



Xdebug in a nutshell

Conférence PHP Québec

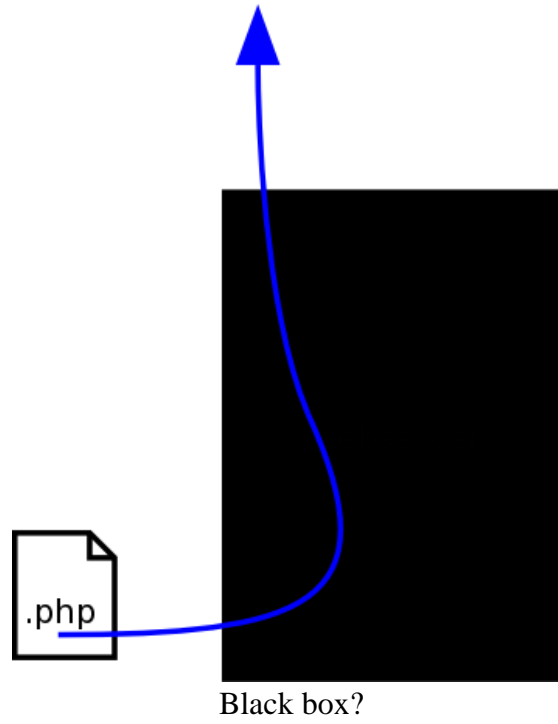
March 26th, 2004. Montreal, Canada

Derick Rethans <derick@php.net>

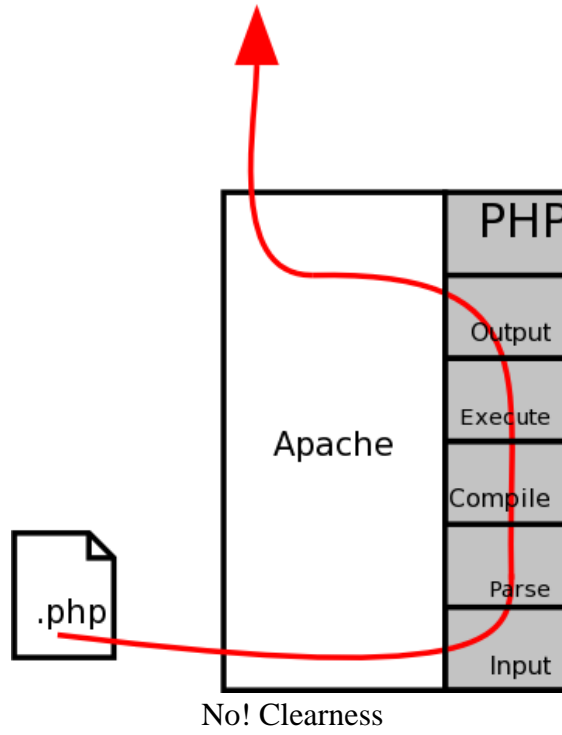
<http://pres.derickrethans.nl/xdebug-montreal>

?

- o PHP registers handlers for:
- o Mime types:
 - o application/x-httpd-php
 - o application/x-httpd-php-source
- o Others:
 - o text/html (Apache xbithack)
 - o php-script (Apache 2)
- o On a request, Apache:
 - o checks if there is a handler for the mime type
 - o opens the file
 - o fills a meta data record
 - o hands PHP a filepointer



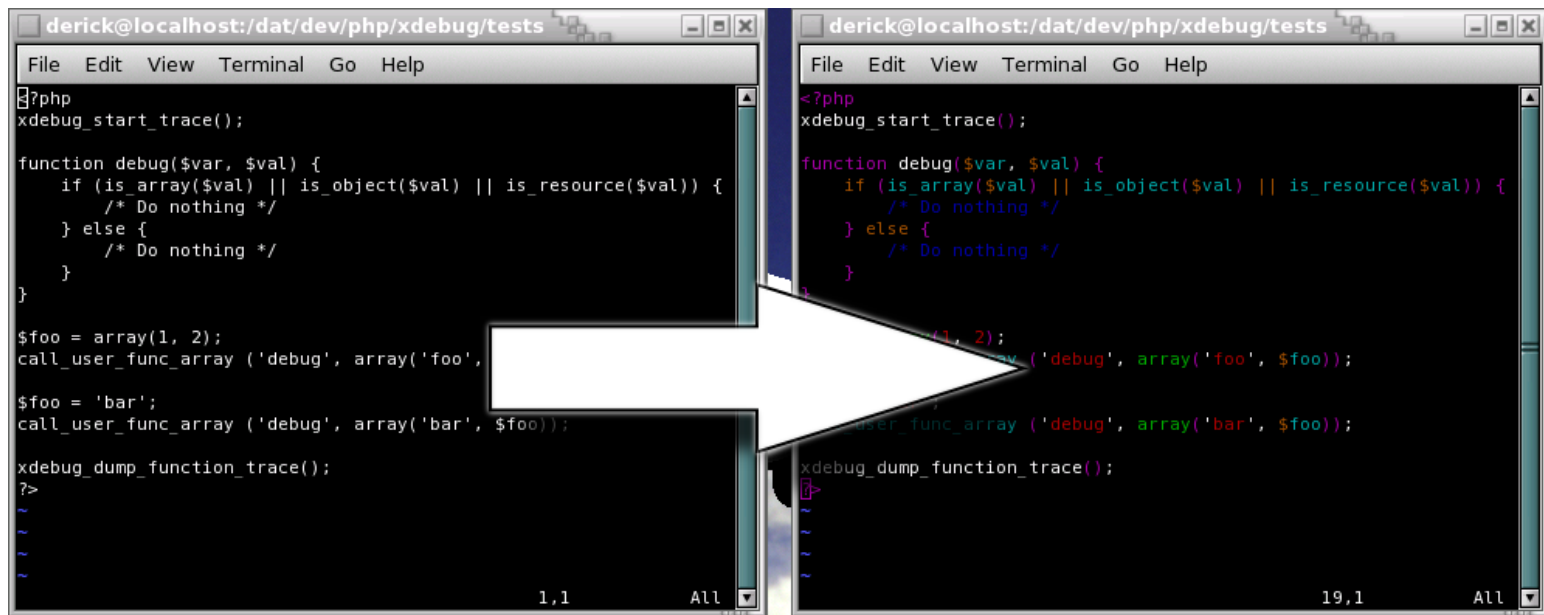
Black box?



- o Lexical analyze script source
- o Divide into logical blocks of characters
- o Give special blocks a meaning
- o flex

Our friend Mr Parse Error:

```
Parse error: parse error,  
unexpected T_CLASS, expecting ',' or ';' in - on line 2
```



Syntax highlighting = Tokenizing

- o The building blocks of a script
- o Strings, variables, keywords
- o Action rules:

```
<ST_IN_SCRIPTING>"::" {  
    return T_PAAMAYIM_NEKUDOTAYIM;  
}  
  
<ST_IN_SCRIPTING>{DNUM}|{EXPONENT_DNUM} {  
    zendlval->value.dval = strtod(yytext, NULL);  
    zendlval->type = IS_DOUBLE;  
    return T_DNUMBER;  
}
```

Tokenizer extension:

magic.php:

```
<?php
    function apply_magic(&$a) {
        $a = $a + rand(0,3);
    }
? >
```

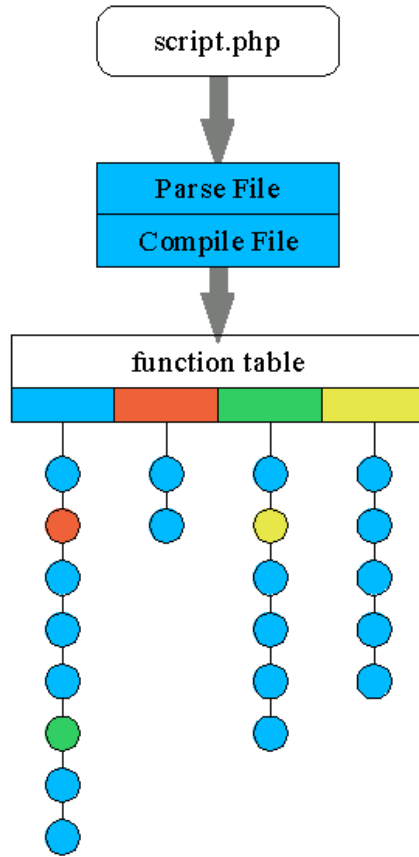
tokenize.php:

```
<?php
    foreach (token_get_all($script) as $token) {
        if (count($token) == 2) {
            printf ("-25s [s]\n", token_name($token[0]), $token[1]);
        } else {
            printf ("-25s [s]\n", "", $token[0]);
        }
    }
? >
```

```
<?php
    function apply_magic(&$a) {
        $a = $a + rand(0,3);
    }
? >
```

T_OPEN_TAG	[<?php\n]
T_WHITESPACE	[]
T_FUNCTION	[function]
T_WHITESPACE	[]
T_STRING	[apply_magic]
	[(
	[&]
T_VARIABLE	[\$a]
	[)]
T_WHITESPACE	[]
	[{
T_WHITESPACE	[\n]
T_VARIABLE	[\$a]
T_WHITESPACE	[]
	[=]
T_WHITESPACE	[]
T_VARIABLE	[\$a]
T_WHITESPACE	[]
	[+]
T_WHITESPACE	[]
T_STRING	[rand]
	[(
T_LNUMBER	[0]
	[,]
T_LNUMBER	[3]
	[)]
	[;]
T_WHITESPACE	[\n]
	[}]
T_WHITESPACE	[\n]
T_CLOSE_TAG	[? >]

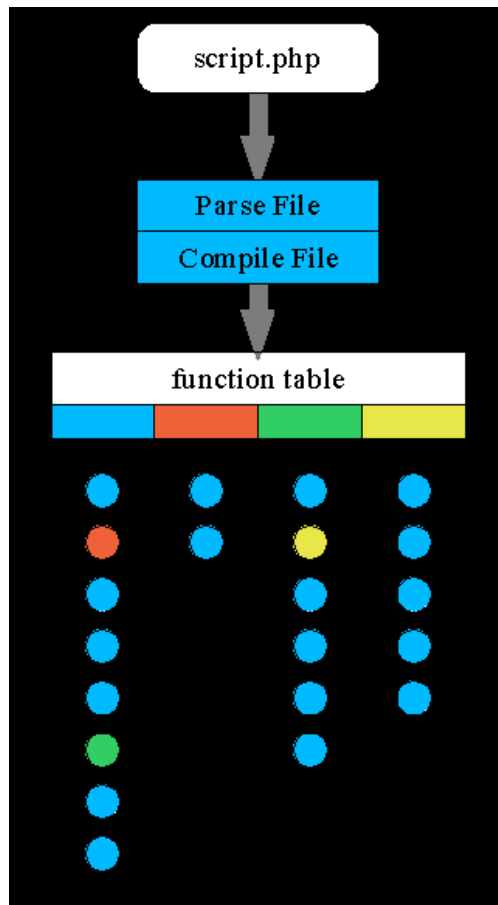
- o Interpreting tokens
- o Generating execution units
- o Generating class and function tables



- o Oparrays
- o Compiled code
- o One for every script element

- o Opcode
- o Basic execution unit
- o Two operands
- o One result

- o Executor executes opcodes
- o 116 or 140 different opcodes
- o Per file execution



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

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

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

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

```
array
  0 => 3
  'apel' =>
    object(monkey)
      'name' => 'Magumbi'
      'color' => null
  'ape2' =>
    object(monkey)
      'name' => 'Suzan'
      'color' => null
  'apies' =>
    array
      0 =>
        object(monkey)
          'name' => 'Magumbi'
          'color' => null
      1 =>
        object(monkey)
          'name' => 'Suzan'
          'color' => null
  'types' =>
    array
      'boolean' => true
      'int' => 39
      'float' => 3.141592654
      'string' => 'foo'
      'null' => null
```

Warning: getimagesize(presentations/slides/xdebug/stacktrace.png): failed to open stream: No such file or directory in /dat/dev/pres2/display.php on line 293

Call Stack

#	Function	Location
1	{main}()	/dat/dev/pres2/show.php:0
2	_presentation->display()	/dat/dev/pres2/show.php:98
3	html->_presentation(class _presentation {var \$title = 'Xdebug in a nutshell'; var \$navmode = 'html'; var \$mode = 'html'; var \$navsize = NULL; var \$template = 'php'; var \$jskeyboard = TRUE; var \$logo1 = 'presentations/slides/xdebug/maguma.png'; var \$logo2 = NULL; var \$basefontsize = '12pt'; var \$backgroundcol = FALSE; var \$backgroundfixed = '1'; var \$backgroundimage = FALSE; var \$backgroundrepeat = FALSE; var \$navbarbackground = '#227722'; var \$navbartopiclinks = TRUE; var \$navbarheight = '6em'; var \$examplebackground = '#cccccc'; var \$outputbackground = '#eeee33'; var \$shadowbackground = '#777777'; var \$stylesheet = 'presentations/slides/xdebug/style.css'; var \$logoimage1url = 'http://pres//index.php'; var \$animate = FALSE; var \$navcolor = '#f1fbff'; var \$topic = 'Internals'; var \$event = 'International PHP Conference'; var \$location = 'Frankfurt, Germany'; var \$date = 'November 5th, 2003'; var \$speaker = 'Derick Rethans'; var \$email = 'derick@maguma.com'; var \$url = 'http://pres.derickrethans.nl/xdebug-frankfurt'; var \$slides = array (0 => class _pres_slide {var \$filename = 'slides/xdebug/title.xml'}, 1 => class _pres_slide {var \$filename = 'slides/xdebug/how-started.xml'}, 2 => class _pres_slide {var \$filename = 'slides/xdebug/about.xml'}, 3 => class _pres_slide {var \$filename = 'slides/xdebug/infinite.xml'}, 4 => class _pres_slide {var \$filename = 'slides/xdebug/vardump.xml'}, 5 => class _pres_slide {var \$filename = 'slides/xdebug/stacktrace.xml'}, 6 => class _pres_slide {var \$filename = 'slides/xdebug/profile.xml'}, 7 => class _pres_slide {var \$filename = 'slides/xdebug/gathering-data.xml'}, 8 => class _pres_slide {var \$filename = 'slides/xdebug/hooks1.xml'}, 9 => class _pres_slide {var \$filename = 'slides/xdebug/execute.xml'}, 10 => class _pres_slide {var \$filename = 'slides/xdebug/hooks2.xml'}, 11 => class _pres_slide {var \$filename = 'slides/xdebug/client-xdebugc.xml'}, 12 => class _pres_slide {var \$filename = 'slides/xdebug/client-macosx.xml'}, 13 => class _pres_slide {var \$filename = 'slides/xdebug/client-studio.xml'}})	/dat/dev/pres2/objects.php:124
4	_image->display()	/dat/dev/pres2/display.php:62
5	html->_image(class _image {var \$filename = 'stacktrace.png'; var \$align = 'center'; var \$marginleft = 'auto'; var \$marginright = 'auto'; var \$effect = ""; var \$width = ""; var \$height = ""})	/dat/dev/pres2/objects.php:124
6	getimagesize ('presentations/slides/xdebug/stacktrace.png')	/dat/dev/pres2/display.php:293

Notice: Undefined variable: e in /home/httpd/html/crap/xdebug-local-scope.php on line 7		
Call Stack		
#	Function	Location
1	{main}()	/home/httpd/html/crap/xdebug-local-scope.php:0
2	a()	/home/httpd/html/crap/xdebug-local-scope.php:10
Variables in local scope		
Variable	Value	
\$d =	object (stdClass)	
\$a =	5	
\$c =	array 0 => 5 1 => 36	
\$b =	6	

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

```

TRACE START [2004-03-10 15:25:18]
0.0019 88880 +88880 -> {main}() /home/httpd/pres2/show.php:0
0.0021 88880 +0 -> error_reporting(2047) /home/httpd/pres2/show.php:2
0.0026 97680 +8800 -> require_once('config.php') /home/httpd/pres2/show.php:4
0.0027 95392 -2288 -> compact('presentationDir', 'baseDir', 'showScript', 'helpPage', 'baseFontSize', 'flashF
0.0030 100168 +4776 -> require_once('sniff.php') /home/httpd/pres2/show.php:8
0.0031 100152 -16 -> strpos('Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.5) Gecko/20040130 Firebird/0.7',
0.0031 96720 -3432 -> set_time_limit(0) /home/httpd/pres2/show.php:10
0.0032 96720 +0 -> strlen('/xdebug-montreal/21') /home/httpd/pres2/show.php:11
0.0047 138728 +42008 -> require_once('XML/Presentation.php') /home/httpd/pres2/show.php:16
0.0087 195592 +56864 -> require_once('XML/Parser.php') /home/httpd/pres2/XML_Presentation.php:23
0.0137 370512 +174920 -> require_once('PEAR.php') /usr/local/lib/php/XML/Parser.php:21
0.0137 370512 +0 -> define('PEAR_ERROR_RETURN', 1) /usr/local/lib/php/PEAR.php:24
0.0138 370536 +24 -> define('PEAR_ERROR_PRINT', 2) /usr/local/lib/php/PEAR.php:25
0.0139 370560 +24 -> define('PEAR_ERROR_TRIGGER', 4) /usr/local/lib/php/PEAR.php:26
0.0139 370504 -56 -> define('PEAR_ERROR_DIE', 8) /usr/local/lib/php/PEAR.php:27
0.0140 370456 -48 -> define('PEAR_ERROR_CALLBACK', 16) /usr/local/lib/php/PEAR.php:28
0.0140 370400 -56 -> define('PEAR_ERROR_EXCEPTION', 32) /usr/local/lib/php/PEAR.php:29
0.0141 370424 +24 -> function_exists('version_compare') /usr/local/lib/php/PEAR.php:30
0.0141 370440 +16 -> zend_version() /usr/local/lib/php/PEAR.php:31
0.0141 370456 +16 -> version_compare('2.0.0-dev', '2-dev', 'ge') /usr/local/lib/php/PEAR.php:31

```

```

xdebug.auto_trace=1 ; enable tracing
xdebug.trace_output_dir=/tmp ; configure trace output directory
xdebug.trace_output_name=crc32 ; configure tracefile extension
xdebug.collect_params=0 ; disable parameter collection
xdebug.collect_includes=1 ; enable collecting filenames to include
xdebug.extended_info=0 ; disable 'extended' code generation
xdebug.show_mem_delta=1 ; show difference in memory usage before
; and after call

```

~/vim/filetype.vim

```
augroup filetypepedetect
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\(_once\)\)=('.\+')"

if version >= 508 || !exists("did_diff_syntax_inits")
  if version < 508
    let did_diff_syntax_inits = 1
    command -nargs=+ HiLink hi link <args>
  else
    command -nargs=+ HiLink hi def link <args>
  endif

  HiLink begin          Label
  HiLink end            Label
  HiLink date           Label
  HiLink level          SpecialChar

  HiLink min_memory    Constant
  HiLink pls_memory    Structure
  HiLink nll_memory    Comment

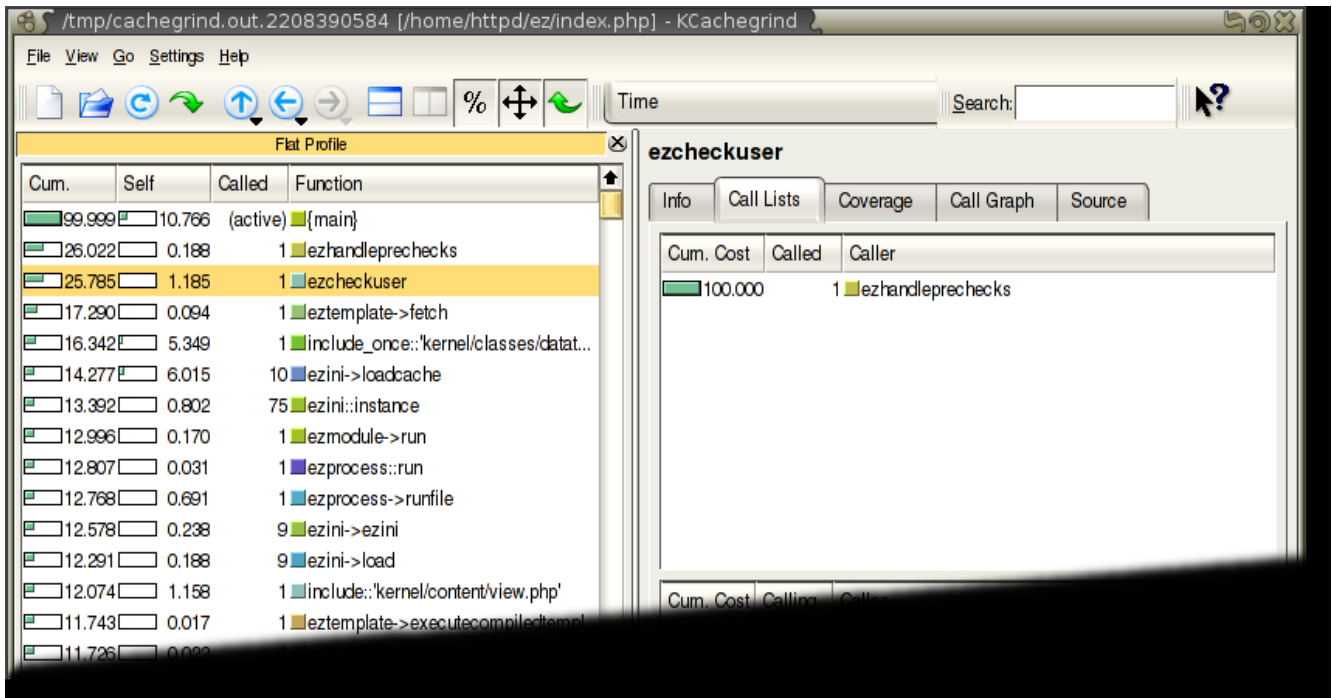
  HiLink main           Structure
  HiLink include        Structure

  HiLink lineno        Delimiter
  HiLink methodcall    Function
  HiLink staticcall    Function
  HiLink functionb     Function
  HiLink functione     Function

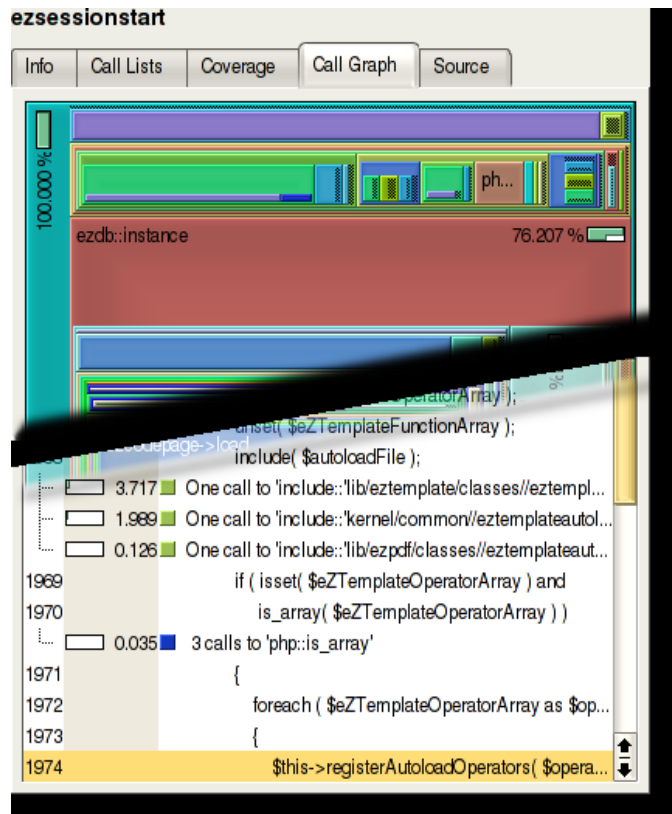
  delcommand HiLink
endif

let b:current_syntax = "xt"

" vim: ts=8 sw=2
```



```
xdebug.profiler_enable=1           ; enable profiler
xdebug.profile_output_dir=/tmp     ; configure profile output directory
xdebug.profile_output_name=crc32   ; configure profilefile extension
```



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

- By hooking into the execution of functions
- By reading the memory usage
- By scanning compiled code (for used vars)
- By remembering locations of variables



- o Functions and classnames
- o Normal functions (strlen())
- o Class members (\$obj->strlen())
- o Static class members (class::strlen())
- o include/require/eval
- o Parameters
- o From argument stack
- o Converted to strings

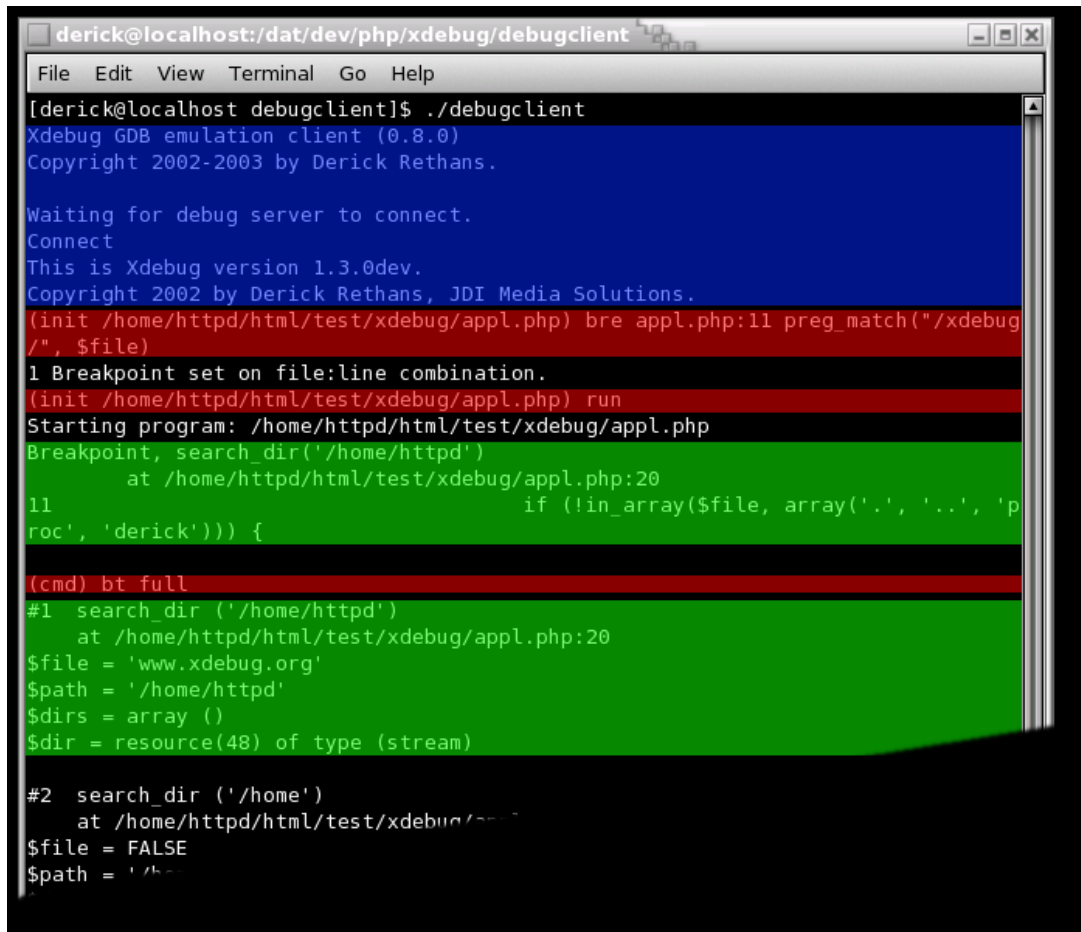
- o Memory footprint
- o Script memory
- o emalloc()
- o --enable-memory-limit
- o Changes since last function call
- o Time index
- o Simple profiling

- o Xdebug hooks into these events:



- o Executing user defined function
- o Executing internal function
- o Compiling a file
- o Executing a statement

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

- o Input commands the same
- o option response_format 1
- o Output in 'proprietary' XML
- o 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>
```

- o Input commands as "command line parameters"

- o `property_get -n $foo->property -d 2 -i 81`

- o Output data as documented XML:

```
data_length
```

```
[NULL]
```

```
<response command="command_name" transaction_id="transaction_id"/>
```

```
[NULL]
```

- o Cross-language: PHP, Python, Perl, Tcl, XSLT
- o Supported in Xdebug 2
- o Client implementation in Maguma Studio 2,
ActiveState's Komodo



These Slides: <http://pres.derickrethans.nl>

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

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

PHP: <http://www.php.net>

Questions?: derick@xdebug.org

Index

Questions	2
Input: Apache module	3
Introduction	4
Introduction (2)	5
Parsing	6
Parsing: Tokenize	7
Parsing: Tokens	8
Parsing: Example (1)	9
Parsing: Example (2)	10
Compiling	11
Compiling: Diagram	12
Compiling: Oparrays	13
Executing	14
Execution	15
How did it start?	16
Xdebug	17
Protection	18
New var_dump()	19
Stacktrace	20
Local variables	21
Execution trace to file	22
Vim Syntax	23
Profiling	24
Profiling (2)	25
Gathering data	26
Gathering debug data 1	27
Gathering debug data 2	28
Engine hooks	29
Remote Interfaces	30
GDB mode	31
GDB mode/XML	32
DBGp mode	33
Resources	34