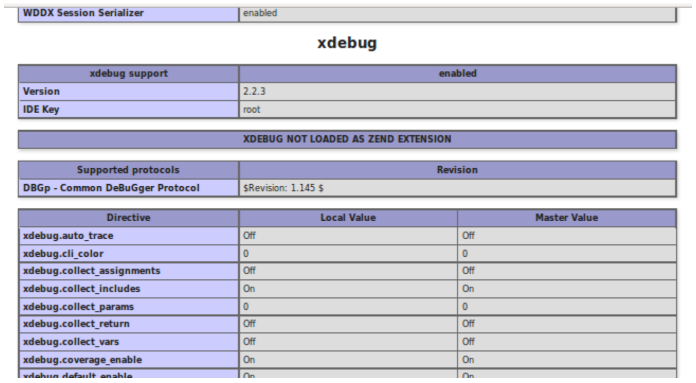
To activate Xdebug, follow these steps:

1. Edit the *php.ini* file in the *etc/* subdirectory of your XAMPP installation directory (usually, */opt/lampp*). Within this file, activate the Xdebug extension by adding the following line to it:

   ```
   extension = xdebug.so
   ```

2. Restart the Apache server using the XAMPP control panel.

Xdebug should now be active. To verify this, browse to the URL http://localhost/xampp/phpinfo.php, which displays the output of the *phpinfo()* command. Look through the script and verify that the Xdebug extension is now active.



WDDX Session Serializer	enabled	

xdebug

xdebug support	enabled
Version	2.2.3
IDE Key	root

XDEBUG NOT LOADED AS ZEND EXTENSION

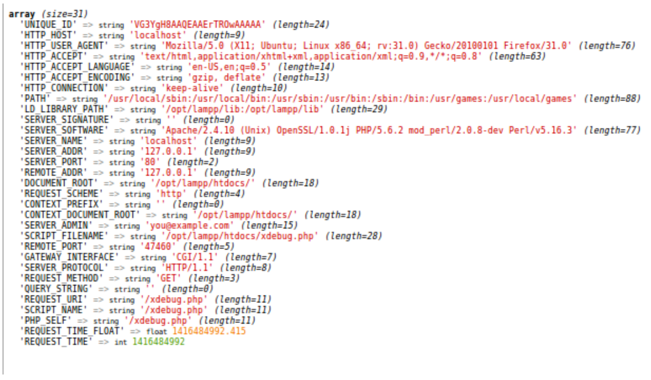
Supported protocols	Revision
DBGp - Common DeBuGger Protocol	$Revision: 1.145 $

Directive	Local Value	Master Value
xdebug.auto_trace	Off	Off
xdebug.cli_color	0	0
xdebug.collect_assignments	Off	Off
xdebug.collect_includes	On	On
xdebug.collect_params	0	0
xdebug.collect_return	Off	Off
xdebug.collect_vars	Off	Off
xdebug.coverage_enable	On	On
xdebug.default_enable	On	On

Xdebug overloads the default *var_dump()* function with its own version that includes (among other things) color coding for different PHP types, so you can see it in action immediately by using the *var_dump()* function in a PHP script. For example, create a simple PHP script in the *htdocs/* subdirectory of your XAMPP installation directory with the following content:

```php
<?php
var_dump($_SERVER);
```

When you view your script through a browser, here's an example of what you might see:



```
array (size=31)
  'UNIQUE_ID' => string 'VG3YgH8AAQEAAErTROwAAAAA' (length=24)
  'HTTP_HOST' => string 'localhost' (length=9)
  'HTTP_USER_AGENT' => string 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0' (length=76)
  'HTTP_ACCEPT' => string 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8' (length=63)
  'HTTP_ACCEPT_LANGUAGE' => string 'en-US,en;q=0.5' (length=14)
  'HTTP_ACCEPT_ENCODING' => string 'gzip, deflate' (length=13)
  'HTTP_CONNECTION' => string 'keep-alive' (length=10)
  'PATH' => string '/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games' (length=88)
  'LD_LIBRARY_PATH' => string '/opt/lampp/lib:/opt/lampp/lib' (length=29)
  'SERVER_SIGNATURE' => string '' (length=0)
  'SERVER_SOFTWARE' => string 'Apache/2.4.10 (Unix) OpenSSL/1.0.1j PHP/5.6.2 mod_perl/2.0.8-dev Perl/v5.16.3' (length=77)
  'SERVER_NAME' => string 'localhost' (length=9)
  'SERVER_ADDR' => string '127.0.0.1' (length=9)
  'SERVER_PORT' => string '80' (length=2)
  'REMOTE_ADDR' => string '127.0.0.1' (length=9)
  'DOCUMENT_ROOT' => string '/opt/lampp/htdocs/' (length=18)
  'REQUEST_SCHEME' => string 'http' (length=4)
  'CONTEXT_PREFIX' => string '' (length=0)
  'CONTEXT_DOCUMENT_ROOT' => string '/opt/lampp/htdocs/' (length=18)
  'SERVER_ADMIN' => string 'you@example.com' (length=15)
  'SCRIPT_FILENAME' => string '/opt/lampp/htdocs/xdebug.php' (length=28)
  'REMOTE_PORT' => string '47460' (length=5)
  'GATEWAY_INTERFACE' => string 'CGI/1.1' (length=7)
  'SERVER_PROTOCOL' => string 'HTTP/1.1' (length=8)
  'REQUEST_METHOD' => string 'GET' (length=3)
  'QUERY_STRING' => string '' (length=0)
  'REQUEST_URI' => string '/xdebug.php' (length=11)
  'SCRIPT_NAME' => string '/xdebug.php' (length=11)
  'PHP_SELF' => string '/xdebug.php' (length=11)
  'REQUEST_TIME_FLOAT' => float 1416484992.415
  'REQUEST_TIME' => int 1416484992
```

One of Xdebug's most powerful features is its ability to profile a PHP script and produce detailed statistics on how long each function call or line of code takes to execute. This can be very useful for performance analysis of complex scripts. To turn on script profiling, follow these steps:

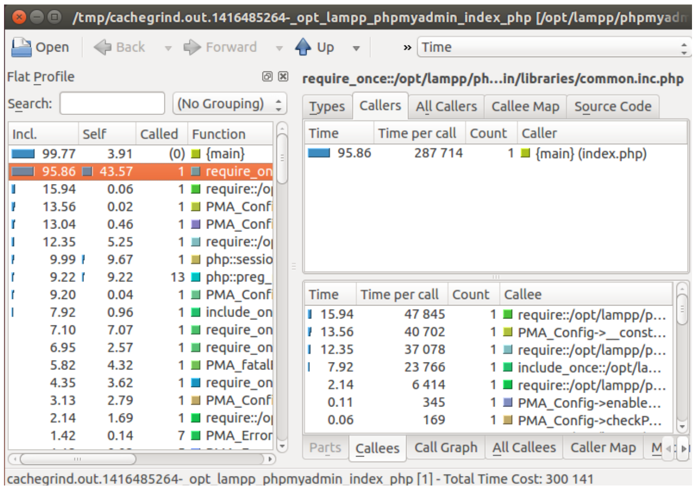
1. Edit the *php.ini* file in the *etc/* subdirectory of your XAMPP installation directory. Within this file, add the following section:

```
[XDebug]
xdebug.profiler_append = 0
xdebug.profiler_enable = 1
xdebug.profiler_enable_trigger = 0
xdebug.profiler_output_dir = "/tmp"
xdebug.profiler_output_name = "cachegrind.out.%t-%s"
```

2. Restart the Apache server using the XAMPP control panel.

At this point, XDebug profiling is active. Every PHP script that you run will be profiled and the results will be placed in the */tmp/* directory as a so-called cachegrind file. You can view this cachegrind file with a tool like KCachegrind, which you must download and install separately.

To illustrate how this works, consider the screenshot below, which shows the profiled output of a script using KCachegrind. As the screenshot illustrates, it's easy to see the entire life cycle of a PHP script, including the call sequence and the amount of time taken by each function, and thereby find targets for further optimization.



/tmp/cachegrind.out.1416485264-_opt_lampp_phpmyadmin_index_php [/opt/lampp/phpmyad

require_once::/opt/lampp/ph...in/libraries/common.inc.php

Incl.	Self	Called	Function
99.77	3.91	(0)	{main}
95.86	43.57	1	require_on
15.94	0.06	1	require::/o
13.56	0.02	1	PMA_Confi
13.04	0.46	1	PMA_Confi
12.35	5.25	1	require::/o
9.99	9.67	1	php::sessio
9.22	9.22	13	php::preg_
9.20	0.04	1	PMA_Confi
7.92	0.96	1	include_on
7.10	7.07	1	require_on
6.95	2.57	1	require_on
5.82	4.32	1	PMA_fatal
4.35	3.62	1	require_on
3.13	2.79	1	PMA_Confi
2.14	1.69	1	require::/o
1.42	0.14	7	PMA_Error

Time	Time per call	Count	Caller
95.86	287 714	1	{main} (index.php)

Time	Time per call	Count	Callee
15.94	47 845	1	require::/opt/lampp/p...
13.56	40 702	1	PMA_Config->__const...
12.35	37 078	1	require::/opt/lampp/p...
7.92	23 766	1	include_once::/opt/la...
2.14	6 414	1	require::/opt/lampp/p...
0.11	345	1	PMA_Config->enable...
0.06	169	1	PMA_Config->checkP...

cachegrind.out.1416485264-_opt_lampp_phpmyadmin_index_php [1] - Total Time Cost: 300 141

TIP To find out more about Xdebug's powerful features, read the Xdebug documentation.