

Xdebug 3.0 Updates from the Author

php[world]

Derick Rethans

derick@xdebug.org—[@derickr](https://twitter.com/derickr)

<https://derickrethans.nl/talks/xdebug-phpworld19>

<https://joind.in/26837>

About Me

Derick Rethans

- I'm European, living in London
- **PHP 7.4 Release Manager**
- **Xdebug** & PHP's Date/Time support
- Host of <https://phpinternals.news> podcast
- I ♥🌐 maps, I ♥🍺 beer, I ♥🥃 whisky
- twitter: @derickr



The Xdebug Experience

Xdebug is 17 years old, and has plenty of issues:

- **Configuration Confusion**

65 configuration settings: pointless, duplicated, or incompatible at the same time

- **Unproductive Usability**

port 9000, missing breakpoints, education (networking)

- **Poor Performance**

- **Lagging Behind**

Xdebug for PHP 7.3 was 4 months late

Configuration Settings

```
xdebug.auto_trace
xdebug.cli_color
xdebug.collect_assignments
xdebug.collect_includes
xdebug.collect_params
xdebug.collect_return
xdebug.collect_vars
xdebug.coverage_enable
xdebug.default_enable
xdebug.dump.COOKIE
xdebug.dump.ENV
xdebug.dump.FILES
xdebug.dump.GET
xdebug.dump.POST
xdebug.dump.REQUEST
xdebug.dump.SERVER
xdebug.dump.SESSION
xdebug.dump_globals
xdebug.dump_once
xdebug.dump_undefined
xdebug.extended_info
xdebug.file_link_format
xdebug.filename_format
xdebug.force_display_errors
xdebug.force_error_reporting
xdebug.gc_stats_enable
xdebug.gc_stats_output_dir
xdebug.gc_stats_output_name
xdebug.halt_level
xdebug.idekey
xdebug.max_nesting_level
xdebug.max_stack_frames
xdebug.overload_var_dump
```

```
xdebug.profiler_aggregate
xdebug.profiler_append
xdebug.profiler_enable
xdebug.profiler_enable_trigger
xdebug.profiler_enable_trigger_value
xdebug.profiler_output_dir
xdebug.profiler_output_name
xdebug.remote_addr_header
xdebug.remote_autostart
xdebug.remote_connect_back
xdebug.remote_cookie_expire_time
xdebug.remote_enable
xdebug.remote_handler
xdebug.remote_host
xdebug.remote_log
xdebug.remote_mode
xdebug.remote_port
xdebug.remote_timeout
xdebug.scream
xdebug.show_error_trace
xdebug.show_exception_trace
xdebug.show_local_vars
xdebug.show_mem_delta
xdebug.trace_enable_trigger
xdebug.trace_enable_trigger_value
xdebug.trace_format
xdebug.trace_options
xdebug.trace_output_dir
xdebug.trace_output_name
xdebug.var_display_max_children
xdebug.var_display_max_data
xdebug.var_display_max_depth
```

Missed Breakpoints

```
1 <?php
2 function demo($a, $b, $c, $d)
3 {
4     if ( // breakpoint here
5         $a
6         && $b
7         && ($c || $d)
8     ) {
9         echo "It's true!";
10    }
11 }
12
13 demo(true, true, true, false);
```

demo

Missed Breakpoints

```

1 <?php
2 function demo($a, $b, $c, $d)
3 {
4     if ( // breakpoint here
5         $a
6         && $b
7         && ($c || $d)
8     ) {
9         echo "It's true!";
10    }
11 }
12
13 demo(true, true, true, false);

```

```

function name: demo
number of ops: 14
compiled vars: !0 = $a, !1 = $b, !2 = $c, !3 = $d
line   #*  E I 0 op          return operands
  2     0  E >  EXT_NOP
  3     1          RECV          !0
  4     2          RECV          !1
  5     3          RECV          !2
  6     4          RECV          !3
  7     5          EXT_STMT
  8     6          > JMPZ          !0, ->12
  9     7          > > JMPZ          !1, ->12
 10    8          > > JMPNZ         !2, ->10
 11    9          > > JMPZ          !3, ->12
 12   10          >  EXT_STMT
 13   11          ECHO          'It27s+true21'
 14  12          >  EXT_STMT
 15  13          >  RETURN          null

```

Missed Breakpoints

```

1 <?php
2 function demo($a, $b, $c, $d)
3 {
4     if ( // breakpoint here
5         $a
6         && $b
7         && ($c || $d)
8     ) {
9         echo "It's true!";
10    }
11 }
12
13 demo(true, true, true, false);

```

```

function name: demo
number of ops: 14
compiled vars: !0 = $a, !1 = $b, !2 = $c, !3 = $d
line   #*  E I 0 op          return operands
  2     0  E >  EXT_NOP
  3     1          RECV          !0
  4     2          RECV          !1
  5     3          RECV          !2
  6     4          RECV          !3
  7     5          EXT_STMT
  8     6          > JMPZ          !0, ->12
  9     7          > > JMPZ          !1, ->12
 10    8          > > JMPNZ         !2, ->10
 11    9          > > JMPZ          !3, ->12
 12   10          >  EXT_STMT
 13   11          ECHO          'It27s+true21'
 14  12          >  EXT_STMT
 15  13          >  RETURN          null

```

Copy-cats

9000	Yes		SonarQube Web Server ^[281]	Unofficial
	Yes		DBGp	Unofficial
	Yes		SqueezeCenter web server & streaming	Unofficial
		Yes	UDPCast	Unofficial
	Yes		Play! Framework web server ^[282]	Unofficial
	Yes		Hadoop NameNode default port	Unofficial
	Yes		PHP-FPM default port	Unofficial
	Yes		QBittorrent 's embedded torrent tracker default port ^[283]	Unofficial

PHP-FPM picked the same port as Xdebug (DBGp) 😞

Activating the Step Debugger

php.ini settings:

```
xdebug.remote_enable=1
```

```
xdebug.remote_host=localhost
```

```
xdebug.remote_port=9000
```

Activating the Step Debugger

php.ini settings:

```
xdebug.remote_enable=1
```

```
xdebug.remote_host=localhost
```

```
xdebug.remote_port=9000
```

```
xdebug.remote_connect_back=1
```

Activating the Step Debugger

On the shell:

```
export XDEBUG_CONFIG="idekey=xdebugrocks"
```

Activating the Step Debugger

On the shell:

```
export XDEBUG_CONFIG="idekey=xdebugrocks"
```

Step Debugger Logs

Logging debugging communication:

```
xdebug.remote_log=/tmp/remote.log
```

Also shows connection attempts and failures:

```
xdebug.remote_connect_back=0  
xdebug.remote_port=9991
```

```
I: Connecting to configured address/port: localhost:9991.  
I: Connected to client. :-)
```

Step Debugger Logs

Logging debugging communication:

```
xdebug.remote_log=/tmp/remote.log
```

Also shows connection attempts and failures:

```
xdebug.remote_connect_back=0  
xdebug.remote_port=9991
```

```
I: Connecting to configured address/port: localhost:9991.  
I: Connected to client. :-)
```

```
xdebug.remote_connect_back=1  
xdebug.remote_host=SomethingEntirelyDifferent  
xdebug.remote_port=9003
```

```
I: Checking remote connect back address.  
W: Remote address not found, connecting to configured address/port: nothere:9003. :-|  
E: Could not connect to client. :-(
```

Step Debugger Logs

Logging debugging communication:

```
xdebug.remote_log=/tmp/remote.log
```

Also shows connection attempts and failures:

```
xdebug.remote_connect_back=0  
xdebug.remote_port=9991
```

```
I: Connecting to configured address/port: localhost:9991.  
I: Connected to client. :-)
```

```
xdebug.remote_connect_back=1  
xdebug.remote_host=SomethingEntirelyDifferent  
xdebug.remote_port=9003
```

```
I: Checking remote connect back address.  
W: Remote address not found, connecting to configured address/port: nothere:9003. :-|  
E: Could not connect to client. :-|
```

```
xdebug.remote_connect_back=0  
xdebug.remote_host=nothere  
xdebug.remote_port=9002
```

```
I: Connecting to configured address/port: nothere:9002.  
E: Could not connect to client. :-|
```

Step Debugger Logs

Logging debugging communication:

```
xdebug.remote_log=/tmp/remote.log
```

Also shows connection attempts and failures:

```
xdebug.remote_connect_back=0  
xdebug.remote_port=9991
```

```
I: Connecting to configured address/port: localhost:9991.  
I: Connected to client. :-)
```

```
xdebug.remote_connect_back=1  
xdebug.remote_host=SomethingEntirelyDifferent  
xdebug.remote_port=9003
```

```
I: Checking remote connect back address.  
W: Remote address not found, connecting to configured address/port: nothere:9003. :-|  
E: Could not connect to client. :-|
```

```
xdebug.remote_connect_back=0  
xdebug.remote_host=nothere  
xdebug.remote_port=9002
```

```
I: Connecting to configured address/port: nothere:9002.  
E: Could not connect to client. :-|
```

Performance Woes



Dustin Fraker @dustin_fraker · Jan 11

Replying to @derickr @Xdebug @laravelphp

True, but code coverage reports are **slow** too. I have a separate command to run tests with coverage that doesn't remove **xdebug**.



2



Tom Jowitt @tommyjowitt · Jan 8

Replying to @jordanthoff @calebporzio

Interesting. I didn't know about this. We currently calculate coverage in a midnight CI run as **XDebug** is so painfully **slow**.



1



1



Cody Covey @ccovey14 · Jan 6

Replying to @GeeH

I got similar responses when I mentioned we couldn't run our symfony app with **xdebug** enabled because **xdebug** caused the app to be too **slow**.



Larachat Slack @larachatslack · 19 Dec 2018

Speed up PHPUnit Code Coverage Analysis – Nicolas Cabot – Medium PHPUnit Code Coverage Analysis is often used with **xdebug** as the coverage data source. But it has one major drawback: it can become very **slow**. As you ca...

LaraChat - Speed up PHPUnit Code Coverage Analysi...

Various Versions

	7.1.15	7.1.23-zts	7.1.32-zts	7.2.10-32bit	7.2.19-zts	7.2.7	7.3.4-zts
	7.1.15-32bit	7.1.24	7.1.3-32bit	7.2.10-zts	7.2.1-zts	7.2.7-32bit	7.3.5
	7.1.15-zts	7.1.24-32bit	7.1.3-zts	7.2.11	7.2.2	7.2.7-zts	7.3.5-32bit
	7.1.16	7.1.24-zts	7.1.4	7.2.11-32bit	7.2.20	7.2.8	7.3.5-zts
	7.1.16-32bit	7.1.25	7.1.4-32bit	7.2.11-zts	7.2.20-32bit	7.2.8-32bit	7.3.6
	7.1.16-zts	7.1.25-32bit	7.1.4-zts	7.2.12	7.2.20-zts	7.2.8-zts	7.3.6-32bit
	7.1.17	7.1.25-zts	7.1.5	7.2.12-32bit	7.2.21	7.2.9	7.3.6-zts
	7.1.17-32bit	7.1.26	7.1.5-32bit	7.2.12-zts	7.2.21-32bit	7.2.9-32bit	7.3.7
	7.1.17-zts	7.1.26-32bit	7.1.5-zts	7.2.13	7.2.21-zts	7.2.9-zts	7.3.7-32bit
7.1.0	7.1.18	7.1.26-zts	7.1.6	7.2.1-32bit	7.2.22	7.2dev	7.3.7-zts
7.1.0-32bit	7.1.18-32bit	7.1.27	7.1.6-32bit	7.2.13-32bit	7.2.22-32bit	7.2dev-32bit	7.3.8
7.1.0-zts	7.1.18-zts	7.1.27-32bit	7.1.6-zts	7.2.13-zts	7.2.22-zts	7.2dev-zts	7.3.8-32bit
7.1.1	7.1.19	7.1.27-zts	7.1.7	7.2.14	7.2.23	7.3.0	7.3.8-zts
7.1.10	7.1.19-32bit	7.1.28	7.1.7-32bit	7.2.14-32bit	7.2.2-32bit	7.3.0-32bit	7.3.9
7.1.10-32bit	7.1.19-zts	7.1.28-32bit	7.1.7-zts	7.2.14-zts	7.2.23-32bit	7.3.0-zts	7.3.9-32bit
7.1.10-zts	7.1.1-zts	7.1.28-zts	7.1.8	7.2.15	7.2.23-zts	7.3.1	7.3.9-zts
7.1.11	7.1.2	7.1.29	7.1.8-32bit	7.2.15-32bit	7.2.2-zts	7.3.10	7.3dev
7.1.11-32bit	7.1.20	7.1.29-32bit	7.1.8-zts	7.2.15-zts	7.2.3	7.3.10-32bit	7.3dev-32bit
7.1.11-zts	7.1.20-32bit	7.1.29-zts	7.1.9	7.2.16	7.2.3-32bit	7.3.10-zts	7.3dev-zts
7.1.12	7.1.20-zts	7.1.2-zts	7.1.9-32bit	7.2.16-32bit	7.2.3-zts	7.3.1-32bit	7.4dev
7.1.12-32bit	7.1.21	7.1.3	7.1.9-zts	7.2.16-zts	7.2.4	7.3.1-zts	7.4dev-32bit
7.1.12-zts	7.1.21-32bit	7.1.30	7.1dev	7.2.17	7.2.4-32bit	7.3.2	7.4dev-zts
7.1.13	7.1.21-zts	7.1.30-32bit	7.1dev-32bit	7.2.17-32bit	7.2.4-zts	7.3.2-32bit	master
7.1.1-32bit	7.1.22	7.1.30-zts	7.1dev-zts	7.2.17-zts	7.2.5	7.3.2-zts	master-32bit
7.1.13-32bit	7.1.22-32bit	7.1.31	7.2.0	7.2.18	7.2.5-32bit	7.3.3	master-zts
7.1.13-zts	7.1.22-zts	7.1.31-32bit	7.2.0-32bit	7.2.18-32bit	7.2.5-zts	7.3.3-32bit	
7.1.14	7.1.23	7.1.31-zts	7.2.0-zts	7.2.18-zts	7.2.6	7.3.3-zts	
7.1.14-32bit	7.1.2-32bit	7.1.32	7.2.1	7.2.19	7.2.6-32bit	7.3.4	
7.1.14-zts	7.1.23-32bit	7.1.32-32bit	7.2.10	7.2.19-32bit	7.2.6-zts	7.3.4-32bit	

The Xdebug Experience: Solutions

- Setting up CI (with PHP 7.4/8.0 nightlies)
- The **Big Refactor** project
- Education
Networking, settings, how Xdebug interacts
- Performance analysis and improvements for specific features

I can't do that all in my spare time

The Xdebug Experience: The Plan

Xdebug 2.8

- ✓ Improve 'breakpoint' behaviour
- ✓ PHP 7.4 support
- ✓ Set up a local CI

Xdebug 3

- The Big Refactor™
 - Introducing modes
 - Simplify configuration and validation
- Performance Improvements

Xdebug 2.8

Breakpoint Behaviour

```

1: <?php
2: function scoped()
3: {
4:     $func = function()
5:     {
6:         $a = array( // breakpoint
7:             1 => "first",
8:             2 => "second",
9:         );
10:    };
11:
12:    $func();
13: }
14:
15: scoped();
16: scoped();
17: ?>

```

```

function name:  scoped
compiled vars:  !0 = $func
line    #*  E I  O op                return  operands

   2     0  E >   EXT_NOP
   4     1      EXT_STMT
   4     2      DECLARE_LAMBDA_FUNCTION      '%00...closure.php0x7f295f0c207e'

  12     3      EXT_STMT
  12     4      INIT_DYNAMIC_CALL           !0
  12     5      EXT_FCALL_BEGIN
  12     6      DO_FCALL
  12     7      EXT_FCALL_END
  13     8      EXT_STMT
  13     9      > RETURN                    null

```

```

function name:  {closure}
line    #*  E I  O op                return  operands

   4     0  E >   EXT_NOP
   7     1      EXT_STMT

  10     2      EXT_STMT
  10     3      > RETURN                    null

```

Breakpoint Behaviour

```

1: <?php
2: function scoped()
3: {
4:     $func = function()
5:     {
6:         $a = array( // breakpoint
7:             1 => "first",
8:             2 => "second",
9:         );
10:    };
11:
12:    $func();
13: }
14:
15: scoped();
16: scoped();
17: ?>

```

Turned off OPcache's optimiser

→

```

function name:  scoped
compiled vars:  !0 = $func
line   #*  E  I  O  op                return  operands

   2    0   E  >   EXT_NOP
   4    1           EXT_STMT
   6    2           DECLARE_LAMBDA_FUNCTION      '%00...closure.php0x7efeba02b07e'
  10    3           ASSIGN                      !0, ~1
  12    4           EXT_STMT
        5           INIT_DYNAMIC_CALL            !0
        6           EXT_FCALL_BEGIN
        7           DO_FCALL
        8           EXT_FCALL_END
  13    9           EXT_STMT
        10          > RETURN                      null

```

```

function name:  {closure}
compiled vars:  !0 = $a
line   #*  E  I  O  op                return  operands

   4    0   E  >   EXT_NOP
   7    1           EXT_STMT
        2           ASSIGN                      !0, <array>
  10    3           EXT_STMT
        4           > RETURN                      null

```

Breakpoint Behaviour

```

1: <?php
2: function scoped()
3: {
4:     $func = function()
5:     {
6:         $a = array( // breakpoint
7:             1 => "first",
8:             2 => "second",
9:         );
10:    };
11:
12:    $func();
13: }
14:
15: scoped();
16: scoped();
17: ?>

```

Turned off OPcache's optimiser

→

```

function name:  scoped
compiled vars:  !0 = $func
line    #*  E I  O op                return  operands

   2     0  E >   EXT_NOP
   4     1      EXT_STMT
   6     2      DECLARE_LAMBDA_FUNCTION      '%00...closure.php0x7efeba02b07e'
  10    3      ASSIGN                        !0, ~1
  12     4      EXT_STMT
        5      INIT_DYNAMIC_CALL            !0
        6      EXT_FCALL_BEGIN
        7      DO_FCALL
        8      EXT_FCALL_END
  13     9      EXT_STMT
        10     > RETURN                        null

```

```

function name:  {closure}
compiled vars:  !0 = $a
line    #*  E I  O op                return  operands

   4     0  E >   EXT_NOP
   7     1      EXT_STMT
        2      ASSIGN                        !0, <array>
  10     3      EXT_STMT
        4      > RETURN                        null

```

demo

Breakpoint Behaviour

```

1: <?php
2: function scoped()
3: {
4:     $func = function()
5:     {
6:         $a = array( // breakpoint
7:             1 => "first",
8:             2 => "second",
9:         );
10:    };
11:
12:    $func();
13: }
14:
15: scoped();
16: scoped();
17: ?>

```

Turned off OPcache's optimiser

→

```

function name:  scoped
compiled vars:  !0 = $func
line    #*  E I  O op                return  operands

   2     0  E >   EXT_NOP
   4     1      EXT_STMT
   6     2      DECLARE_LAMBDA_FUNCTION    '%00...closure.php0x7efeba02b07e'
  10    3      ASSIGN                    !0, ~1
  12     4      EXT_STMT
        5      INIT_DYNAMIC_CALL          !0
        6      EXT_FCALL_BEGIN
        7      DO_FCALL
        8      EXT_FCALL_END
  13     9      EXT_STMT
        10     > RETURN                    null

```

```

function name:  {closure}
compiled vars:  !0 = $a
line    #*  E I  O op                return  operands

   4     0  E >   EXT_NOP
   7     1      EXT_STMT
        2      ASSIGN                    !0, <array>
  10     3      EXT_STMT
        4      > RETURN                    null

```

demo

Local Continuous Integration

- Any cloud based CI is too slow due to lack of parallelisation
Travis CI, Circle CI, AppVeyor
- Many configuration pivots:
NTS/ZTS, 64-bit/32-bit, OPcache on/off
- Changes between PHP patch versions
Line changes for OPcodes, bug fixes in PHP
- I have a local machine with plenty of cores and memory
- Nightly builds of PHP 7.4 and 8.0

<https://xdebug.org/ci>

Xdebug 3

Code Reorganisation

```
derick@singlemalt:~/dev/php/xdebug-xdebug$ ls *.*[hc]
```

config.h	xdebug_filter.c	xdebug_monitor.c	xdebug_superglobals.h
php_xdebug.h	xdebug_filter.h	xdebug_monitor.h	xdebug_trace_computerized.c
usefulstuff.c	xdebug_gc_stats.c	xdebug_private.c	xdebug_trace_computerized.h
usefulstuff.h	xdebug_gc_stats.h	xdebug_private.h	xdebug_trace_html.c
xdebug_branch_info.c	xdebug_handler_dbgp.c	xdebug_profiler.c	xdebug_trace_html.h
xdebug_branch_info.h	xdebug_handler_dbgp.h	xdebug_profiler.h	xdebug_trace_textual.c
xdebug.c	xdebug_handlers.c	xdebug_set.c	xdebug_trace_textual.h
xdebug_code_coverage.c	xdebug_handlers.h	xdebug_set.h	xdebug_tracing.c
xdebug_code_coverage.h	xdebug_hash.c	xdebug_stack.c	xdebug_tracing.h
xdebug_com.c	xdebug_hash.h	xdebug_stack.h	xdebug_var.c
xdebug_com.h	xdebug_llist.c	xdebug_str.c	xdebug_var.h
xdebug_compat.c	xdebug_llist.h	xdebug_str.h	xdebug_xml.c
xdebug_compat.h	xdebug_mm.h	xdebug_superglobals.c	xdebug_xml.h

Code Reorganisation

```
derick@gargleblaster:~/dev/php/derickr-xdebug/src$ ls -v ../*. [ch] */*. [ch]
```

../config.h	debugger/com.h	lib/str.c
../php_xdebug.h	debugger/handlers.c	lib/str.h
../xdebug.c	debugger/handlers.h	lib/usefulstuff.c
base/base.c	debugger/handler_dbgp.c	lib/usefulstuff.h
base/base.h	debugger/handler_dbgp.h	lib/var.c
base/filter.c	gcstats/gc_stats.c	lib/var.h
base/filter.h	gcstats/gc_stats.h	lib/xml.c
base/monitor.c	gcstats/gc_stats_private.h	lib/xml.h
base/monitor.h	lib/compat.c	profiler/profiler.c
base/stack.c	lib/compat.h	profiler/profiler.h
base/stack.h	lib/hash.c	tracing/trace_computerized.c
base/superglobals.c	lib/hash.h	tracing/trace_computerized.h
base/superglobals.h	lib/llist.c	tracing/trace_html.c
coverage/branch_info.c	lib/llist.h	tracing/trace_html.h
coverage/branch_info.h	lib/mm.h	tracing/trace_textual.c
coverage/code_coverage.c	lib/private.c	tracing/trace_textual.h
coverage/code_coverage.h	lib/private.h	tracing/tracing.c
coverage/code_coverage_private.h	lib/set.c	tracing/tracing.h
debugger/com.c	lib/set.h	tracing/tracing_private.h

Code Reorganisation

```
derick@gargleblaster:~/dev/php/derickr-xdebug/src$ ls -v ../*. [ch] */*. [ch]
```

../config.h	debugger/com.h	lib/str.c
../php_xdebug.h	debugger/handlers.c	lib/str.h
../xdebug.c	debugger/handlers.h	lib/usefulstuff.c
base/base.c	debugger/handler_dbgp.c	lib/usefulstuff.h
base/base.h	debugger/handler_dbgp.h	lib/var.c
base/filter.c	gcstats/gc_stats.c	lib/var.h
base/filter.h	gcstats/gc_stats.h	lib/xml.c
base/monitor.c	gcstats/gc_stats_private.h	lib/xml.h
base/monitor.h	lib/compat.c	profiler/profiler.c
base/stack.c	lib/compat.h	profiler/profiler.h
base/stack.h	lib/hash.c	tracing/trace_computerized.c
base/superglobals.c	lib/hash.h	tracing/trace_computerized.h
base/superglobals.h	lib/llist.c	tracing/trace_html.c
coverage/branch_info.c	lib/llist.h	tracing/trace_html.h
coverage/branch_info.h	lib/mm.h	tracing/trace_textual.c
coverage/code_coverage.c	lib/private.c	tracing/trace_textual.h
coverage/code_coverage.h	lib/private.h	tracing/tracing.c
coverage/code_coverage_private.h	lib/set.c	tracing/tracing.h
debugger/com.c	lib/set.h	tracing/tracing_private.h

Introducing Modes

Off

Loaded, but miniscule overhead

Introducing Modes

Base

Every day debugging aide

Introducing Modes

Debugger

Debugging applications with an IDE
Previously: remote debugger

Introducing Modes

Tracing

Tracing the execution of a script to a file

Introducing Modes

Profiler

Profiling application performance

Introducing Modes

Coverage

Which code is run during tests

Introducing Modes

Time Traveller



Introducing Modes

Configuration Settings

```
xdebug.auto_trace
xdebug.cli_color
xdebug.collect_assignments
xdebug.collect_includes
xdebug.collect_params
xdebug.collect_return
xdebug.collect_vars
xdebug.coverage_enable
xdebug.default_enable
xdebug.dump.COOKIE
xdebug.dump.ENV
xdebug.dump.FILES
xdebug.dump.GET
xdebug.dump.POST
xdebug.dump.REQUEST
xdebug.dump.SERVER
xdebug.dump.SESSION
xdebug.dump_globals
xdebug.dump_once
xdebug.dump_undefined
xdebug.extended_info
xdebug.file_link_format
xdebug.filename_format
xdebug.force_display_errors
xdebug.force_error_reporting
xdebug.gc_stats_enable
xdebug.gc_stats_output_dir
xdebug.gc_stats_output_name
xdebug.halt_level
xdebug.idekey
xdebug.max_nesting_level
xdebug.max_stack_frames
xdebug.overload_var_dump
```

```
xdebug.profiler_aggregate
xdebug.profiler_append
xdebug.pprofiler_enable
xdebug.profiler_enable_trigger
xdebug.profiler_enable_trigger_value
xdebug.profiler_output_dir
xdebug.profiler_output_name
xdebug.remote_addr_header
xdebug.remote_autostart
xdebug.remote_connect_back
xdebug.remote_cookie_expire_time
xdebug.remote_enable
xdebug.remote_handler
xdebug.remote_host
xdebug.remote_log
xdebug.remote_mode
xdebug.remote_port
xdebug.remote_timeout
xdebug.scream
xdebug.show_error_trace
xdebug.show_exception_trace
xdebug.show_local_vars
xdebug.show_mem_delta
xdebug.trace_enable_trigger
xdebug.trace_enable_trigger_value
xdebug.trace_format
xdebug.trace_options
xdebug.trace_output_dir
xdebug.trace_output_name
xdebug.var_display_max_children
xdebug.var_display_max_data
xdebug.var_display_max_depth
```

Configuration Settings

xdebug. auto_trace	xdebug.always_start
xdebug. cli_color	
xdebug.base.collect_assignments	
xdebug.base.collect_includes	
xdebug.base.collect_params	
xdebug.base.collect_return	
xdebug.base.collect_vars	
xdebug. coverage_enable	xdebug.mode
xdebug. default_enable	xdebug.mode
xdebug.display.dump.COOKIE	
xdebug.display.dump.ENV	
xdebug.display.dump.FILES	
xdebug.display.dump.GET	
xdebug.display.dump.POST	
xdebug.display.dump.REQUEST	
xdebug.display.dump.SERVER	
xdebug.display.dump.SESSION	
xdebug.display.dump_globals	
xdebug. dump_once	
xdebug.display.dump_undefined	
xdebug.extended_info	removed in Xdebug 2.8
xdebug.base.file_link_format	
xdebug.base.filename_format	
xdebug.base.force_display_errors	
xdebug.base.force_error_reporting	
xdebug.base.gc_stats_enable	
xdebug. gc_stats_output_dir	xdebug.output_dir
xdebug. gc_stats_output_name	xdebug.output_name
xdebug.base.halt_level	
xdebug. idekey	
xdebug.base.max_nesting_level	
xdebug.display.max_stack_frames	
xdebug.display.overload_var_dump	

xdebug. profiler_aggregate	
xdebug. profiler_append	
xdebug. profiler_enable	xdebug.mode
xdebug. profiler_enable_trigger	xdebug.enable_trigger
xdebug. profiler_enable_trigger_value	xdebug.enable_trigger_value
xdebug. profiler_output_dir	xdebug.output_dir
xdebug. profiler_output_name	xdebug.output_name
xdebug.debugger.addr_header	
xdebug. remote.autostart	xdebug.always_start
xdebug.debugger.connect_back	
xdebug.debugger.cookie_expire_time	
xdebug. remote_enable	xdebug.mode
xdebug. remote_handler	
xdebug.debugger.host	
xdebug.debugger.log	
xdebug.debugger.mode	
xdebug.debugger.port	
xdebug.debugger.timeout	
xdebug. scream	always on ☺
xdebug.display.show_error_trace	
xdebug.display.show_exception_trace	
xdebug.display.show_local_vars	
xdebug. show_mem_delta	always on
xdebug. trace_enable_trigger	xdebug.enable_trigger
xdebug. trace_enable_trigger_value	xdebug.enable_trigger_value
xdebug.tracing.format	
xdebug.tracing.options	
xdebug. trace_output_dir	xdebug.output_dir
xdebug. trace_output_name	xdebug.output_name
xdebug.display.max_children	
xdebug.display.max_data	
xdebug.display.max_depth	

Performance Woes



Dustin Fraker @dustin_fraker · Jan 11

Replying to @derickr @Xdebug @laravelphp

True, but code coverage reports are **slow** too. I have a separate command to run tests with coverage that doesn't remove **xdebug**.



2



Tom Jowitt @tommyjowitt · Jan 8

Replying to @jordanthoff @calebporzio

Interesting. I didn't know about this. We currently calculate coverage in a midnight CI run as **XDebug** is so painfully **slow**.



1



1



Cody Covey @ccovey14 · Jan 6

Replying to @GeeH

I got similar responses when I mentioned we couldn't run our symfony app with **xdebug** enabled because **xdebug** caused the app to be too **slow**.



Larachat Slack @larachatslack · 19 Dec 2018

Speed up PHPUnit Code Coverage Analysis – Nicolas Cabot – Medium PHPUnit Code Coverage Analysis is often used with **xdebug** as the coverage data source. But it has one major drawback: it can become very **slow**. As you ca...

LaraChat - Speed up PHPUnit Code Coverage Analysi...

Setup Woes

**Sebastian Ruchlewicz** @SRUchlewicz · 12 Aug 2018
Anyone had **problems** with configuring [#xdebug](#) with [#Docker](#) containers and [#phpstorm](#) ? If yes, check out my last post ruchlewicz.net/blog/xdebug-do... [#php](#)



Xdebug + Docker + PhpStorm | Web Develop...
Have you tried to run xdebug on docker container and catching requests with ruchlewicz.net



**The Bahree** @arjunbahree · 14 Apr 2018
Docker Swarm + **XDebug**: **problem** and solution: ift.tt/2qxDBxp



**Victor Bolshov** @crocodile2u · 14 Apr 2018
I just published "**Docker** Swarm + **XDebug**: **problem** and solution" medium.com/p/docker-swarm...



**Christian Engel** @Paratron · 22 Mar 2018
Its funny how [#PHP](#) [#xdebug](#) always comes back once a while and makes trouble. Connection **problems** back in the days with [#XAMPP](#), later with [#vagrant](#), now with [#docker](#) [#dejavue](#) [#GroundhogDay](#)



**Vojin Urosevic** 沃因 @vojinoosevic · 20 Feb 2018
Problem Docker-Xdebug-Osx/Mac dlvr.it/QH5tRf



Supporting Xdebug

It's Open Source and free (as in "free beer")

Using Xdebug saves you a lot of time

I would like to work more on Xdebug, to help **you**

Features and Bugs

<https://bugs.xdebug.org>

● **BECOME A PATRON**

<https://www.patreon.com/derickr>

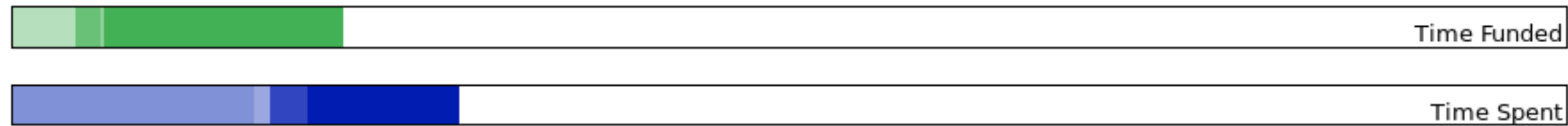
Business Support

COMMUNITY	PRO 1 developer	BUSINESS 2-20 developers
PATREON	£ 199/YEAR	£ 999/YEAR
✗ Support: Stackoverflow ✗ Issue Priority: Low ✓ Monthly News Post	✓ Support: E-mail ✓ Issue Priority: Medium ✓ Access to Commercial Add-ons (soon) ✓ Monthly E-mail Newsletter (opt-in) ✓ Invoice	✓ Support: E-mail ✓ Issue Priority: High ✓ Access to Commercial Add-ons (soon) ✓ Monthly E-mail Newsletter (opt-in) ✓ Invoice ✓ "Thank You" with link on site
	E-mail billing+pro@xdebug.org to sign up	E-mail billing+business@xdebug.org to sign up

<https://xdebug.org/support>

Transparency

SEPTEMBER 2019



Day	Type	Description	Hours
3	generic	Writing Xdebug August 2019 update	1.50
3	bug	Investigate Pull Request #474 for issue #1534 and #1665	1.00
5	bug	Fixed issue #1665: Segfault with garbage collection and complex function arguments	3.00
6	bug	Investigating #1699: Crash during debugging Phalcon project	1.00
16	bug	Failing again to reproduce #1699: Crash during debugging Phalcon project	0.75
16	bug	Triaging OSX bug #1701: Slowness due to calls to gettimeofday	0.50
17	bug	Reproduced #1699, and provided a fix	1.25
19	bug	Triaging bug #1703 (Symbol not found: _compiler_globals)	0.25
19	bug	Triaging bug #1704 (Xdebug requires Zend Engine API version 320190529), starting and publishing new Windows builds of 2.8.0beta2	0.50
20	xdebug3	Refactoring Xdebug globlas into constituent parts	2.50
21	xdebug3	Refactoring Xdebug globlas into constituent parts	2.00
22	xdebug3	Refactoring Xdebug globlas into constituent parts	0.50
23	website	Merging Matt's website re-architecture PR, added caching for download and log	1.00
23	xdebug3	Windows part of moving files, and fixing the "core" issue/directory.	3.00
26	xdebug3	Moving and renaming the files again	1.50
27	xdebug3	Moving and renaming the files	2.50
28	xdebug3	Moving and renaming the files	2.00
29	bug	Triaging and fixing bug #1705: Crash while debugging with ionCube being used	1.50
29	xdebug3	Refactoring initialisation (part 1 of n)	1.00
30	generic	Writing Xdebug September 2019 update	1.00
30	xdebug3	Streaming the fix for the crash that I had introduced during refactoring (part 1)	0.50

CURRENT SUPPORTERS

[TYPO3 GmbH](#)

[Tideways](#)

[Intracto](#)

You can also be listed as a supporter by [signing up](#) for a *Business* package.



[BECOME A PATRON](#)

<https://phpinternals.news>



Any Queries?

Resources



Slides

<https://joind.in/26837>

Derick Rethans — @derickr — derick@xdebug.org