

Zend and Extending PHP

PHP Forum 2002

December 10, 2002. Paris, France

Derick Rethans <d.rethans@jdimedia.nl>

- o Introduction to the engine
- o Parse
- o Compile
- o Execute
- o Extending PHP
- o ext_skel
- o resources
- o return values

- o Analyze script source
- o Divide into logical blocks
- o Give special blocks a meaning

Our friend Mr Parse Error:

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

- 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;  
}
```

tokenize.php:

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

$a = 5;
apply_magic($a);
echo "$a\n";
? >
```

- o Tokenizer extension
- o Refactoring

```
<?php
    function apply_magic(&$a) {
```

0	T_OPEN_TAG	[<?php]
1	T_WHITESPACE	[]
2	T_FUNCTION	[function]
3	T_WHITESPACE	[]
4	T_STRING	[apply_magic]
5		[(]
6		[&]
7	T_VARIABLE	[\$a]
8		[)]
9	T_WHITESPACE	[]
10		[{]

- 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

tokenize.php:

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

$a = 5;
apply_magic($a);
echo "$a\n";
? >
```

- o Disassembler: vld
- o Dumps oparray per element

script body:

number of ops: 12

```
0  NOP          0, 0, 0
1  FETCH_W     $0, 'a', 1
2  ASSIGN      $1, $0, '5'
3  FETCH_W     $2, 'a', 1
4  SEND_REF    0, $2, 1
5  DO_FCALL    $3, 'apply_magic', 0
6  INIT_STRING %4, 0, 0
7  FETCH_R     $5, 'a', 1
8  ADD_VAR     4, 4, $5
9  ADD_CHAR    4, 4, '10'
10 ECHO        0, %4, 0
11 RETURN     0, '1', 0
```

apply_magic() function:

number of ops: 10

```
0  FETCH_W      $0, 'a', 1
1  RECV         $0, '1', 0
2  FETCH_R      $2, 'a', 1
3  SEND_VAL     0, '0', 1
4  SEND_VAL     0, '3', 2
5  DO_FCALL     $3, 'rand', 0
6  ADD          %4, $2, $3
7  FETCH_W      $1, 'a', 1
8  ASSIGN       $5, $1, %4
9  RETURN       0, '', 0
```

- o Generated code not always optimal
- o Optimizing techniques:
 - o Removing NOPs
 - o Removing unnecessary jumps
 - o Concatenation of strings
 - o Dead-code elimination
 - o Cache function table lookups
 - o Function inlining

foo.php:

```
<?php
function foo() {
}

if (40 < 50) {
    echo "foo\n";
}
echo "end\n";
? >
```

generated opcodes:

0	NOP	0, 0, 0
1	IS_SMALLER	%0, '40', '50'
2	JMPZ	0, %0, 5
3	ECHO	0, 'foo\n', 0
4	JMP	0, 5, 0
5	ECHO	0, 'end\n', 0
6	RETURN	0, '1', 0

Eliminating the NOP and unnecessary jump:

before:

```
0  NOP                0, 0, 0
1  IS_SMALLER        %0, '40', '50'
2  JMPZ              0, %0, 5
3  ECHO              0, 'foo\n', 0
4  JMP               0, 5, 0
5  ECHO              0, 'end\n', 0
6  RETURN            0, '1', 0
```

after:

```
0  IS_SMALLER        %0, '40', '50'
1  JMPZ              0, %0, 3
2  ECHO              0, 'foo\n', 0
3  ECHO              0, 'end\n', 0
4  RETURN            0, '1', 0
```

Eliminating the NOP, unnecessary jump and unnecessary jump condition:

before:

```
0  NOP                0, 0, 0
1  IS_SMALLER        %0, '40', '50'
2  JMPZ              0, %0, 5
3  ECHO              0, 'foo\n', 0
4  JMP               0, 5, 0
5  ECHO              0, 'end\n', 0
6  RETURN            0, '1', 0
```

after:

```
0  ECHO              0, 'foo\n', 0
1  ECHO              0, 'end\n', 0
2  RETURN            0, '1', 0
```

- o Optimizers:
- o ZendOptimizer
- o PHP Accelerator and Zend Cache
- o Zend Encoder and ionCube encoder
- o When is it useful?
- o Large speed increases
- o Cached PHP scripts
- o Encoded PHP scripts

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

test.inc:

```
<?php
    function foo () {
        echo "bar\n";
    }
    echo "inc\n";
? >
```

test.php

```
<?php
    echo "foo\n";
    include "test.inc";
    foo();
    echo "more bar\n";
? >
```

The engine:

- o parses and compiles test.php
- o starts executing test.php
- o parses and compiles test.inc when the include() statement is encountered
- o starts executing test.inc
- o resumes executing test.php

protos

```
resource aa_init()
    Initialize the AA library

int aa_get_screen_height(resource c)
    Get the screen height

int aa_get_screen_width(resource c)
    Get the screen width

void aa_put_pixel(resource c, int x, int y, int value)
    Draw a pixel

void aa_render(resource c [, int contrast [, int dither [, int inversion [, int
randomval]]]])
    Render the image

string aa_get_all(resource c)
    Get the result as one string

array aa_get_lines(resource c)
    Get the result with one line in an array element

array aa_get_array(resource c)
    Get the result with one element per pixel
```

- o Go to the php4/ext directory
- o Execute the following command:

```
# ./ext_skel --extname=aalib --proto=/path/to/protofile
```

- o This produces the following source files:
- o config.m4
- o aalib.c
- o php_aalib.h

php_aalib.h

```

<html> <head> <title>php_aalib.h</title> </head> <body bgcolor="#ffffff" text="#000000">
<pre>
1 <font color="#444444"/>*/ 2
+-----+ 3 | PHP Version 4
| 4 +-----+ 5 |
Copyright (c) 1997-2002 The PHP Group | 6
+-----+ 7 | This source file is subject to
version 2.02 of the PHP license, | 8 | that is bundled with this package in the file LICENSE,
and is | 9 | available at through the world-wide-web at | 10 |
http://www.php.net/license/2_02.txt. | 11 | If you did not receive a copy of
the PHP license and are unable to | 12 | obtain it through the world-wide-web, please send a
note to | 13 | license@php.net so we can mail you a copy immediately. | 14
+-----+ 15 | Author:
| 16 +-----+ 17 18 $Id:
protos-result.xml,v 1.3 2002/12/09 09:48:03 derick Exp $ 19 */</font> 20 21 <font
color="0000ff">#ifndef PHP_AALIB_H</font> 22 <font color="0000ff">#define
PHP_AALIB_H</font> 23 24 extern <font color="#2040a0">zend_module_entry</font> <font
color="#2040a0">aalib_module_entry</font><font color="4444FF">;</font> 25 <font
color="0000ff">#define phpext_aalib_ptr &aalib_module_entry</font> 26 27 <font
color="0000ff">#ifdef PHP_WIN32</font> 28 <font color="0000ff">#define PHP_AALIB_API
__declspec(dllexport)</font> 29 <font color="0000ff">#else</font> 30 <font
color="0000ff">#define PHP_AALIB_API</font> 31 <font color="0000ff">#endif</font> 32 33
<font color="0000ff">#ifdef ZTS</font> 34 <font color="0000ff">#include <font
color="#008000">"&quot;TSRM.h&quot;</font></font> 35 <font color="0000ff">#endif</font>
36 37 <font color="#2040a0">PHP_MINIT_FUNCTION</font><font
color="4444FF"></font><font color="#2040a0">aalib</font><font
color="4444FF"></font><font color="4444FF">;</font> 38 <font
color="#2040a0">PHP_MSHUTDOWN_FUNCTION</font><font
color="4444FF"></font><font color="#2040a0">aalib</font><font
color="4444FF"></font><font color="4444FF">;</font> 39 <font
color="#2040a0">PHP_RINIT_FUNCTION</font><font color="4444FF"></font><font
color="#2040a0">aalib</font><font color="4444FF"></font><font color="4444FF">;</font> 40
<font color="#2040a0">PHP_RSHUTDOWN_FUNCTION</font><font
color="4444FF"></font><font color="#2040a0">aalib</font><font
color="4444FF"></font><font color="4444FF">;</font> 41 <font
color="#2040a0">PHP_MINFO_FUNCTION</font><font color="4444FF"></font><font
color="#2040a0">aalib</font><font color="4444FF"></font><font color="4444FF">;</font> 42
43 <font color="#2040a0">PHP_FUNCTION</font><font color="4444FF"></font><font
color="#2040a0">confirm_aalib_compiled</font><font color="4444FF"></font><font
color="4444FF">;</font> <font color="#444444"/> / For testing, remove later. /</font> 44 <font
color="#2040a0">PHP_FUNCTION</font><font color="4444FF"></font><font
color="#2040a0">aa_init</font><font color="4444FF"></font><font color="4444FF">;</font>
45 <font color="#2040a0">PHP_FUNCTION</font><font color="4444FF"></font><font
color="#2040a0">aa_get_screen_height</font><font color="4444FF"></font><font
color="4444FF">;</font> 46 <font color="#2040a0">PHP_FUNCTION</font><font
color="4444FF"></font><font color="#2040a0">aa_get_screen_width</font><font
color="4444FF"></font><font color="4444FF">;</font> 47 <font
color="#2040a0">PHP_FUNCTION</font><font color="4444FF"></font><font
color="#2040a0">aa_put_pixel</font><font color="4444FF"></font><font
color="4444FF"></font> 48 <font color="#2040a0">PHP_FUNCTION</font><font
color="4444FF"></font><font color="#2040a0">aa_render</font><font
color="4444FF"></font><font color="4444FF">;</font> 49 <font
color="#2040a0">PHP_FUNCTION</font><font color="4444FF"></font><font
color="#2040a0">aa_get_all</font><font color="4444FF"></font><font
color="4444FF">;</font> 50 <font color="#2040a0">PHP_FUNCTION</font><font
color="4444FF"></font><font color="#2040a0">aa_get_lines</font><font
color="4444FF"></font><font color="4444FF">;</font> 51 <font
color="#2040a0">PHP_FUNCTION</font><font color="4444FF"></font><font
color="#2040a0">aa_get_array</font><font color="4444FF"></font><font
color="4444FF">;</font> 52 53 <font color="#444444"/>*/ 54 Declare any global variables
you may need between the BEGIN 55 and END macros here: 56 57
ZEND_BEGIN_MODULE_GLOBALS(aalib) 58 int global_value; 59 char
*global_string; 60 ZEND_END_MODULE_GLOBALS(aalib) 61 */</font> 62 63 <font
color="#444444"/>*/ In every utility function you add that needs to use variables 64 in
php_aalib_globals, call TSRM_FETCH(); after declaring other 65 variables used by that
function, or better yet, pass in TSRMLS_CC 66 after the last function argument and declare

```


aalib.c

```

<html> <head> <title>aalib.c</title> </head> <body bgcolor="#ffffff" text="#000000"> <pre>
1 <font color="#444444"> /* 2 +-----+ 3
  | PHP Version 4 | 4
+-----+ 5 | Copyright (c) 1997-2002 The
PHP Group | 6 +-----+
7 | This source file is subject to version 2.02 of the PHP license, | 8 | that is bundled with
this package in the file LICENSE, and is | 9 | available at through the world-wide-web at
| 10 | http://www.php.net/license/2_02.txt. | 11 | If you did not
receive a copy of the PHP license and are unable to | 12 | obtain it through the world-wide-web,
please send a note to | 13 | license@php.net so we can mail you a copy immediately.
| 14 +-----+ 15 | Author:
| 16 +-----+ 17 18
$Id: protos-result.xml,v 1.3 2002/12/09 09:48:03 derick Exp $ 19 */</font> 20 21 <font
color="0000ff">#ifdef HAVE_CONFIG_H</font> 22 <font color="0000ff">#include <font
color="#008000">"config.h"</font></font> 23 <font color="0000ff">#endif</font>
24 25 <font color="0000ff">#include <font
color="#008000">"php.h"</font></font> 26 <font color="0000ff">#include <font
color="#008000">"php_ini.h"</font></font> 27 <font color="0000ff">#include
<font color="#008000">"ext/standard/info.h"</font></font> 28 <font
color="0000ff">#include <font color="#008000">"php_aalib.h"</font></font> 29
30 <font color="#444444"> /* If you declare any globals in php_aalib.h uncomment this: 31
ZEND_DECLARE_MODULE_GLOBALS(aalib) 32 */</font> 33 34 <font color="#444444"> /
True global resources - no need for thread safety here </font> 35 static int <font
color="#2040a0">le_aalib</font><font color="4444FF">;</font> 36 37 <font
color="#444444"> /* {{{ aalib_functions[] 38 * 39 * Every user visible function must have an
entry in aalib_functions[]. 40 */</font> 41 <font color="#2040a0">function_entry</font> <font
color="#2040a0">aalib_functions</font><font color="4444FF">[</font><font
color="4444FF">]</font> <font color="4444FF">=</font> <font color="4444FF">{</font> 42
<font color="#2040a0">PHP_FE</font><font color="4444FF">(</font><font
color="#2040a0">confirm_aalib_compiled</font>, <font color="#2040a0">NULL</font><font
color="4444FF">)</font> <font color="#444444"> / For testing, remove later. </font> 43
<font color="#2040a0">PHP_FE</font><font color="4444FF">(</font><font
color="#2040a0">aa_init</font>,<font color="4444FF">(</font><font
color="#2040a0">NULL</font><font color="4444FF">)</font> 44 <font color="#2040a0">PHP_FE</font><font
color="4444FF">(</font><font color="#2040a0">aa_get_screen_height</font>,<font
color="4444FF">(</font><font color="#2040a0">NULL</font><font color="4444FF">)</font> 45 <font
color="#2040a0">PHP_FE</font><font color="4444FF">(</font><font
color="#2040a0">aa_get_screen_width</font>,<font color="4444FF">(</font><font
color="#2040a0">NULL</font><font color="4444FF">)</font> 46 <font color="#2040a0">PHP_FE</font><font
color="4444FF">(</font><font color="#2040a0">aa_put_pixel</font>,<font
color="4444FF">(</font><font color="#2040a0">NULL</font><font color="4444FF">)</font> 47 <font
color="#2040a0">PHP_FE</font><font color="4444FF">(</font><font
color="#2040a0">aa_render</font>,<font color="4444FF">(</font><font
color="#2040a0">NULL</font><font color="4444FF">)</font> 48 <font color="#2040a0">PHP_FE</font><font
color="4444FF">(</font><font color="#2040a0">aa_get_all</font>,<font
color="4444FF">(</font><font color="#2040a0">NULL</font><font color="4444FF">)</font> 49 <font
color="#2040a0">PHP_FE</font><font color="4444FF">(</font><font
color="#2040a0">aa_get_lines</font>,<font color="4444FF">(</font><font
color="#2040a0">NULL</font><font color="4444FF">)</font> 50 <font color="#2040a0">PHP_FE</font><font
color="4444FF">(</font><font color="#2040a0">aa_get_array</font>,<font
color="4444FF">(</font><font color="#2040a0">NULL</font><font color="4444FF">)</font> 51 <font
color="4444FF">{</font><font color="#2040a0">NULL</font>,<font
color="4444FF">(</font><font color="#2040a0">NULL</font>,<font
color="4444FF">(</font><font color="#2040a0">NULL</font>,<font
color="4444FF">}</font> <font color="#444444"> / Must be the last line in aalib_functions[]
</font> 52 <font color="4444FF">}</font><font color="4444FF">;</font> 53 <font
color="#444444"> / }}} </font> 54 55 <font color="#444444"> /* {{{ aalib_module_entry 56
*/</font> 57 <font color="#2040a0">zend_module_entry</font> <font
color="#2040a0">aalib_module_entry</font> <font color="4444FF">=</font> <font
color="4444FF">{</font> 58 <font color="0000ff">#if ZEND_MODULE_API_NO >=
20010901</font> 59 <font color="#2040a0">STANDARD_MODULE_HEADER</font>,<font
color="0000ff">#endif</font> 60 <font color="0000ff">#endif</font> 61 <font
color="#008000">"aalib"</font>,<font color="4444FF">(</font> 62 <font
color="#2040a0">aalib_functions</font>,<font color="4444FF">(</font> 63 <font
color="#2040a0">PHP_MINIT</font><font color="4444FF">(</font><font
color="#2040a0">aalib</font><font color="4444FF">)</font>,<font
color="4444FF">(</font> 64 <font

```

config.m4

```

<html> <head> <title>config.m4</title> </head> <body bgcolor="#ffffff" text="#000000">
<pre> <a name="line4"> 4</a> PHP_ARG_WITH(aalib, for aalib support, <a name="line5">
5</a> [ --with-aalib Include aalib support]) <a name="line6"> 6</a> <a name="line7">
7</a> if test &quot;${PHP_AALIB}&quot; != &quot;no&quot;; then <a name="line8"> 8</a> <a
name="line9"> 9</a> SEARCH_PATH=&quot;/usr/local /usr&quot; <font
color="#444444"># you might want to change this</font> <a name="line10">10</a>
SEARCH_FOR=&quot;include</strong>/aalib.h&quot; <font color="#444444"># you
most likely want to change this</font> <a name="line11">11</a> if test -r
$PHP_AALIB/$SEARCH_FOR; then <font color="#444444"># path given as parameter</font>
<a name="line12">12</a> AALIB_DIR=$PHP_AALIB <a name="line13">13</a> else <font
color="#444444"># search default path list</font> <a name="line14">14</a>
AC_MSG_CHECKING([for aalib files in default path]) <a name="line15">15</a> for i in
$SEARCH_PATH ; do <a name="line16">16</a> if test -r $i/$SEARCH_FOR; then <a
name="line17">17</a> AALIB_DIR=$i <a name="line18">18</a>
AC_MSG_RESULT(found in $i) <a name="line19">19</a> fi <a name="line20">20</a>
done <a name="line21">21</a> fi <a name="line22">22</a> <a name="line23">23</a> if test
-z &quot;${AALIB_DIR}&quot;; then <a name="line24">24</a> AC_MSG_RESULT([not
found]) <a name="line25">25</a> AC_MSG_ERROR([Please reinstall the aalib distribution])
<a name="line26">26</a> fi </pre> </body> </html>

```

Check if the library works

config.m4

```
<pre>          <a          name="line28">28</a>
PHP_ADD_INCLUDE($AALIB_DIR/<strong>include</strong>) <a name="line29">29</a> <a
name="line30">30</a> LIBNAME=aalib <font color="#444444"># you may want to change
this</font> <a name="line31">31</a> LIBSYMBOL=aa_init <font color="#444444"># you
most likely want to change this </font> <a name="line32">32</a> <a name="line33">33</a>
PHP_CHECK_LIBRARY($LIBNAME,$LIBSYMBOL, <a name="line34">34</a> [ <a
name="line35">35</a> PHP_ADD_LIBRARY_WITH_PATH($LIBNAME, $AALIB_DIR/lib,
AALIB_SHARED_LIBADD) <a name="line36">36</a> AC_DEFINE(HAVE_AALIBLIB,1,[
]) <a name="line37">37</a> ],[ <a name="line38">38</a> AC_MSG_ERROR([wrong aalib
lib version or lib not found]) <a name="line39">39</a> ],[ <a name="line40">40</a>
-L$AALIB_DIR/lib -lm -ldl <a name="line41">41</a> ]) <a name="line42">42</a> <a
name="line43">43</a> PHP_SUBST(AALIB_SHARED_LIBADD) <a name="line44">44</a>
<a name="line45">45</a> PHP_NEW_EXTENSION(aalib, aalib.c, $ext_shared) <a
name="line46">46</a> fi </pre>
```

- o Copy our newly created extension to some other place:

```
# mv ext/aalib /dat/dev/php
```

- o Go to the new directory and run phpize:

```
# cd /dat/dev/php/aalib  
# phpize
```

- o You should see the following output (or similar):

```
You should update your `aclocal.m4' by running aclocal.  
Configuring for:  
PHP Api Version:    20020918  
Zend Module Api No: 20020429  
Zend Extension Api No: 20021010
```

Run configure to test for the library and create the Makefile:

```
# ./configure
checking how to run the C preprocessor... gcc -E
checking for aalib support... yes, shared
checking for aalib files in default path... found in /usr/local
checking for aalib in -laalib... no
configure: error: wrong aalib lib version or lib not found
```

We messed up :(

config.log to the rescue:

```
<pre> configure:1659: checking for aa_init in -laalib configure:1678: gcc -o conftest -g -O2
-L/usr/local/lib -lm -ldl conftest.c -laalib 1>&5 <font color='red'>/usr/bin/ld: cannot find
-laalib</font> collect2: ld returned 1 exit status configure: failed program was: #line 1667
"configure" #include "confdefs.h" /* Override any gcc2 internal prototype to avoid an error. We
use char because int might match the return type of a gcc2 builtin and then its argument
prototype would still apply. */ char aa_init(); int main() { aa_init() ; return 0; } </pre>
```

The library is called 'aa' (and not 'aalib'), so we fix config.m4:

```
<pre> <a name="line29">29</a> <a name="line30">30</a> LIBNAME=aa <font
color="#444444"># you may want to change this</font> <a name="line31">31</a>
LIBSYMBOL=aa_init <font color="#444444"># you most likely want to change this </font> <a
name="line32">32</a> </pre>
```

Run configure to test for the library and create the Makefile:

```
# ./configure
creating cache ./config.cache
checking for gcc... gcc
checking whether the C compiler (gcc ) works... yes
...
creating ./config.status
creating config.h
```

Now run make:

```
# make
```

And of course we get lots of compile errors :-)

- o Includes
- o Function table
- o Module entry
- o INI settings
- o Module/Request start and shutdown handlers
- o phpinfo() function
- o Function implementations

aalib.c:

```

<pre> <a name="line117">117</a> <font color="#444444"> /* {{{ proto resource aa_init() <a
name="line118">118</a> Initialize the AA library */</font> <a name="line119">119</a> <font
color="#2040a0">PHP_FUNCTION</font><font color="4444FF"></font><font
color="#2040a0">aa_init</font><font color="4444FF"></font> <a name="line120">120</a>
<font color="4444FF"><strong>{</strong></font> <a name="line121">121</a> <font
color="#ff0000">aa_context *c;</font> <a name="line122">122</a> <a
name="line123">123</a> <strong>if</strong> <font color="4444FF"></font><font
color="#2040a0">ZEND_NUM_ARGS</font><font color="4444FF"></font><font
color="4444FF"></font> <font color="4444FF">!</font><font color="4444FF">=</font> <font
color="#FF0000">0</font><font color="4444FF"></font> <font
color="4444FF"><strong>{</strong></font> <a name="line124">124</a> <font
color="#2040a0">WRONG_PARAM_COUNT</font><font color="4444FF"></font> <a
name="line125">125</a> <font color="4444FF"><strong>}</strong></font> <a
name="line126">126</a> <a name="line127">127</a> <font color="#ff0000">c =
aa_init(&amp;mem_d, &amp;aa_defparams, NULL);</font> <a name="line128">128</a> <font
color="#ff0000">ZEND_REGISTER_RESOURCE(return_value, c, le_aalib);</font> <a
name="line129">129</a> <font color="4444FF"><strong>}</strong></font> <a
name="line130">130</a> <font color="#444444">/ } } /</font> </pre>

```

aalib.c:

```

<pre> <a name="line77"> 77</a> <strong>static</strong> <strong>void</strong> <font
color="#2040a0">rsrc_close_aalib</font><font color="4444FF">(</font><font
color="#2040a0">zend_rsrc_list_entry</font> <font color="4444FF">*</font><font
color="#2040a0">rsrc</font> <font color="#2040a0">TSRMLS_DC</font><font
color="4444FF">)</font> <a name="line78"> 78</a> <font
color="4444FF"><strong>{</strong></font> <a name="line79"> 79</a> <font
color="#2040a0">aa_context</font> <font color="4444FF"></font><font
color="#2040a0">c</font> <font color="4444FF">=</font> <font color="4444FF">(</font><font
color="#2040a0">aa_context</font> <font color="4444FF"></font><font
color="4444FF">)</font> <font color="#2040a0">rsrc</font><font
color="4444FF">-</font><font color="4444FF">&gt;</font><font
color="#2040a0">ptr</font><font color="4444FF">;</font> <a name="line80"> 80</a> <font
color="#2040a0">aa_close</font><font color="4444FF">(</font><font
color="#2040a0">c</font><font color="4444FF">)</font><font color="4444FF">;</font><font
name="line81"> 81</a> <font color="4444FF"><strong>}</strong></font> <a name="line82">
82</a> <a name="line83"> 83</a> <a name="line84"> 84</a> <font
color="#2040a0">PHP_MINIT_FUNCTION</font><font color="4444FF">(</font><font
color="#2040a0">aalib</font><font color="4444FF">)</font> <a name="line85"> 85</a> <font
color="4444FF"><strong>{</strong></font> <a name="line86"> 86</a> <font
color="#2040a0">le_aalib</font> <font color="4444FF">=</font> <font
color="#2040a0">zend_register_list_destructors_ex</font><font color="4444FF">(</font><font
color="#2040a0">rsrc_close_aalib</font>, <font color="#2040a0">NULL</font>, <font
color="#008000">&quot;aa&quot;</font>, <font color="#2040a0">module_number</font><font
color="4444FF">)</font><font color="4444FF">;</font> <a name="line87"> 87</a> <font
color="4444FF"><strong>}</strong></font> </pre>

```

aalib.c:

```

<pre>  <a name="line132">132</a>  <font color="#444444">/*  {{{  proto  int
aa_get_screen_height(resource c) <a name="line133">133</a>  Get the screen height */</font>
<a name="line134">134</a>  <font color="#2040a0">PHP_FUNCTION</font><font
color="4444FF"></font><font color="#2040a0">aa_get_screen_height</font><font
color="4444FF"></font>  <a name="line135">135</a>  <font
color="4444FF"><strong>{</strong></font>  <a name="line136">136</a>  <strong>int</strong>
<font color="#2040a0">argc</font>  <font color="4444FF">=</font>  <font
color="4444FF"></font><font color="4444FF"></font>  <a name="line137">137</a>
<strong>int</strong> <font color="#2040a0">c_id</font> <font color="4444FF">=</font> <font
color="4444FF"></font><font color="#FF0000">1</font><font color="4444FF">;</font>  <a
name="line138">138</a>  <font color="#2040a0">zval</font>  <font
color="4444FF">*</font><font color="#2040a0">c</font>  <font color="4444FF">=</font>
<font color="#2040a0">NULL</font><font color="4444FF">;</font>  <a
name="line139">139</a>  <font color="#ff0000">aa_context  *ac;</font>  <a
name="line140">140</a>  <a name="line141">141</a>  <strong>if</strong>  <font
color="4444FF"></font><font color="#2040a0">zend_parse_parameters</font><font
color="4444FF"></font><font color="#2040a0">argc</font>  <font
color="#2040a0">TSRMLS_CC</font>, <font color="#008000">&quot;r&quot;</font>, <font
color="4444FF">&amp;</font><font color="#2040a0">c</font><font color="4444FF"></font>
<font color="4444FF">=</font><font color="4444FF">=</font>  <font
color="#2040a0">FAILURE</font><font color="4444FF"></font>  <a
name="line142">142</a>  <strong>return</strong><font color="4444FF">;</font>  <a
name="line143">143</a>  <a name="line144">144</a>  <strong>if</strong>  <font
color="4444FF"></font><font color="#2040a0">c</font><font color="4444FF"></font>  <font
color="4444FF"><strong>{</strong></font>  <a name="line145">145</a>  <font
color="#ff0000">ZEND_FETCH_RESOURCE(ac, aa_context *, &amp;c, c_id, &quot;aa&quot;,
le_aalib);</font>  <a name="line146">146</a>  <font
color="4444FF"><strong>}</strong></font>  <a name="line147">147</a>  <font
color="4444FF"><strong>}</strong></font>  <a name="line148">148</a>  <font
color="#444444">/>  }}}  </font>

```

Run make:

```
# make
...
Libraries have been installed in:
  /dat/dev/php/aalib/modules
```

Test library:

```
# php -r 'dl("/dat/dev/php/aalib/modules/aalib.so");
          echo "\n".confirm_aalib_compiled("aalib")."\n";'
```

Output:

```
Congratulations! You have successfully modified ext/aalib/config.m4. Module
aalib is now compiled into PHP.
```

aalib.c (aa_get_screen_height()):

```
<pre> <a name="line144">144</a> <strong>if</strong> <font color="4444FF"></font><font
color="#2040a0">c</font><font color="4444FF"></font> <font
color="4444FF"><strong>{</strong></font> <a name="line145">145</a> <font
color="#2040a0">ZEND_FETCH_RESOURCE</font><font color="4444FF"></font><font
color="#2040a0">ac</font>, <font color="#2040a0">aa_context</font> <font
color="4444FF">*</font>, <font color="4444FF">&amp;</font><font
color="#2040a0">c</font>, <font color="#2040a0">c_id</font>, <font
color="#008000">&quot;aa&quot;</font>, <font color="#2040a0">le_aalib</font><font
color="4444FF"></font><font color="4444FF">;</font> <a name="line146">146</a> <font
color="4444FF"><strong>}</strong></font> <a name="line147">147</a> <a
name="line148">148</a> <font color="#ff0000">RETURN_LONG(aa_scrheight(ac));</font>
<a name="line149">149</a> <font color="4444FF"><strong>}</strong></font> </pre>
```

aalib.c (aa_get_screen_width()):

```
<pre> <a name="line154">154</a> <font color="#2040a0">PHP_FUNCTION</font><font
color="4444FF"></font><font color="#2040a0">aa_get_screen_width</font><font
color="4444FF"></font> <a name="line155">155</a> <font
color="4444FF"><strong>{</strong></font> <a name="line156">156</a> <strong>int</strong>
<font color="#2040a0">argc</font> <font color="4444FF">=</font> <font
color="#2040a0">ZEND_NUM_ARGS</font><font color="4444FF"></font><font
color="4444FF"></font><font color="4444FF">;</font> <a name="line157">157</a>
<strong>int</strong> <font color="#2040a0">c_id</font> <font color="4444FF">=</font> <font
color="4444FF">-</font><font color="#FF0000">1</font><font color="4444FF">;</font> <a
name="line158">158</a> <font color="#2040a0">zval</font> <font
color="4444FF">*</font><font color="#2040a0">c</font> <font color="4444FF">=</font>
<font color="#2040a0">NULL</font><font color="4444FF">;</font> <a
name="line159">159</a> <font color="#2040a0">aa_context</font> <font
color="4444FF">*</font><font color="#2040a0">ac</font><font color="4444FF">;</font> <a
name="line160">160</a> <a name="line161">161</a> <strong>if</strong> <font
color="4444FF"></font><font color="#2040a0">zend_parse_parameters</font><font
color="4444FF"></font><font color="#2040a0">argc</font> <font
color="#2040a0">TSRMLS_CC</font>, <font color="#008000">&quot;r&quot;</font>, <font
color="4444FF">&amp;</font><font color="#2040a0">c</font><font color="4444FF"></font>
<font color="4444FF">=</font><font color="4444FF">=</font> <font
color="#2040a0">FAILURE</font><font color="4444FF"></font> <a
name="line162">162</a> <strong>return</strong><font color="4444FF">;</font> <a
name="line163">163</a> <a name="line164">164</a> <strong>if</strong> <font
color="4444FF"></font><font color="#2040a0">c</font><font color="4444FF"></font> <font
color="4444FF"><strong>{</strong></font> <a name="line165">165</a> <font
color="#2040a0">ZEND_FETCH_RESOURCE</font><font color="4444FF"></font><font
color="#2040a0">ac</font>, <font color="#2040a0">aa_context</font> <font
color="4444FF">*</font>, <font color="4444FF">&amp;</font><font
color="#2040a0">c</font>, <font color="#2040a0">c_id</font>, <font
color="#008000">&quot;aa&quot;</font>, <font color="#2040a0">le_aalib</font><font
color="4444FF"></font><font color="4444FF">;</font> <a name="line166">166</a> <font
color="4444FF"><strong>}</strong></font> <a name="line167">167</a> <a
name="line168">168</a> <font color="#ff0000">RETURN_LONG(aa_scrwidth(ac));</font> <a
name="line169">169</a> <font color="4444FF"><strong>}</strong></font> </pre>
```



```
Optional long and string:<pre> <strong>long</strong> <font color="#2040a0">opt_l</font>
<font color="4444FF">=</font> <font color="#FF0000">0</font><font
color="4444FF">;</font> <strong>char</strong> <font color="4444FF">*</font><font
color="#2040a0">opt_str</font> <font color="4444FF">=</font> <font
color="#2040a0">NULL</font><font color="4444FF">;</font> <strong>long</strong> <font
color="#2040a0">opt_str_len</font> <font color="4444FF">=</font> <font
color="#FF0000">0</font><font color="4444FF">;</font> <strong>if</strong> <font
color="4444FF">(</font><font color="#2040a0">zend_parse_parameters</font><font
color="4444FF">(</font><font color="#2040a0">ZEND_NUM_ARGS</font><font
color="4444FF">(</font><font color="4444FF">)</font> <font
color="#2040a0">TSRMLS_CC</font>,< font color="#008000">&quot;|s&quot;</font>,<
<font color="4444FF">&amp;</font><font color="#2040a0">opt_l</font>,< font
color="4444FF">&amp;</font><font color="#2040a0">opt_str</font>,< font
color="4444FF">&amp;</font><font color="#2040a0">opt_str_len</font><font
color="4444FF">)</font> <font color="4444FF">=</font><font color="4444FF">=</font> <font
color="#2040a0">FAILURE</font><font color="4444FF">)</font> <font
color="4444FF"><strong>{</strong></font> </pre>
```

```
Get the first two parameters as resource and optional boolean:<pre> <font
color="#2040a0">zval</font> <font color="4444FF">*</font><font
color="#2040a0">resource</font><font color="4444FF">;</font> <font
color="#2040a0">zend_bool</font> <font color="#2040a0">b</font> <font
color="4444FF">=</font> <font color="#2040a0">FALSE</font><font
color="4444FF">;</font> <strong>if</strong> <font color="4444FF">(</font><font
color="#2040a0">zend_parse_parameters</font><font color="4444FF">(</font><font
color="#FF0000">2</font> <font color="#2040a0">TSRMLS_CC</font>,< font
color="#008000">&quot;r|b&quot;</font>,< font color="4444FF">&amp;</font><font
color="#2040a0">resource</font>,< font color="4444FF">&amp;</font><font
color="#2040a0">b</font><font color="4444FF">)</font> <font color="4444FF">=</font><font
color="4444FF">=</font> <font color="#2040a0">FAILURE</font><font
color="4444FF">)</font> <font color="4444FF"><strong>{</strong></font> </pre>
```

```
Get a long or a string:<pre> <strong>long</strong> <font color="#2040a0">number</font><font
color="4444FF">;</font> <strong>char</strong> <font color="4444FF">*</font><font
color="#2040a0">str</font><font color="4444FF">;</font> <strong>int</strong> <font
color="#2040a0">str_len</font><font color="4444FF">;</font> <strong>if</strong> <font
color="4444FF">(</font><font color="#2040a0">zend_parse_parameters_ex</font><font
color="4444FF">(</font><font color="#2040a0">ZEND_PARSE_PARAMS_QUIET</font>,<
<font color="#2040a0">ZEND_NUM_ARGS</font><font color="4444FF">(</font><font
color="4444FF">)</font> <font color="#2040a0">TSRMLS_CC</font>,< font
color="#008000">&quot;l&quot;</font>,< font color="4444FF">&amp;</font><font
color="#2040a0">number</font><font color="4444FF">)</font> <font
color="4444FF">=</font><font color="4444FF">=</font> <font
color="#2040a0">SUCCESS</font><font color="4444FF">)</font> <font
color="4444FF"><strong>{</strong></font> <font color="#444444">}/ Do something with the
number /</font> <font color="4444FF"><strong>}</strong></font> <strong>else</strong>
<strong>if</strong> <font color="4444FF">(</font><font
color="#2040a0">zend_parse_parameters_ex</font><font color="4444FF">(</font><font
color="#2040a0">ZEND_PARSE_PARAMS_QUIET</font>,< font
color="#2040a0">ZEND_NUM_ARGS</font><font color="4444FF">(</font><font
color="4444FF">)</font> <font color="#2040a0">TSRMLS_CC</font>,< font
color="#008000">&quot;s&quot;</font>,< font color="4444FF">&amp;</font><font
color="#2040a0">str</font>,< font color="4444FF">&amp;</font><font
color="#2040a0">str_len</font><font color="4444FF">)</font> <font
color="4444FF">=</font><font color="4444FF">=</font> <font
color="#2040a0">SUCCESS</font><font color="4444FF">)</font> <font
color="4444FF"><strong>{</strong></font> <font color="#444444">}/ Do something with the
string /</font> <font color="4444FF"><strong>}</strong></font> <strong>else</strong> <font
color="4444FF"><strong>{</strong></font> <font color="#444444">}/ Output error /</font>
<font color="4444FF"><strong>}</strong></font> </pre>
```

aalib.c (aa_put_pixel()):

```
<pre> <a name="line187">187</a> <strong>if</strong> <font color="4444FF"></font><font
color="#2040a0">c</font><font color="4444FF"></font> <font
color="4444FF"><strong>{</strong></font> <a name="line188">188</a> <font
color="#2040a0">ZEND_FETCH_RESOURCE</font><font color="4444FF"></font><font
color="#2040a0">ac</font>, <font color="#2040a0">aa_context</font> <font
color="4444FF">*</font>, <font color="4444FF">&amp;</font><font
color="#2040a0">c</font>, <font color="#2040a0">c_id</font>, <font
color="#008000">&quot;aa&quot;</font>, <font color="#2040a0">le_aalib</font><font
color="4444FF"></font><font color="4444FF">;</font> <a name="line189">189</a> <font
color="4444FF"><strong>}</strong></font> <a name="line190">190</a> <a
name="line191">191</a> <font color="#2040a0">aa_putpixel</font><font
color="4444FF"></font><font color="#2040a0">ac</font>, <font color="#2040a0">x</font>,
<font color="#2040a0">y</font>, <font color="#2040a0">value</font><font
color="4444FF"></font><font color="4444FF">;</font> <a name="line192">192</a> <font
color="4444FF"><strong>}</strong></font> </pre>
```

aalib.c (aa_render()):

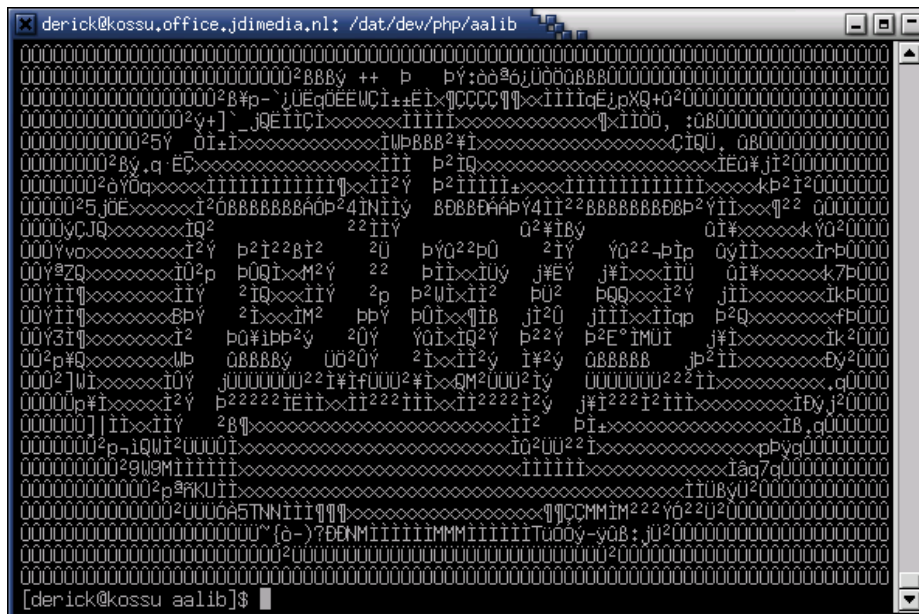
```
<pre> <a name="line197">197</a> <font color="#2040a0">PHP_FUNCTION</font><font
color="4444FF"></font><font color="#2040a0">aa_render</font><font
color="4444FF"></font> <a name="line198">198</a> <font
color="4444FF"><strong>{</strong></font> <a name="line199">199</a> <strong>int</strong>
<font color="#2040a0">argc</font> <font color="4444FF">=</font> <font
color="#2040a0">ZEND_NUM_ARGS</font><font color="4444FF"></font><font
color="4444FF"></font><font color="4444FF">;</font> <a name="line200">200</a>
<strong>int</strong> <font color="#2040a0">c_id</font> <font color="4444FF">=</font> <font
color="4444FF">-</font><font color="#FF0000">1</font><font color="4444FF">;</font> <a
name="line201">201</a> <strong>long</strong> <font color="#2040a0">contrast</font> <font
color="4444FF">=</font> <font color="#FF0000">0</font><font color="4444FF">;</font> <a
name="line202">202</a> <strong>long</strong> <font color="#2040a0">dither</font> <font
color="4444FF">=</font> <font color="#FF0000">0</font><font color="4444FF">;</font> <a
name="line203">203</a> <strong>long</strong> <font color="#2040a0">inversion</font> <font
color="4444FF">=</font> <font color="#FF0000">0</font><font color="4444FF">;</font> <a
name="line204">204</a> <strong>long</strong> <font color="#2040a0">randomval</font>
<font color="4444FF">=</font> <font color="#FF0000">0</font><font
color="4444FF">;</font> <a name="line205">205</a> <font color="#2040a0">zval</font>
<font color="4444FF">*</font><font color="#2040a0">c</font> <font
color="4444FF">=</font> <font color="#2040a0">NULL</font><font color="4444FF">;</font>
<a name="line206">206</a> <font color="#2040a0">aa_context</font> <font
color="4444FF">*</font><font color="#2040a0">ac</font><font color="4444FF">;</font> <a
name="line207">207</a> <font color="#2040a0">aa_renderparams</font> <font
color="4444FF">*</font><font color="#2040a0">params</font><font color="4444FF">;</font>
<a name="line208">208</a> <a name="line209">209</a> <strong>if</strong> <font
color="4444FF"></font><font color="#2040a0">zend_parse_parameters</font><font
color="4444FF"></font><font color="#2040a0">argc</font> <font
color="#2040a0">TSRMLS_CC</font>, <font color="#008000">&quot;rllll&quot;</font>,
<font color="4444FF">&amp;</font><font color="#2040a0">c</font>, <font
color="4444FF">&amp;</font><font color="#2040a0">contrast</font>, <font
color="4444FF">&amp;</font><font color="#2040a0">dither</font>, <font
color="4444FF">&amp;</font><font color="#2040a0">inversion</font>, <font
color="4444FF">&amp;</font><font color="#2040a0">randomval</font><font
color="4444FF"></font> <font color="4444FF">=</font><font color="4444FF">=</font> <font
color="#2040a0">FAILURE</font><font color="4444FF"></font> <a
name="line210">210</a> <strong>return</strong><font color="4444FF">;</font> <a
name="line211">211</a> <a name="line212">212</a> <strong>if</strong> <font
color="4444FF"></font><font color="#2040a0">c</font><font color="4444FF"></font> <font
color="4444FF"><strong>{</strong></font> <a name="line213">213</a> <font
color="#2040a0">ZEND_FETCH_RESOURCE</font><font color="4444FF"></font><font
color="#2040a0">ac</font>, <font color="#2040a0">aa_context</font> <font
color="4444FF">*</font> <font color="4444FF">&amp;</font><font
```


test1.php:

```
<?php
dl("/dat/dev/php/aalib/modules/aalib.so");

/ Open .gif file /
$img = imagecreatefromgif("/home/derick/php_logo_c.gif");
$x = imagesx($img);
$y = imagesy($img);

$a = aa_init();
for ($i = 0; $i < $x; $i++) {
    for ($j = 0; $j < $y; $j++) {
        aa_put_pixel($a, $i, $j / 2, imagecolorat($img, $i, $j));
    }
}
aa_render($a, 40, 0, 0, -10);
echo aa_get_all($a);
? >
```



Returning from a function

```

<pre> <a name="line1"> 1</a> <font color="#2040a0">RETURN_NULL</font><font
color="4444FF"></font><font color="4444FF"></font> <a name="line3"> 3</a> <font
color="#2040a0">zend_bool</font> <font color="#2040a0">b</font><font
color="4444FF">;</font> <a name="line4"> 4</a> <font
color="#2040a0">RETURN_BOOL</font><font color="4444FF"></font><font
color="#2040a0">b</font><font color="4444FF"></font> <a name="line5"> 5</a> <font
color="#2040a0">RETURN_FALSE</font> <a name="line6"> 6</a> <font
color="#2040a0">RETURN_TRUE</font> <a name="line8"> 8</a> <strong>long</strong>
<font color="#2040a0">l</font><font color="4444FF">;</font> <a name="line9"> 9</a> <font
color="#2040a0">RETURN_LONG</font><font color="4444FF"></font><font
color="#2040a0">l</font><font color="4444FF"></font> <a name="line11">11</a>
<strong>double</strong> <font color="#2040a0">d</font><font color="4444FF">;</font> <a
name="line12">12</a> <font color="#2040a0">RETURN_DOUBLE</font><font
color="4444FF"></font><font color="#2040a0">d</font><font color="4444FF"></font> <a
name="line14">14</a> <strong>char</strong> <font color="4444FF">*</font><font
color="#2040a0">s</font><font color="4444FF">;</font> <a name="line15">15</a>
<strong>int</strong> <font color="#2040a0">l</font><font color="4444FF">;</font> <a
name="line16">16</a> <strong>int</strong> <font color="#2040a0">duplicate</font><font
color="4444FF">;</font> <a name="line17">17</a> <font
color="#2040a0">RETURN_STRING</font><font color="4444FF"></font><font
color="#2040a0">s</font>, <font color="#2040a0">duplicate</font><font
color="4444FF"></font> <a name="line18">18</a> <font
color="#2040a0">RETURN_STRINGL</font><font color="4444FF"></font><font
color="#2040a0">s</font>, <font color="#2040a0">l</font>, <font
color="#2040a0">duplicate</font><font color="4444FF"></font> <a name="line19">19</a>
<font color="4444FF"></font><font color="4444FF"></font> <a name="line21">21</a> <font
color="#2040a0">zval</font> <font color="4444FF">*</font><font
color="#2040a0">r</font><font color="4444FF">;</font> <a name="line22">22</a> <font
color="#2040a0">RETURN_RESOURCE</font><font color="4444FF"></font><font
color="#2040a0">r</font><font color="4444FF"></font> </pre>

```

Returning an array

```

<pre> <a name="line253">253</a> <font color="#2040a0">PHP_FUNCTION</font><font
color="4444FF">(</font><font
color="4444FF">)</font> <a
name="line254">254</a> <font
color="4444FF"><strong>{</strong></font> <a name="line255">255</a> <strong>int</strong>
<font
color="#2040a0">argc</font> <font
color="4444FF">=</font> <font
color="#2040a0">ZEND_NUM_ARGS</font><font
color="4444FF">)</font><font
color="4444FF">;</font> <a name="line256">256</a>
<strong>int</strong> <font color="#2040a0">c_id</font> <font color="4444FF">=</font> <font
color="4444FF">-</font><font color="#FF0000">1</font><font color="4444FF">;</font> <a
name="line257">257</a> <font
color="#2040a0">zval</font> <font
color="4444FF">*</font><font color="#2040a0">c</font> <font color="4444FF">=</font>
<font
color="#2040a0">NULL</font><font
color="4444FF">;</font> <a
name="line258">258</a> <font
color="#2040a0">aa_context</font> <font
color="4444FF">*</font><font color="#2040a0">ac</font><font color="4444FF">;</font> <a
name="line259">259</a> <strong>char</strong> <font color="4444FF">*</font><font
color="#2040a0">buffer</font><font color="4444FF">;</font> <a name="line260">260</a>
<strong>int</strong> <font color="#2040a0">i</font><font color="4444FF">;</font> <a
name="line261">261</a> <a name="line262">262</a> <strong>if</strong> <font
color="4444FF">(</font><font
color="#2040a0">zend_parse_parameters</font><font
color="4444FF">)</font><font
color="#2040a0">)</font><font
color="4444FF">)</font> <font
color="#2040a0">TSRMLS_CC</font>, <font color="#008000">&quot;r&quot;</font>, <font
color="4444FF">&amp;</font><font color="#2040a0">c</font><font color="4444FF">)</font>
<font
color="4444FF">=</font><font
color="4444FF">=</font> <font
color="#2040a0">FAILURE</font><font
color="4444FF">)</font> <a
name="line263">263</a> <strong>return</strong><font color="4444FF">;</font> <a
name="line264">264</a> <a name="line265">265</a> <strong>if</strong> <font
color="4444FF">(</font><font color="#2040a0">c</font><font color="4444FF">)</font> <font
color="4444FF"><strong>{</strong></font> <a name="line266">266</a> <font
color="#2040a0">ZEND_FETCH_RESOURCE</font><font color="4444FF">(</font><font
color="#2040a0">ac</font>, <font
color="#2040a0">aa_context</font> <font
color="4444FF">*</font>, <font
color="4444FF">&amp;</font><font
color="#2040a0">c</font>, <font
color="#2040a0">c_id</font>, <font
color="#008000">&quot;aa&quot;</font>, <font color="#2040a0">le_aalib</font><font
color="4444FF">)</font><font color="4444FF">;</font> <a name="line267">267</a> <font
color="4444FF"><strong>}</strong></font> <a name="line268">268</a> <a
name="line269">269</a> <strong>if</strong> <font color="4444FF">(</font><font
color="#2040a0">array_init</font><font
color="4444FF">)</font><font
color="#2040a0">return_value</font><font
color="4444FF">=</font><font
color="4444FF">=</font> <font
color="#2040a0">FAILURE</font><font
color="4444FF">)</font> <font
color="4444FF"><strong>{</strong></font> <a name="line270">270</a> <font
color="#2040a0">RETURN_FALSE</font><font
color="4444FF">;</font> <a
name="line271">271</a> <font color="4444FF"><strong>}</strong></font> <a
name="line272">272</a> <a name="line273">273</a> <font color="#2040a0">buffer</font>
<font
color="4444FF">=</font> <font
color="#2040a0">ecalloc</font><font
color="4444FF">(</font><font
color="4444FF">(</font><font color="#2040a0">aa_scrheight</font><font
color="4444FF">)</font><font color="#2040a0">ac</font><font color="4444FF">)</font> <font
color="4444FF">*</font> <font
color="#2040a0">aa_scrwidth</font><font
color="4444FF">(</font><font color="#2040a0">ac</font><font color="4444FF">)</font> <font
color="4444FF">+</font> <font
color="#FF0000">1</font>, <font
color="4444FF">1</font><font color="4444FF">)</font><font color="4444FF">;</font> <a
name="line274">274</a> <font
color="#2040a0">memcpy</font><font
color="4444FF">(</font><font
color="#2040a0">buffer</font>, <font
color="4444FF">)</font><font
color="#2040a0">aa_text</font><font
color="4444FF">)</font><font
color="#2040a0">ac</font><font
color="4444FF">)</font>, <font
color="#2040a0">aa_scrheight</font><font
color="4444FF">)</font><font
color="#2040a0">ac</font><font color="4444FF">)</font> <font color="4444FF">*</font>
<font
color="#2040a0">aa_scrwidth</font><font
color="4444FF">)</font><font
color="4444FF">)</font><font
color="4444FF">)</font><font
color="4444FF">;</font> <a name="line275">275</a> <a name="line276">276</a>
<strong>for</strong> <font color="4444FF">(</font><font color="#2040a0">i</font> <font
color="4444FF">=</font> <font color="#FF0000">0</font><font color="4444FF">:</font>

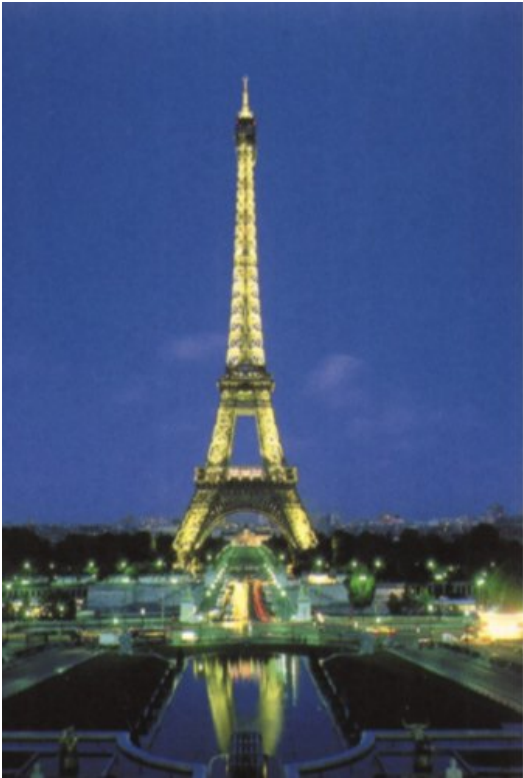
```

Returning a 2D array

```

<pre> <a name="line285">285</a> <font color="#2040a0">PHP_FUNCTION</font><font
color="4444FF"></font><font color="#2040a0">aa_get_array</font><font
color="4444FF"></font> <a name="line286">286</a> <font
color="4444FF"><strong>{</strong></font> <a name="line287">287</a>
<strong>int</strong> <font color="#2040a0">argc</font> <font color="4444FF">=</font> <font
color="#2040a0">ZEND_NUM_ARGS</font><font color="4444FF"></font><font
color="4444FF"></font><font color="4444FF">;</font> <a name="line288">288</a>
<strong>int</strong> <font color="#2040a0">c_id</font> <font color="4444FF">=</font> <font
color="4444FF"></font><font color="#FF0000">1</font><font color="4444FF">;</font> <a
name="line289">289</a> <font color="#2040a0">zval</font> <font
color="4444FF">*</font><font color="#2040a0">c</font> <font color="4444FF">=</font>
<font color="#2040a0">NULL</font><font color="4444FF">;</font> <a
name="line290">290</a> <font color="#2040a0">aa_context</font> <font
color="4444FF">*</font><font color="#2040a0">ac</font><font color="4444FF">;</font> <a
name="line291">291</a> <strong>char</strong> <font color="4444FF">*</font><font
color="#2040a0">buffer</font><font color="4444FF">;</font> <a name="line292">292</a>
<font color="#2040a0">zval</font> <font color="4444FF">*</font><font
color="#2040a0">line</font><font color="4444FF">;</font> <a name="line293">293</a>
<strong>int</strong> <font color="#2040a0">i</font>, <font color="#2040a0">j</font><font
color="4444FF">;</font> <a name="line302">302</a> <strong>if</strong> <font
color="4444FF"></font><font color="#2040a0">array_init</font><font
color="4444FF"></font><font color="#2040a0">return_value</font><font
color="4444FF">=</font><font color="4444FF">=</font> <font
color="#2040a0">FAILURE</font><font color="4444FF"></font> <font
color="4444FF"><strong>{</strong></font> <a name="line303">303</a> <font
color="#2040a0">RETURN_FALSE</font><font color="4444FF">;</font> <a
name="line304">304</a> <font color="4444FF"><strong>}</strong></font> <a
name="line305">305</a> <a name="line306">306</a> <font
color="#2040a0">buffer</font> <font color="4444FF">=</font> <font
color="#2040a0">ecalloc</font><font color="4444FF"></font><font
color="#2040a0">aa_scrheight</font><font color="4444FF"></font><font
color="#2040a0">ac</font><font color="4444FF"></font> <font color="4444FF">*</font>
<font color="#2040a0">aa_scrwidth</font><font color="4444FF"></font><font
color="4444FF"></font><font color="#2040a0">ac</font><font color="4444FF"></font>
<font color="4444FF">+</font>
<font color="#FF0000">1</font>, <font color="#FF0000">1</font><font
color="4444FF"></font><font color="4444FF">;</font> <a name="line307">307</a> <font
color="#2040a0">memcpy</font><font color="4444FF"></font><font
color="#2040a0">buffer</font>, <font color="#2040a0">aa_text</font><font
color="4444FF"></font><font color="#2040a0">ac</font><font color="4444FF"></font>,
<font color="#2040a0">aa_scrheight</font><font color="4444FF"></font><font
color="#2040a0">ac</font><font color="4444FF"></font> <font color="4444FF">*</font>
<font color="#2040a0">aa_scrwidth</font><font color="4444FF"></font><font
color="4444FF"></font><font color="#2040a0">ac</font><font color="4444FF"></font><font
color="4444FF"></font><font color="4444FF"></font><font
color="4444FF">;</font> <a name="line308">308</a> <a name="line309">309</a>
<strong>for</strong> <font color="4444FF"></font><font color="#2040a0">i</font> <font
color="4444FF">=</font> <font color="#FF0000">0</font><font color="4444FF">;</font>
<font color="#2040a0">i</font> <font color="4444FF">&lt;</font> <font
color="#2040a0">aa_scrheight</font><font color="4444FF"></font><font
color="#2040a0">ac</font><font color="4444FF"></font><font color="4444FF">;</font> <font
color="#2040a0">i</font><font color="4444FF">+</font><font color="4444FF">+</font><font
color="4444FF"></font> <font color="4444FF"><strong>{</strong></font> <a
name="line310">310</a> <font color="#2040a0">MAKE_STD_ZVAL</font><font
color="4444FF"></font><font color="#2040a0">line</font><font
color="4444FF"></font><font color="4444FF">;</font> <a name="line311">311</a>
<strong>if</strong> <font color="4444FF"></font><font
color="#2040a0">array_init</font><font color="4444FF"></font><font
color="#2040a0">line</font><font color="4444FF"></font>
<font color="4444FF">=</font><font color="4444FF">=</font>
<font color="#2040a0">FAILURE</font><font color="4444FF"></font>
<font color="4444FF"><strong>{</strong></font> <a name="line312">312</a>
<font color="#2040a0">FREE_ZVAL</font><font color="4444FF"></font><font
color="#2040a0">line</font><font color="4444FF"></font><font color="4444FF">:</font> <a

```



These Slides: <http://pres.derickrethans.nl/ze-ext>

VLD site: <http://www.derickrethans.nl/vld.php>

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

SRM: <http://www.vl-srm.net>

Index

Introduction	2
Parsing	3
Tokens	4
Tokenizer Example 1	5
Tokenizer Example 2	6
Compiling	7
Oparrays / Opcode	8
Example	9
Disassembling 1	10
Disassembling 2	11
Optimizing 1	12
Optimizing 2	13
Optimizing 3	14
Optimizing 4	15
Optimizing 5	16
Executing 1	17
Executing 2	18
Executing 3	19
aalib - Ascii Art	20
Prototypes	21
Framework generation	22
Generated files	23
Search for the library	27
Check if the library works	28
First test compile	29
Running configure	30
Running configure (2)	31
Extension structure	32
Resource creation	33
Resource destruction	34
Resource fetching	35
Compilation	36
Implementation (1)	37
Parameter Parsing API (1)	38
Parameter Parsing API (2)	39
Implementation (2)	40
Implementation (3)	41
Test script	42
Returning from a function	44
Returning an array	45
Returning a 2D array	46
Resources	47