

# Efficient debugging

OSCOM 4

September 30th, 2004. Zürich, Switzerland

Derick Rethans <[derick@php.net](mailto:derick@php.net)>

<http://pres.derickrethans.nl/xdebug-oscom4>



Share your information



Who...

- did never hear about **PHP**?
- used PHP in a web application?
- wrote PHP applications > **10k** LOC?
- wrote PHP applications > **100k** LOC?
- uses Xdebug?
- is bored of these questions?

```
[derick@localhost derick]$ cat recursive.php
<?php
    function a() {
        a();
    }

    a();
?>
[derick@localhost derick]$ php recursive.php
Segmentation fault
```

**Fatal error:** Maximum function nesting level of '4' reached, aborting! in **/home/httpd/html/test/xdebug/infinite.php** on line **3**

### Call Stack

#	Function	Location
1	{main}()	/home/httpd/html/test/xdebug/infinite.php:0
2	a()	/home/httpd/html/test/xdebug/infinite.php:6
3	a()	/home/httpd/html/test/xdebug/infinite.php:3
4	a()	/home/httpd/html/test/xdebug/infinite.php:3

```
xdebug.max_nesting_level=4
```

```
array
  0 => null
  1 => 3.1415926
  2 =>
    array
      'dutch' =>
        object(locale)[1]
          public 'lang' => 'nl'
          public 'variation' => 'NL'
          public 'charsets' =>
            array
              0 => 'iso-8859-1'
              1 => 'iso-8859-15'
          private 'mb_supported' => false
          private 'self' =>
            &object(locale)[1]
          protected 'id' => 1
```

**Warning: `chdir()` [[function.chdir](#)]: No such file or directory (errno 2) in `/dat/docs/magazine/phpa/code/xdebug/listing2.php` on line 34**

## Call Stack

#	Function	Location
1	{main}()	/dat/docs/magazine/phpa/code/xdebug/listing2.php:0
2	change_dir( 'foo123' )	/dat/docs/magazine/phpa/code/xdebug/listing2.php:37
3	<a href="#">chdir</a> ( '/foo123' )	/dat/docs/magazine/phpa/code/xdebug/listing2.php:34

**Warning: `chdir()` [[function.chdir](#)]: No such file or directory (errno 2) in `/dat/docs/magazine/phpa/code/xdebug/listing2.php` on line 34**

### Call Stack

#	Function	Location
1	{main}()	/dat/docs/magazine/phpa/code/xdebug/listing2.php:0
2	change_dir( 'foo123' )	/dat/docs/magazine/phpa/code/xdebug/listing2.php:37
3	<a href="#">chdir</a> ( '/foo123' )	/dat/docs/magazine/phpa/code/xdebug/listing2.php:34

### Variables in local scope (#2)

Variable	Value
\$path =	'/foo123'
\$dir =	'foo123'

```
; enable showing local vars in error messages
xdebug.show_local_vars=1
```

**Warning: chdir(): No such file or directory (errno 2) in /dat/docs/magazine/phpa/code/xdebug/listing2.php on line 36**

### Call Stack

#	Function	Location
1	{main}()	/dat/docs/magazine/phpa/code/xdebug/listing2.php:0
2	change_dir('foo123')	/dat/docs/magazine/phpa/code/xdebug/listing2.php:39
3	chdir('/foo123')	/dat/docs/magazine/phpa/code/xdebug/listing2.php:36

### Dump \$\_GET

`$_GET[42] = 'the answer to life, the universe and everything'`

```
show all GET variables in error messages
xdebug.dump.GET=*
; show the login and password POST variables
xdebug.dump.POST=login,password
; show the PHP's session cookie
xdebug.dump.COOKIE=PHPSESSID
```

```

[derick@kossu] - mc - /dat/ez.no-33/var/ezno
TRACE START [2004-06-03 13:52:21]
0.0032 235760 -> {main}() /dat/ez.no-33/index.php:0
0.0036 235840 -> microtime() /dat/ez.no-33/index.php:34
0.0037 235856 -> ob_start() /dat/ez.no-33/index.php:35
0.0037 278880 -> error_reporting(2047) /dat/ez.no-33/index.php:74
0.0084 611512 -> include_once(/dat/ez.no-33/lib/ezutils/classes/ezdebug.php) /dat/ez.no-33/index.php:77
0.0106 766520 -> include_once(/dat/ez.no-33/lib/ezutils/classes/ezsys.php) /dat/ez.no-33/lib/ezutils/cl
0.0106 766568 -> define('EZ_SYS_DEBUG_INTERNALS', FALSE) /dat/ez.no-33/lib/ezutils/classes/ezsys.php:
0.0107 765856 -> define('EZ_LEVEL_NOTICE', 1) /dat/ez.no-33/lib/ezutils/classes/ezdebug.php:85
0.0107 765872 -> define('EZ_LEVEL_WARNING', 2) /dat/ez.no-33/lib/ezutils/classes/ezdebug.php:86
/tmp/trace.2043925204.xt [R0] 2.1 Top
0.0456 1698456 -> eztextcodec::internalcharset() /dat/ez.no-33/lib/ezil8n/classes/eztextcodec.php:581
0.0456 1698456 -> ezcharsetinfo::realcharsetcode('iso-8859-15') /dat/ez.no-33/lib/ezil8n/classes/eztex
0.0457 1698424 -> ezcharsetinfo::aliastable() /dat/ez.no-33/lib/ezil8n/classes/ezcharsetinfo.php:212
0.0457 1702584 -> is_array(array('ascii' => 'us-ascii', 'latin1' => 'iso-8859-1', 'latin2' => 'is
0.0461 1702664 -> strtolower('iso-8859-15') /dat/ez.no-33/lib/ezil8n/classes/ezcharsetinfo.php:213
0.0500 1966816 -> include_once(/dat/ez.no-33/lib/ezlocale/classes/ezlocale.php) /dat/ez.no-33/index.php:24
0.0501 1966904 -> define('EZ_LOCALE_DEBUG_INTERNALS', FALSE) /dat/ez.no-33/lib/ezlocale/classes/ezlocale
0.0501 1965888 -> ezini::instance() /dat/ez.no-33/index.php:246
0.0502 1966224 -> get_class(class ezini { var $Charset = 'iso-8859-1'; var $BlockValues = array ('Databa
0.0510 1966840 -> ezini->variable('RegionalSettings', 'Debug') /dat/ez.no-33/index.php:247
/tmp/trace.2043925204.xt [R0] 349,5 1%
    
```

```

xdebug.auto_trace=1 ; enable tracing
xdebug.trace_output_dir=/tmp ; trace output directory
xdebug.trace_output_name=crc32 ; tracefile extension
xdebug.collect_params=0 ; parameter collection
xdebug.collect_includes=1 ; collecting includes
xdebug.extended_info=0 ; extended code generation
xdebug.show_mem_delta=1 ; memory difference
    
```

## ~/.vim/filetype.vim

```
augroup filetypedetect
au BufNewFile,BufRead *.xt setf xt
augroup END
```

## ~/.vim/syntax/xt.vim

(<http://xdebug.org/files/xt.vim>)

```
if version < 600
  syntax clear
elseif exists("b:current_syntax")
  finish
endif

syn match begin          "^TRACE START"
syn match end            "^TRACE END"
syn match date           "\[.*\]"

syn match min_memory    "+\d\+"
syn match pls_memory    "-\d\+"
syn match nll_memory    "+0"

syn match level         "->"
syn match lineno        ":\d\+$"

syn match methodcall    "\k\+-->"
syn match staticcall    "\k\+::"
syn match functionb     "\k\+("
syn match functione     ") "

syn match main          "{main}()"
syn match include       "include\(_once\)\=('.\+')
```

## Execution trace to file

Automatic readable format

```

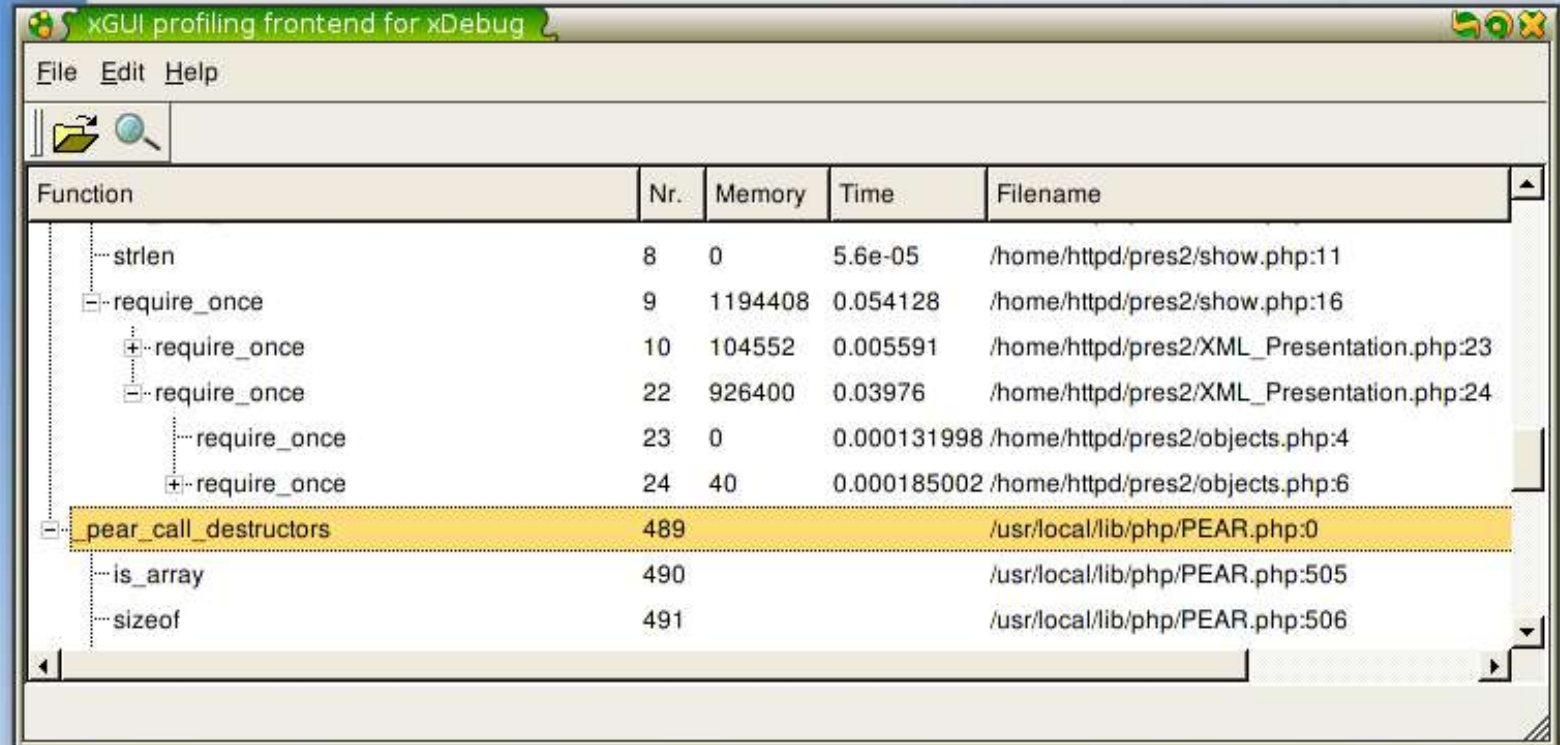
[derick@kossu]
File Edit View Terminal Tabs Help
Version: 2.0.0dev
TRACE START [2004-08-28 21:12:37]
1 0 0 0.001881 57336 {main} 1 /home/httpd/pres2/show.php 0
2 1 0 0.002255 57336 error_reporting 0 /home/httpd/pres2/show.php 2
2 1 1 0.002332 57344
2 2 0 0.002862 64360 require_once 1 /home/httpd/pres2/config.php /home/httpd
2 2 1 0.003043 65576
2 3 0 0.003120 62840 compact 0 /home/httpd/pres2/show.php 7
2 3 1 0.003225 63816
2 4 0 0.003459 66824 require_once 1 /home/httpd/pres2/sniff.php /home/httpd/pre
3 5 0 0.003563 66832 strstr 0 /home/httpd/pres2/sniff.php 4
3 5 1 0.003641 66832
2 4 1 0.003696 66888
2 6 0 0.003748 64160 set_time_limit 0 /home/httpd/pres2/show.php 10
2 6 1 0.003855 64160
2 7 0 0.003893 64160 strlen 0 /home/httpd/pres2/show.php 11
2 7 1 0.003949 64160
2 8 0 0.005437 99816 require_once 1 /home/httpd/pres2/XML_Presentation.php /ho
3 9 0 0.010074 143232 require_once 1 /usr/local/lib/php/XML/Parser.php /home/h
4 10 0 0.014375 250952 require_once 1 /usr/local/lib/php/PEAR.php /usr/local/lib/
5 11 0 0.014486 250992 define 0 /usr/local/lib/php/PEAR.php 25
5 11 1 0.014576 250992
/tmp/trace.2043925204.xt [R0] 1,1 Top

```

```

xdebug.auto_trace=1 ; enable tracing
xdebug.trace_format=1 ; selects computerized format
xdebug.trace_options=0 ; sets extra option (1 = append)

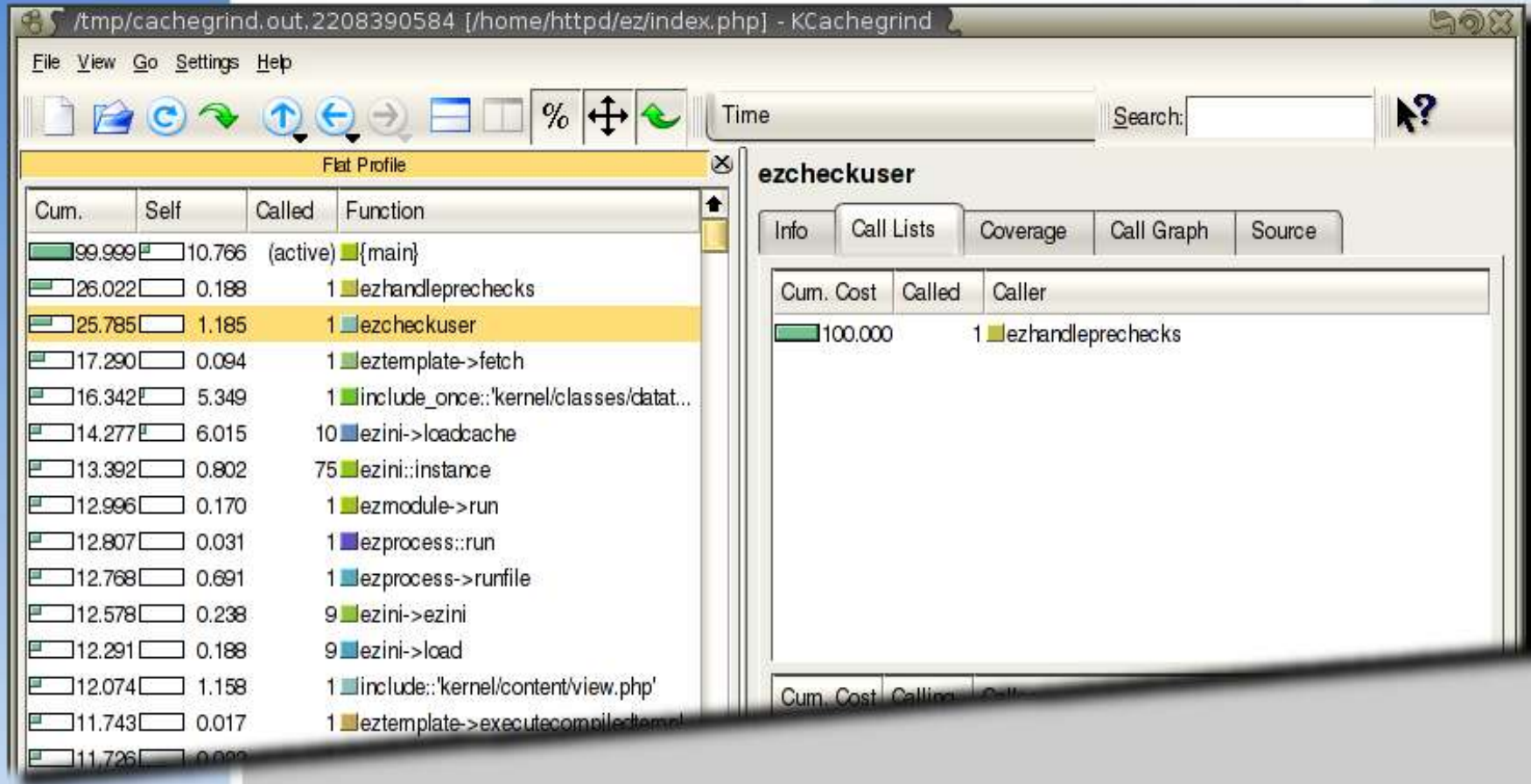
```



The screenshot shows a window titled "xGUI profiling frontend for xDebug". The window contains a table with the following columns: Function, Nr., Memory, Time, and Filename. The table lists various PHP functions and their call counts, memory usage, and execution times. The function "pear\_call\_destructors" is highlighted in yellow.

Function	Nr.	Memory	Time	Filename
strlen	8	0	5.6e-05	/home/httpd/pres2/show.php:11
require_once	9	1194408	0.054128	/home/httpd/pres2/show.php:16
+ require_once	10	104552	0.005591	/home/httpd/pres2/XML_Presentation.php:23
- require_once	22	926400	0.03976	/home/httpd/pres2/XML_Presentation.php:24
- require_once	23	0	0.000131998	/home/httpd/pres2/objects.php:4
+ require_once	24	40	0.000185002	/home/httpd/pres2/objects.php:6
- pear_call_destructors	489			/usr/local/lib/php/PEAR.php:0
- is_array	490			/usr/local/lib/php/PEAR.php:505
- sizeof	491			/usr/local/lib/php/PEAR.php:506

Still in development, and not yet public.  
Qt, multi-platform (Linux, MacOSX, Windows).



The screenshot shows the KCacheGrind application interface. The main window displays a 'Flat Profile' table with columns for Cumulative cost, Self cost, Called count, and Function name. The 'ezcheckuser' function is highlighted in yellow. To the right, a 'Call Lists' panel for 'ezcheckuser' shows a single entry: 'ezhandleprechecks' called once.

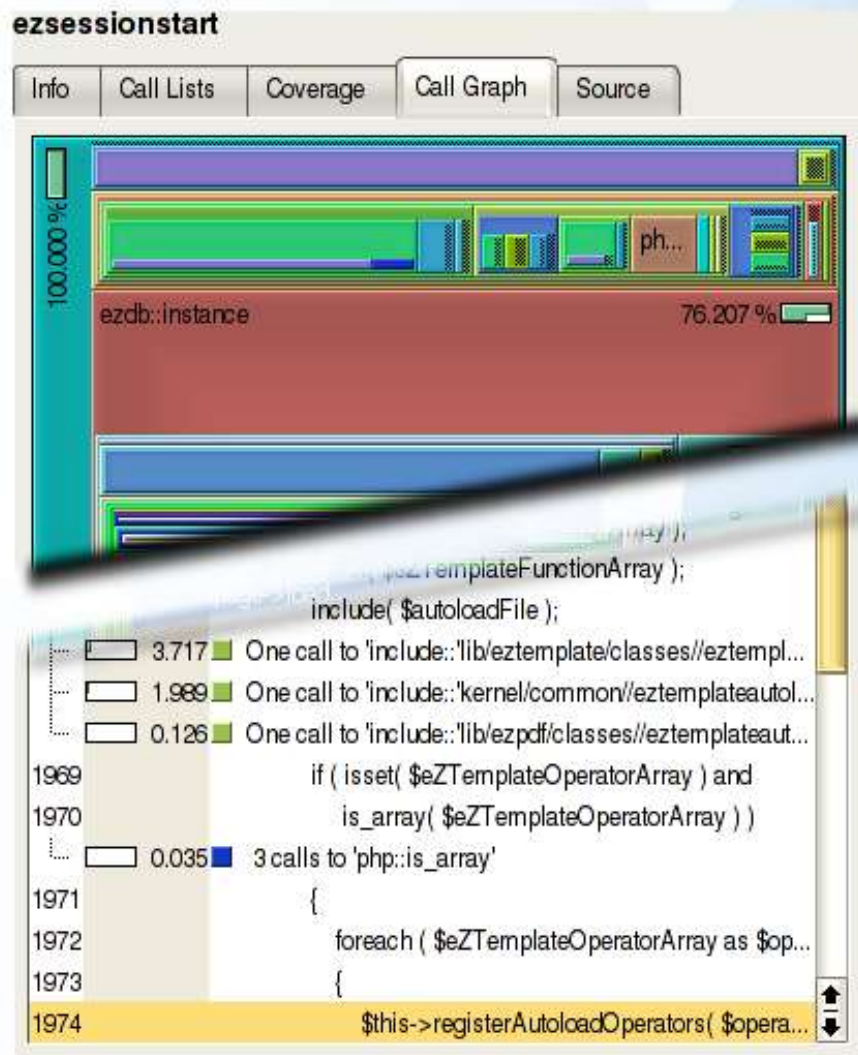
Cum.	Self	Called	Function
99.999	10.766	(active)	{main}
26.022	0.188	1	ezhandleprechecks
25.785	1.185	1	ezcheckuser
17.290	0.094	1	eztemplate->fetch
16.342	5.349	1	include_once:'kernel/classes/datat...
14.277	6.015	10	ezini->loadcache
13.392	0.802	75	ezini::instance
12.996	0.170	1	ezmodule->run
12.807	0.031	1	ezprocess::run
12.768	0.691	1	ezprocess->runfile
12.578	0.238	9	ezini->ezini
12.291	0.188	9	ezini->load
12.074	1.158	1	include:'kernel/content/view.php'
11.743	0.017	1	eztemplate->executecompiledtempl
11.726	0.000		

Cum. Cost	Called	Caller
100.000	1	ezhandleprechecks

```
xdebug.profiler_enable=1           ; enable profiler
xdebug.profiler_output_dir=/tmp     ; output directory
xdebug.profiler_output_name=crc32   ; file extension
```

- Call graph
  - Area shows time spend
  - Stacked to show callees
- 
- Source annotations
  - Number of calls
  - Total time per function



## Function to test (bigassfunction.php):

```
<?php
function test_parse_time($f)
{
    if (preg_match("/[0-9]{12}/", $f)) {
        $type = "YYYYMMDDHHii";
    }
    else if (preg_match("/[0-9]{8}\ [0-9]{4}/", $f)) {
        $type = "YYYYMMDD HHii";
    }
    else if (preg_match("/[0-9]{4}-[0-9]{2}-[0-9]{2}/", $f)) {
        $type = "YYYY-MM-DD";
    }
    else {
        $type = "*unknown*";
    }

    echo "This is a $type format.\n";
}
?>
```

## Test driver (basic.t):

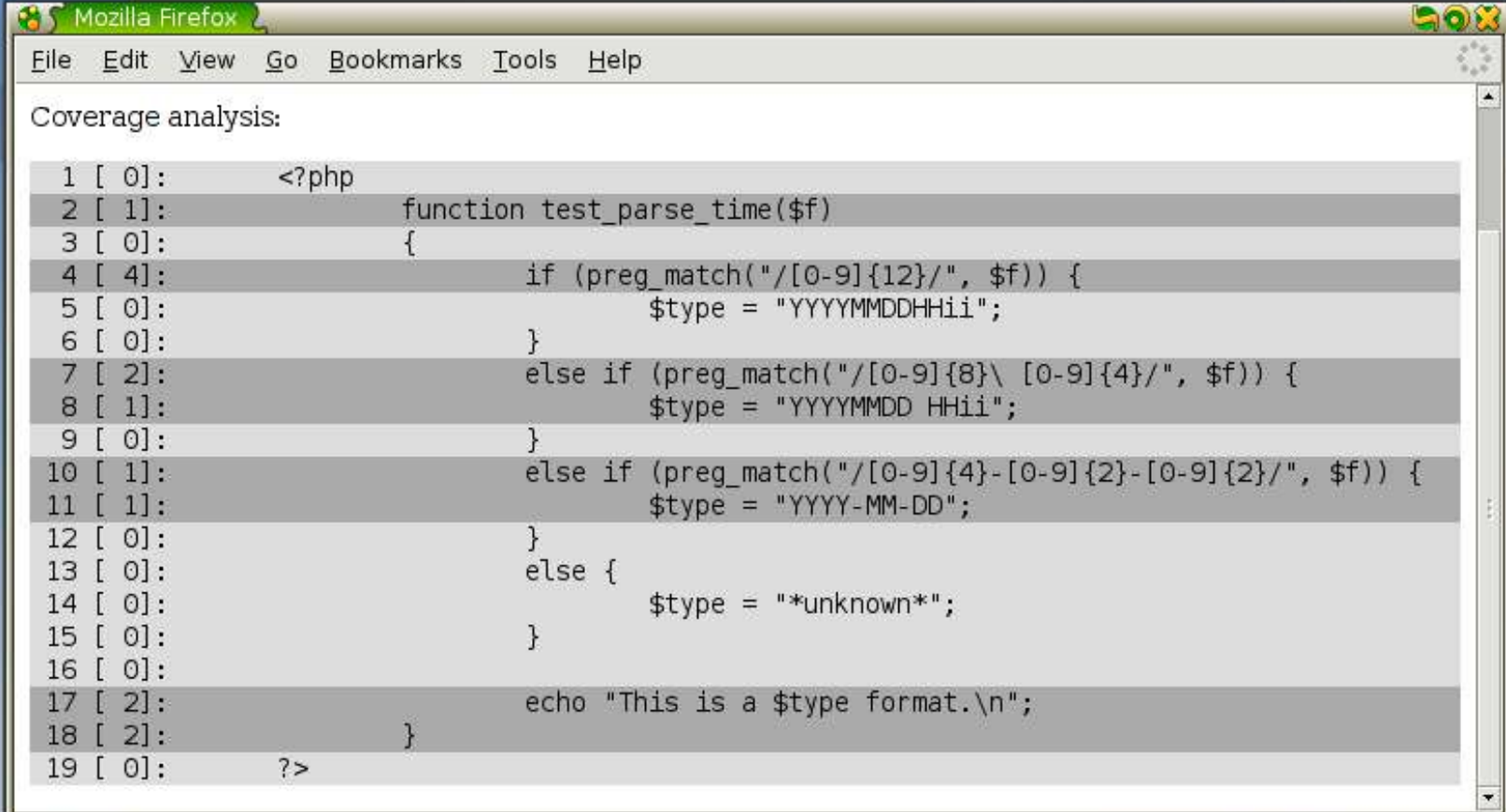
```
<?php
$text = realpath("tests/bigassfunction.php");
include $text;

$formats = array('2004-08-17 21:20', '20040817 2120');
foreach ($formats as $format) {
    test_parse_time($format);
}
?>
```

```
<?php
echo "Output:\n\n";
xdebug_start_code_coverage();
include 'tests/basic.t';
$cc = xdebug_get_code_coverage();
xdebug_stop_code_coverage();

$clean = array();
$clean[$test] = $cc[$test];

foreach ($clean as $file => $cc) {
    echo "<pre>";
    $fc = file($file);
    $line_nos = array_keys($cc);
    foreach ($fc as $ln => $line) {
        if (in_array($ln + 1, $line_nos)) {
            $bgc = "aaaaaa";
        } else {
            $bgc = "dddddd";
        }
        echo "<div style='background-color: #\$bgc'>";
        echo @sprintf("%3d [%2d]:\t", $ln + 1, $cc[$ln + 1]);
        echo htmlspecialchars($line);
        echo "</div>";
    }
    echo "</pre>";
}
?>
```



Coverage analysis:

```
1 [ 0]:      <?php
2 [ 1]:          function test_parse_time($f)
3 [ 0]:          {
4 [ 4]:              if (preg_match("/[0-9]{12}/", $f)) {
5 [ 0]:                  $type = "YYYYMMDDHHii";
6 [ 0]:              }
7 [ 2]:              else if (preg_match("/[0-9]{8}\ [0-9]{4}/", $f)) {
8 [ 1]:                  $type = "YYYYMMDD HHii";
9 [ 0]:              }
10 [ 1]:             else if (preg_match("/[0-9]{4}-[0-9]{2}-[0-9]{2}/", $f)) {
11 [ 1]:                 $type = "YYYY-MM-DD";
12 [ 0]:             }
13 [ 0]:             else {
14 [ 0]:                 $type = "*unknown*";
15 [ 0]:             }
16 [ 0]:
17 [ 2]:             echo "This is a $type format.\n";
18 [ 2]:         }
19 [ 0]:     ?>
```

```
xdebug_start_code_coverage()
xdebug_stop_code_coverage()
xdebug_get_code_coverage()
```

- PHP 3 debugger mode
  - Port of the old remote debug dump interface for PHP 3
  - Available in Xdebug 1.x and 2.x
- GDB debugger mode
  - GDB like environment
  - Two output possibilities: text and XML
  - Available in Xdebug 1.3.x and 2.x
- DBGp debugger mode
  - Language independent
  - Plain text in, XML out
  - Available in Xdebug 2.x

```

derick@localhost:/dat/dev/php/xdebug/debugclient
File Edit View Terminal Go Help
[derick@localhost debugclient]$ ./debugclient
Xdebug GDB emulation client (0.8.0)
Copyright 2002-2003 by Derick Rethans.

Waiting for debug server to connect.
Connect
This is Xdebug version 1.3.0dev.
Copyright 2002 by Derick Rethans, JDI Media Solutions.
(init /home/httpd/html/test/xdebug/appl.php) bre appl.php:11 preg_match("/xdebug
/", $file)
1 Breakpoint set on file:line combination.
(init /home/httpd/html/test/xdebug/appl.php) run
Starting program: /home/httpd/html/test/xdebug/appl.php
Breakpoint, search_dir('/home/httpd')
    at /home/httpd/html/test/xdebug/appl.php:20
11         if (!in_array($file, array('.', '..', 'p
roc', 'derick'))) {

(cmd) bt full
#1 search_dir ('/home/httpd')
    at /home/httpd/html/test/xdebug/appl.php:20
$file = 'www.xdebug.org'
$path = '/home/httpd'
$dirs = array ()
$dir = resource(48) of type (stream)

#2 search_dir ('/home')
    at /home/httpd/html/test/xdebug/appl.php:20
$file = FALSE
$path = '/home'

```

- Input commands the same
- `option response_format 1`
- Output in 'proprietary' XML
- Easier parsing for debug clients... but not for people

```
<xdebug><break><breakpoint><function><name>search_dir</name><params>'/home/httpd  
/www.xdebug.org/html/images'</params></function><file>/home/httpd/html/test/xdeb  
ug/appl.php</file><line>20</line></breakpoint><line file='/home/httpd/html/test/  
xdebug/appl.php' no='11'>                                if (!in_array($file, arr  
ay('.', '..', 'proc', 'derick')) {</line></break></xdebug>
```

## DBGp mode

*DeBuGger Protocol*

- Input commands as "command line parameters"
- `property_get -n $foo -d 2 -i 81`
- Output data as **documented XML**:

```
data_length  
[NULL]  
<response command="command_name" transaction_id="transaction_id"/>  
[NULL]
```

- Cross-language: PHP, Python, Perl...
- Supported in Xdebug 2
- Clients: Komodo, Maguma Workbench

- Protects against infinite recursion
- **Colored** `var_dump()`
- Stacktraces
- Functiontraces
- parameters to functions
- function/methodname, classname
- file:line locations
- memory allocation
- Profiling
- Script execution analysis
- "Remote" debugging



Xdebug site: <http://xdebug.org>

DBGp specification:

<http://xdebug.org/docs-dbgp.php>

Questions?: [derick@xdebug.org](mailto:derick@xdebug.org)