

Debugging: Past, Present and Future

Derick Rethans

derick@mongodb.com – [@derickr](https://twitter.com/derickr)

<http://joind.in/11790>

About Me

Derick Rethans

- Dutchman living in London
- PHP Driver Maintainer for MongoDB
- Anderskor's Camouflage Encoder
- Author of Xdebug
- Author of the mdecrypt, input_filter, dbus, translit and date/time extensions



Today's agenda

- Past
- Present
- Future

Pretty Printing Variables

```
array
  0 => null
  1 => float 3.141592654
  2 =>
    array
      'dutch' =>
        object(locale)[1]
          public 'lang' => string 'nl' (length=2)
          public 'variation' => string 'abcdabcdabcdabcdabcdabcd...' (length=2000)
          public 'charsets' =>
            array
              0 => string 'iso-88591' (length=9)
              1 => string 'iso-8859-15' (length=11)
          private 'mb_supported' => boolean false
          protected 'self' =>
            &object(locale)[1]
          protected 'id' => int 1
```

Options:

```
<?php
ini_set( 'xdebug.var_display_max_data', 25 );
ini_set( 'xdebug.var_display_max_depth', 4 );
ini_set( 'xdebug.var_display_max_children', 32 );
?>
```

Turn off in php.ini or .htaccess with:

```
xdebug.overload_var_dump = 0
```

Ladybug

<https://github.com/raulfraile/ladybug.git>

```
array(3)
  [0] null
  [1] float 3.141592654
  [2] array(1)
      [dutch] object(Xlocale)

/home/httpd/html/test/xdebug/var_dump-
ladybug.php:30
```


<https://github.com/oodle/krumo.git>

► ... *Array(3)* ...

Called from `/home/httpd/html/test/xdebug/var_dump-krumo.php`, line 30

Krumo version 0.5.0 | <https://github.com/oodle/krumo>

<https://github.com/raveren/kint.git>

```

- $a array (3)

```

NULL

```
float 3.141592654
```

- array (1)

```
[-] 'dutch' => object Xlocale (8)
```

```
object (8) Available methods (2)
```

```
public lang -> string (2) "n1"
```

```
public variation -> string (2000) "abcdabcdabcdabcdabcdabcdabcdabcdabcdabcdabcd ..."
```

[illegible]

```
[-] public charsets -> array (2)
```

```
string (9) "iso-88591"
```

```
string (11) "iso-8859-15"
```

```
private mb_supported -> bool FALSE
```

```
protected self -> object Xlocale (8) *RECURSION*
```

```
protected id -> integer 1
```

```
[-]protected color -> string (7) "#004499"
```

CSS color

#004499

hex : #004499

```
rgb : rgb( 0, 68, 153 )
```

```
hsl : hsl( 213, 100%, 30% )
```

```
[-] protected timestamp -> integer 1412243210
```

timectann

Symfony var-dumper

<https://github.com/symfony/var-dumper.git>

```
array:3 [▼  
  0 => null  
  1 => 3.141592654  
  2 => array:1 [▼  
    "dutch" => Xlocale {▶}  
  ]  
]
```


Colours on the Command Line

new in 2.2

Scream

- PHP's @ operator hides warnings and errors
- `xdebug.scream=1` makes PHP ignore @



Tracing



Function traces

- Different types: **Textual**, **Parsable** and **HTML**
- Tracing only parts of an application with **xdebug_start_trace()** and **xdebug_stop_trace()**.
- `xdebug.collect_return`:

```
-> explode('/', '/xdebug-phpnw13/11') /home/httpd/pres2/show2.php:121  
=> array (0 => '', 1 => 'xdebug-phpnw13', 2 => '11')
```

- `xdebug.collect_assignments`:

```
-> ezcTemplateVariableCollection->__set($name = 'slideNr', $value = '11') /home/httpd/p  
=> $this->variables['slideNr'] = '11' /home/derick/dev/zetacomponents/trunk/Template
```

Live Debugging

- DBGp, common Debugging protocol
- Cross-language: PHP, **HHVM**, Python, Perl...
- Komodo (Linux, Windows, Mac – commercial)
- Netbeans (Java-based – free)
- PHPStorm (Java-based – commercial)
- (and many others: <http://xdebug.org/docs/remote#clients>)

Netbeans

www.derickrethans.nl - NetBeans IDE 6.9.1

File Edit View Navigate Source Refactor Run Debug Team Tools Window Help

<default> Search (Ctrl+I)

Proj... Files Services

Start Page x test.php x index.php x

index.php - Navigator

```

1  <?php
2  require 'php/config.php';
3  $url = trim( $_SERVER['REQUEST_URI'], '/' );
4  $dopplr = true;
5
6  if ( $_SERVER['SERVER_NAME'] == 'www.derickrethans.nl' )
7  {
8      header( "Location: http://derickrethans.nl/$url" );
9  }
10
11 //check common thingies
12 if ( $url == '' )
13 {
14     $title = "Home";

```

Variables Call Stack Breakpoints Tasks

Name	Type	Value
Superglobals		
api	drBlogApi	
d	ezcDbHandlerSqlite	
transactionNestingLevel	integer	0
transactionErrorFlag	boolean	0
identifierQuoteChars	array	

Filters: [Icons]

netbeans-xdebug running 4 | 1 INS

Komodo

test.php (~dev/php/xdebug-demos/cli-debug) - ActiveState Komodo IDE 6.1 - Debugger

File Edit Code Navigation View Debug Project Tools Help

Start Page test.php x

```

4      private $priv;
5      protected $prot;
6      public $publ;
7
8      function __construct($a)
9      {
10         $this->priv = $a * 2;
11         $this->prot = $a * 2.5;
12         $this->publ = $a * M_PI;
13     }
14
15
16     function foo($a)
17     {

```

Breakpoints Command Output Test Results SCC Output **Debug: test.php**

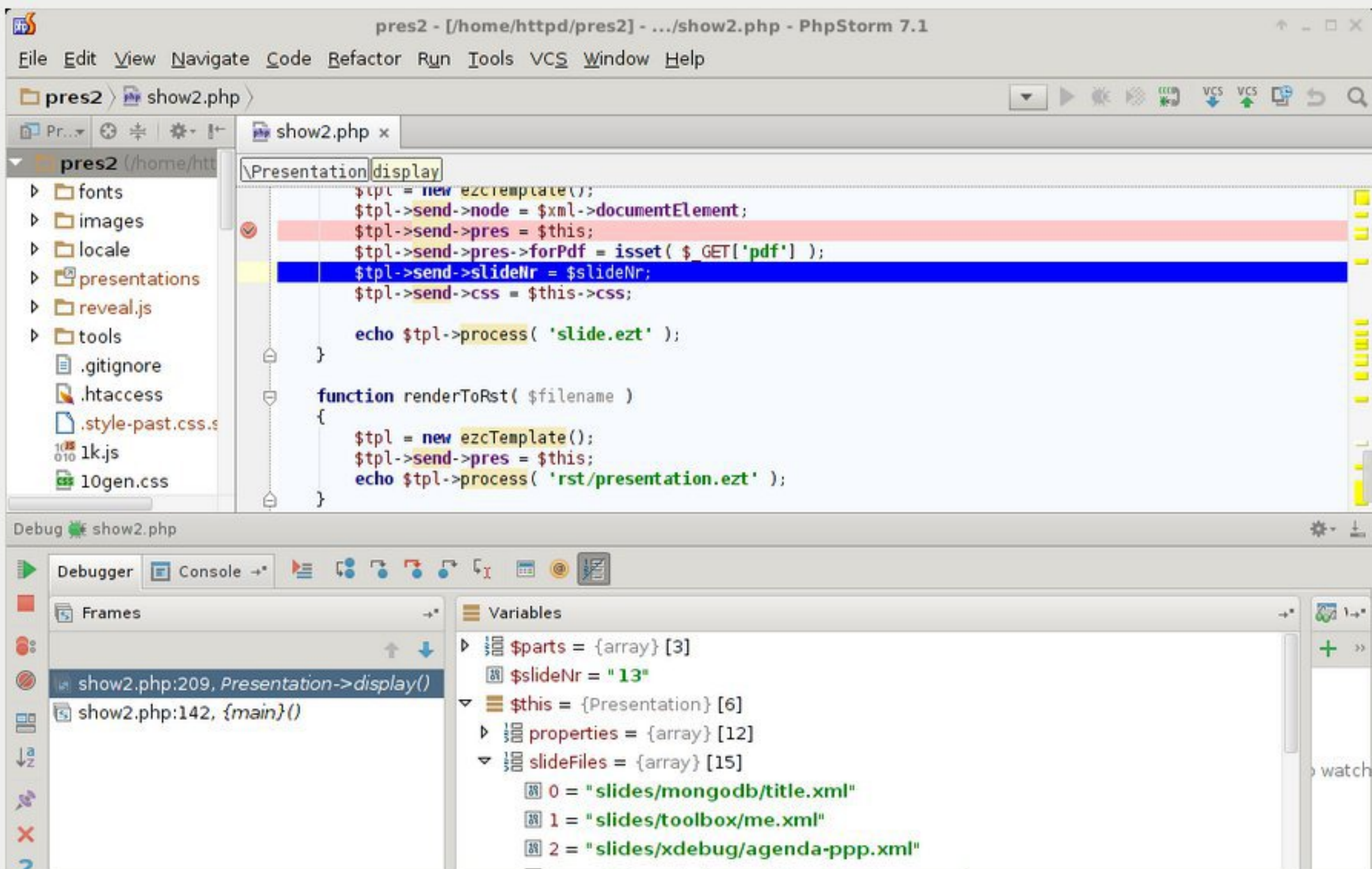
Break at /home/derick/dev/php/xdebug-demos/cli-debug/test.php, line 11.

Name	Type	Value
a	float	2.718281828459
▼ this	TheClass	
priv	float	5.4365636569181
prot	null	
publ	null	

Watch Locals Superglobals

Ready **__construct** UTF-8 Ln: 11 Col: 1 PHP

PHP Storm



Activating the Remote Debugger

On the shell:

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


Activating the Remote Debugger

With browser extensions:

The **easiest** Xdebug (Firefox):

<https://addons.mozilla.org/en-US/firefox/addon/the-easiest-xdebug/>

Xdebug Helper (Chrome): <https://chrome.google.com/extensions/detail/eadndfjplgieldjbigjakmdgkmoaaaoc>

Xdebug Toggler for Safari: <https://github.com/benmatselby/xdebug-toggler>

Xdebug launcher for Opera: <https://addons.opera.com/addons/extensions/details/xdebug-launcher/?display=en>

Remote debugging log

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

- Logs all debugger communication
- Also shows connection attempts and failures:

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

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

```
xdebug.remote_connect_back=1  
xdebug.remote_host=nothere2  
xdebug.remote_port=9003
```

```
I: Checking remote connect back address.  
W: Remote address not found, connecting to configured address/port: nowhere2:9003. :-(  
E: Could not connect to client. :-(
```

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

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

PHPDBG

- Extra SAPI module that comes with PHP 5.6
- it takes itself very serious:
40:0 vote confirms the awesomeness.
- It does some things that Xdebug does, but not quite as good
- It does some things that VLD does, but not quite as good

PHPDBG

```
[GIT: dericks-renderer][PHP: 5.6.2-dev ]
derick@whisky:/home/httpd/pres2 $ phpdbg show2.php
[Welcome to phpdbg, the interactive PHP debugger, v0.4.0]
To get help using phpdbg type "help" and press enter
[Please report bugs to <http://github.com/krakjoe/phpdbg/issues>]
[Attempting compilation of /home/httpd/pres2/show2.php]
[Success]
phpdbg> █
```


vld

Dumps oparray per element:

- main script
- function
- class method
- dead code analysis

Usage:

```
pecl install vld  
php -dextension=vld.so -dvld.active=1 script.php
```

vld paths #1: code

```
<?php
class ExceptionFoo extends Exception {}
class ExceptionBar extends Exception {}
class ExceptionBaz extends Exception {}

function trycatch($action)
{
    echo "Let's do some stuff!\n";
    $f = 5 * pow(3, 7);

    try {
        switch ($action) {
            case 1: throw new ExceptionFoo("test");
            case 2: throw new ExceptionBaz("test");
            case 3: throw new Exception("test");
        }
        echo "Not thrown\n";
    } catch (ExceptionFoo $e) {
        echo "caught\n";
    } catch (ExceptionBar $e) {
        echo "caught\n";
    } catch (ExceptionBaz $e) {
        echo "caught\n";
    }

    echo "And do some more\n";
}

trycatch(1);
```

	46		DO_FCALL_BY_NAME	1	
	47		EXT_FCALL_END		
	48	>	THROW	0	\$11
16	49*		JMP		->50
17	50	>	EXT_STMT		
	51		ECHO		'Not+thrown%0A'
18	52	>	JMP		->64
	53	E >	CATCH	57	'ExceptionFoo',
19	54	>	EXT_STMT		
	55		ECHO		'caught%0A'
20	56	>	JMP		->64
	57	E >	CATCH	61	'ExceptionBar',
21	58	>	EXT_STMT		
	59		ECHO		'caught%0A'
22	60	>	JMP		->64
	61	E >	CATCH	64	'ExceptionBaz',
23	62	>	EXT_STMT		
	63		ECHO		'caught%0A'
26	64	>	EXT_STMT		
	65		ECHO		'And+do+some+mo
27	66		EXT_STMT		
	67	>	RETURN		null

```

branch: # 0; line: 6-13; sop: 0; eop: 16; out1: 17; out2: 26
branch: # 17; line: 13-13; sop: 17; eop: 24; out1: -2
branch: # 26; line: 14-14; sop: 26; eop: 28; out1: 29; out2: 38
branch: # 29; line: 14-14; sop: 29; eop: 36; out1: -2
branch: # 38; line: 15-15; sop: 38; eop: 40; out1: 41; out2: 50
branch: # 41; line: 15-15; sop: 41; eop: 48; out1: -2
branch: # 50; line: 17-18; sop: 50; eop: 52; out1: 64
branch: # 53; line: 18-18; sop: 53; eop: 53; out1: 54; out2: 57
branch: # 54; line: 19-20; sop: 54; eop: 56; out1: 64
branch: # 57; line: 20-20; sop: 57; eop: 57; out1: 58; out2: 61
branch: # 58; line: 21-22; sop: 58; eop: 60; out1: 64

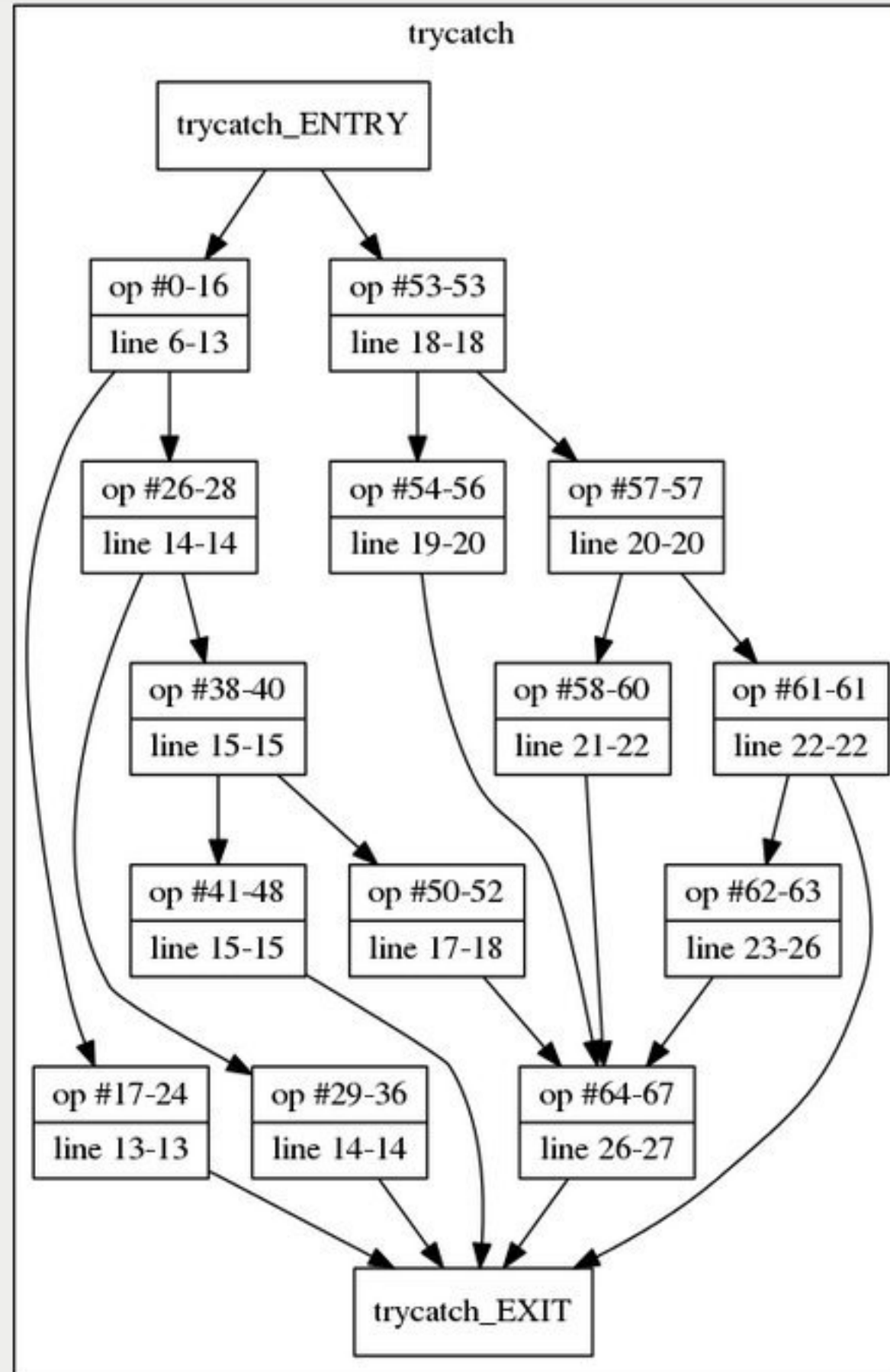
```



```

1 <?php
2 class ExceptionFoo extends Exception
3 class ExceptionBar extends Exception
4 class ExceptionBaz extends Exception
5
6 function trycatch($action)
7 {
8     echo "Let's do some stuff!\n";
9     $f = 5 * pow(3, 7);
10
11     try {
12         switch ($action) {
13             case 1: throw new ExceptionFoo;
14             case 2: throw new ExceptionBar;
15             case 3: throw new ExceptionBaz;
16         }
17         echo "Not thrown\n";
18     } catch (ExceptionFoo $e) {
19         echo "caught\n";
20     } catch (ExceptionBar $e) {
21         echo "caught\n";
22     } catch (ExceptionBaz $e) {
23         echo "caught\n";
24     }
25
26     echo "And do some more\n";
27 }
28
29 trycatch(1);
30 trycatch(2);
31 try {
32     trycatch(3);
33 } catch (Exception $e) {
34     echo "CAUGHT IN MAIN\n";
35 }
36 ?>

```



vld paths #2: code

```
<?php
function loopy($i)
{
    do
    {
        echo $i, ' ';

    } while( $i-- );
    echo "\n";
}

function ifthenelse( $a, $b )
{
    if ($a) {
        echo "A HIT\n";
    }
    if ($b) {
        echo "B HIT\n";
    }
}

loopy(0);
ifthenelse( true, false );
ifthenelse( false, true );
?>
```

vld paths #2: output

Function ifthenelse:

filename: /home/derick/dev/php/derickr-xdebug/tests/bug01034-004.inc

function name: ifthenelse

number of ops: 15

compiled vars: !0 = \$a, !1 = \$b

line	#*	E	I	O	op	fetch	ext	return	operands
12	0	E	>		EXT_NOP				
	1				RECV			!0	
	2				RECV			!1	
14	3				EXT_STMT				
	4		>		JMPZ				!0, ->8
15	5		>		EXT_STMT				
	6				ECHO				'A+HIT%0A'
16	7		>		JMP				->8
17	8		>		EXT_STMT				
	9		>		JMPZ				!1, ->13
18	10		>		EXT_STMT				
	11				ECHO				'B+HIT%0A'
19	12		>		JMP				->13
20	13		>		EXT_STMT				
	14		>		RETURN				null

branch: # 0; line: 12- 14; sop: 0; eop: 4; out1: 5; out2: 8

branch: # 5; line: 15- 16; sop: 5; eop: 7; out1: 8

branch: # 8; line: 17- 17; sop: 8; eop: 9; out1: 10; out2: 13

branch: # 10; line: 18- 19; sop: 10; eop: 12; out1: 13

branch: # 13; line: 20- 20; sop: 13; eop: 14; out1: -2

path #1: 0, 5, 8, 10, 13,

path #2: 0, 5, 8, 13,

path #3: 0, 8, 10, 13,

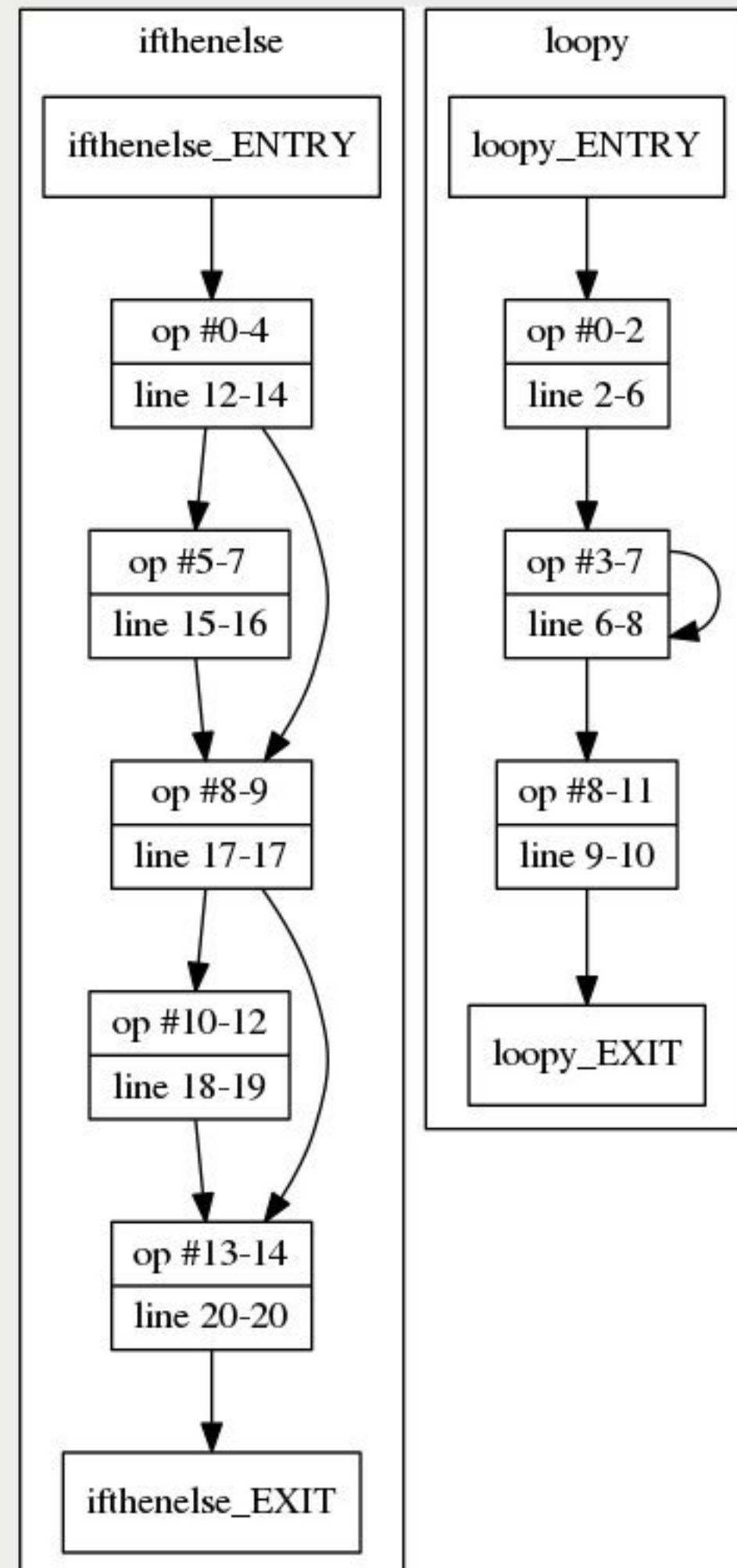
path #4: 0, 8, 13,

End of function ifthenelse

```

1 <?php
2 function loopy($i)
3 {
4     do
5     {
6         echo $i, ' ';
7     } while( $i-- );
8     echo "\n";
9 }
10
11
12 function ifthenelse( $a, $b )
13 {
14     if ($a) {
15         echo "A HIT\n";
16     }
17     if ($b) {
18         echo "B HIT\n";
19     }
20 }
21
22 loopy(0);
23 ifthenelse( true, false );
24 ifthenelse( false, true );
25 ?>

```



Xdebug code coverage

```
--TEST--  
Test for bug #1034: path coverage  
--FILE--  
<?php  
include 'dump-branch-coverage.inc';  
  
xdebug_start_code_coverage(XDEBUG_CC_UNUSED | XDEBUG_CC_DEAD_CODE | XDEBUG_CC_BRANCH_CHECK)  
  
include 'bug01034-004.inc';  
  
xdebug_stop_code_coverage(false);  
$c = xdebug_get_code_coverage();  
dump_branch_coverage($c);  
?>  
--EXPECTF--  
...
```

Xdebug code output

ifthenelse

- branches

- 00; OP: 00-04; line: 12-14 HIT; out1: 05 HIT; out2: 08 HIT
- 05; OP: 05-07; line: 15-16 HIT; out1: 08 HIT
- 08; OP: 08-09; line: 17-17 HIT; out1: 10 HIT; out2: 13 HIT
- 10; OP: 10-12; line: 18-19 HIT; out1: 13 HIT
- 13; OP: 13-14; line: 20-20 HIT; out1: EX X

- paths

- 0 5 8 10 13: X
- 0 5 8 13: HIT
- 0 8 10 13: HIT
- 0 8 13: X

loopy

- branches

- 00; OP: 00-02; line: 02-06 HIT; out1: 03 HIT
- 03; OP: 03-07; line: 06-08 HIT; out1: 08 HIT; out2: 03 X
- 08; OP: 08-11; line: 09-10 HIT; out1: EX X

- paths

- 0 3 8: HIT
- 0 3 3 8: X

```

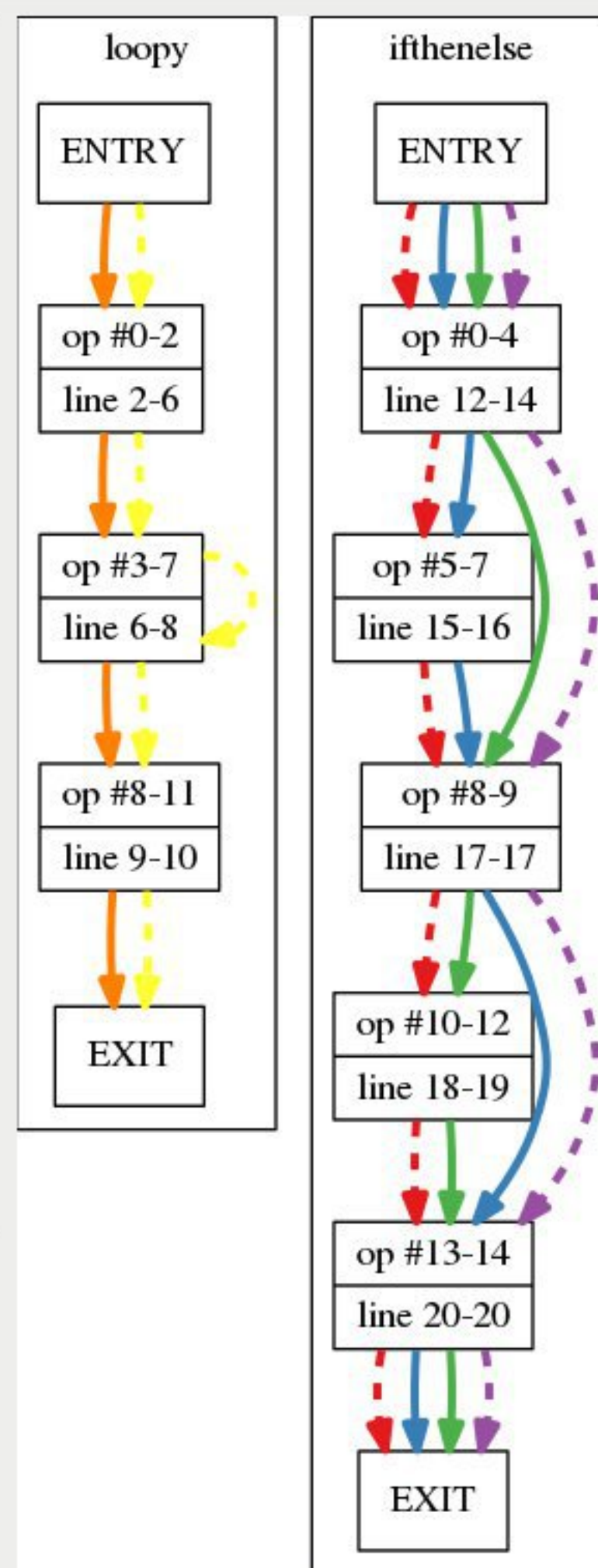
<?php
function loopy($i)
{
    do
    {
        echo $i, ' ';

    } while( $i-- );
    echo "\n";
}

function ifthenelse( $a, $b )
{
    if ($a) {
        echo "A HIT\n";
    }
    if ($b) {
        echo "B HIT\n";
    }
}

loopy(0);
ifthenelse( true, false );
ifthenelse( false, true );
?>

```





Resources



Slides:

<http://derickrethans.nl/talks/xdebug-phpnw14>

Derick Rethans — @derickr — derick@mongodb.com

Feedback: <http://joind.in/11790>