

# HIDDEN FEATURES OF PHP

ZENDCON 2010  
ILIA ALSHANETSKY  
@ILIAA



# \_\_DIR\_\_ MAGIC

- THE `__DIR__` CONSTANT IS A SIMPLE AND FAST SOLUTION TO THE “WHERE AM I?” QUESTION FOR PHP SCRIPTS.

ILIA@S3 /TMP \$ PHP A.PHP



```
<?PHP ECHO __DIR__;
```



/TMP



# WE ♥ PERL

- ALLOWS QUICK RETRIEVAL OF A NON-EMPTY VALUE FROM 2 VALUES AND/OR EXPRESSIONS

```
$A = TRUE ? FALSE; // TRUE
```

```
$A = FALSE ? TRUE; // TRUE
```

```
$A = "" ? 1; // 1
```

```
$A = 0 ? 2; // 2
```

```
$A = ARRAY() ? ARRAY(1); // ARRAY(1);
```

```
$A = STRLEN("") ? STRLEN("A"); // 1
```

**\*\* THE VARIABLE OR ARRAY KEY MUST EXIST**



# GOTO ...

**RESTART:**

```
IF (ERROR_CONDITION1) {  
    GOTO ERROR;
```

```
}
```

```
IF (ERROR_CONDITION2) {  
    GOTO RESTART;
```

```
}
```

**ERROR:**

```
    REPORT_ERROR();
```

```
    EXIT;
```



```
**** COMMODORE 64 BASIC V2 ****  
64K RAM SYSTEM 38911 BASIC BYTES FREE  
READY.  
10 PRINT "THIS IS AWESOME!";  
20 GOTO 10
```

■ MY FAVOURITE 5.3 FEATURE ;-)



# DIGEST FUNCTIONS

## ■ OPENSSL DIGEST FUNCTIONS

```
// MD5, SHA1, SHA512, ETC... >20 ALGORITHMS  
FOREACH (OPENSSL_GET_MD_METHODS() AS $D) {  
    ECHO OPENSSL_DIGEST("FOO", $D) . "\n";  
    // HEX VALUE  
}
```



# ENCRYPTION FUNCTIONS

## ■ OPENSSL TWO-WAY ENCRYPTION FUNCTIONS

```
$PWD = 'VERY SECRET';
$DATA = 'TEST 123';
// OVER 50 SUPPORTED ALGORITHMS
FOREACH (OPENSSL_GET_CIPHER_METHODS() AS $V) {
    // REALLY BAD IV GENERATION
    $IV = SUBSTR(MD5(TIME()), 0,
                OPENSSL_CIPHER_IV_LENGTH($V));

    // ENCRYPT
    $ENC = OPENSSL_ENCRYPT($DATA, $V, $PWD, FALSE, $IV);

    // DESCRIPT
    $DEC = OPENSSL_DECRYPT($ENC, $V, $PWD, FALSE, $IV);
}
```



# DOUBLE ENCODING

- PREVENT DOUBLE ENCODING OF HTML-ENTITIES VIA 4TH ARGUMENT TO HTMLSPECIALCHARS() AND HTMLENTITIES()

```
$FOO = "BAR > FOO &AMP; THAT&QUOT;S ALL";
```

```
HTMLSPECIALCHARS($FOO, ENT_COMPAT, 'UTF-8');
```

```
HTMLENTITIES($FOO, ENT_COMPAT, 'UTF-8');
```



```
BAR &GT; FOO &AMP;AMP; THAT&AMP;QUOT;S ALL
```

```
HTMLSPECIALCHARS($FOO, ENT_COMPAT, 'UTF-8', FALSE);
```

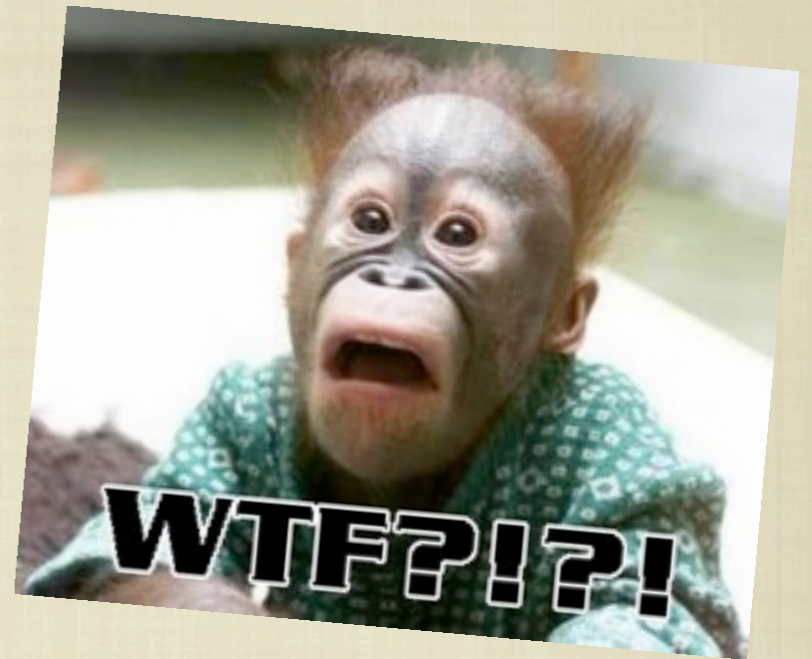
```
HTMLENTITIES($FOO, ENT_COMPAT, 'UTF-8', FALSE);
```



```
BAR &GT; FOO &AMP; THAT&QUOT;S ALL
```

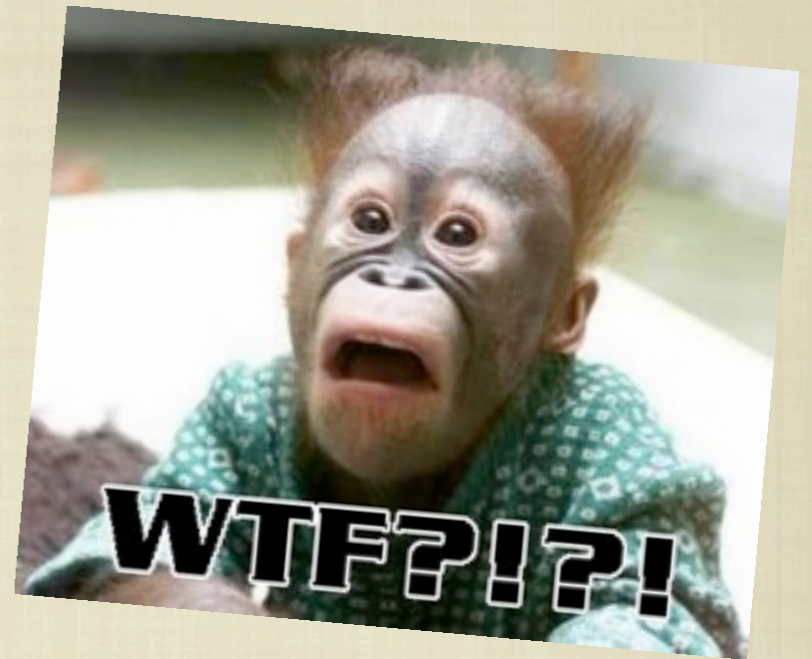


# DATE PARSING





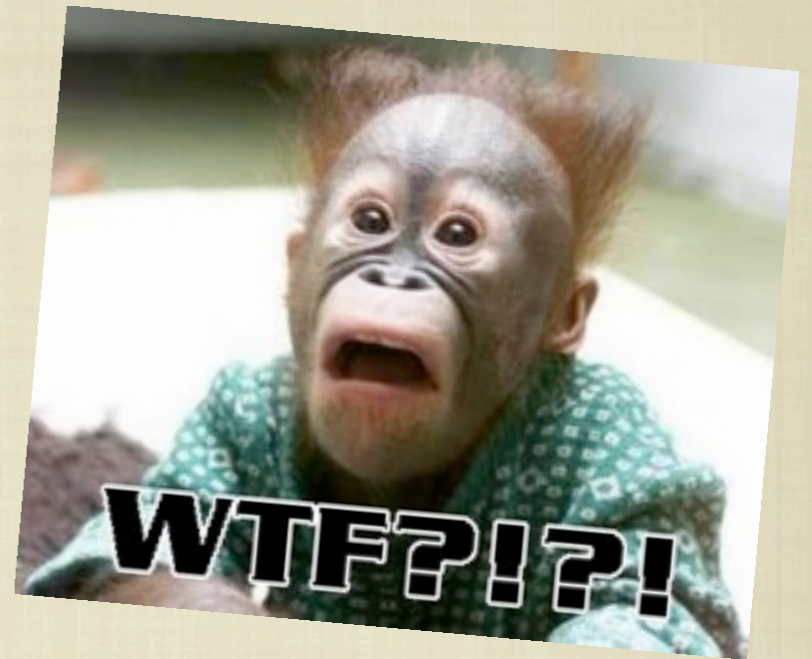
# DATE PARSING



```
$DATE =  
DATE_CREATE_FROM_FORMAT('Y-M-D', '05-10-12');  
VAR_DUMP(DATE_FORMAT($DATE, 'F D, Y'));
```



# DATE PARSING



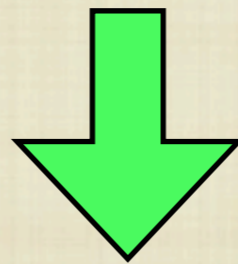
```
$DATE =  
DATE_CREATE_FROM_FORMAT('Y-M-D', '05-10-12');  
VAR_DUMP(DATE_FORMAT($DATE, 'F D, Y'));  
  
STRING(16) "OCTOBER 12, 2005"
```



# DUDE, WHERE IS MY CODE?

- PHP DOES A LOT OF MAGIC TO RESOLVE PARTIAL FILE PATHS FOR INCLUDE/REQUIRE. NOW YOU CAN TOO.

```
STREAM_RESOLVE_INCLUDE_PATH("PEAR.PHP");
```



```
/USR/SHARE/PHP/PEAR.PHP
```



# SESSION INI MAGIC

- IMPROVE RANDOMNESS OF SESSION ID VIA THE USE OF /DEV/URANDOM

```
SESSION.ENTROPY_FILE = /DEV/URANDOM  
SESSION.ENTROPY_LENGTH = 32
```

- SECURE YOUR SESSION COOKIES FROM JAVASCRIPT

```
SESSION.USE_ONLY_COOKIES = 1  
SESSION.COOKIE_HTTPONLY = 1
```



# MAIL LOGGING

- WANT TO KNOW WHAT SCRIPTS ARE SENDING OUT E-MAIL? WELL, NOW YOU CAN!

;; THIS WILL LOG EVERY MAIL() CALL

**MAIL.LOG = /PATH/TO/FILE**

MAIL() ON [/TMP/SCRIPT.PHP:2]: To: [ILIA@ILIA.WS](mailto:ILIA@ILIA.WS) -- HEADERS:

;; ADDS X-PHP-ORIGINATING-SCRIPT HEADER

;; CONTAINS UID & FILENAME OF THE SCRIPT

**MAIL.ADD\_X\_HEADER = ON**

X-PHP-Originating-Script: 1000:script.php



# SPL FS TRICKS

## ■ SIMPLE RECURSIVE DIRECTORY TRAVERSAL

```
FOREACH (  
  NEW RECURSIVEITERATORITERATOR(  
    NEW RECURSIVEDIRECTORYITERATOR('!.')  
  ) AS $FILE) {  
  ECHO $FILE , "\N";  
}
```





# SPL FS TRICKS

## ■ RECURSIVE DIRECTORY TRAVERSAL W/MATCHING

```
$IT = NEW RECURSIVEITERATORITERATOR(  
    NEW RECURSIVEDIRECTORYITERATOR('!')  
);  
$REGX = NEW REGEXITERATOR(  
    $IT,  
    '/^.*\.PHP$/i', // ONLY MATCH WILL BE RETURNED  
    RECURSIVEREGEXITERATOR::GET_MATCH  
);  
  
FOREACH ($REGX AS $FILE) {  
    ECHO $FILE[0], "\n";  
}
```



# IGBINARY

■ THE AWESOME PHP SERIALIZER YOU SHOULD USE!

■ FASTER

■ MORE COMPACT

```
;; LOAD IGBINARY EXTENSION  
EXTENSION=IGBINARY.SO
```

```
;; USE IGBINARY AS SESSION SERIALIZER  
SESSION.SERIALIZE_HANDLER=IGBINARY
```

[HTTP://GITHUB.COM/PHADEJ/IGBINARY](http://github.com/phadej/igbinary)





# IGBINARY

- PROVIDES FUNCTIONS YOU CAN USE FOR NON-SESSION DATA.

```
SERIALIZE($_SERVER);
```

```
INI_SET("IGBINARY.COMPACT_STRINGS", 0);  
IGBINARY_SERIALIZE($_SERVER);
```

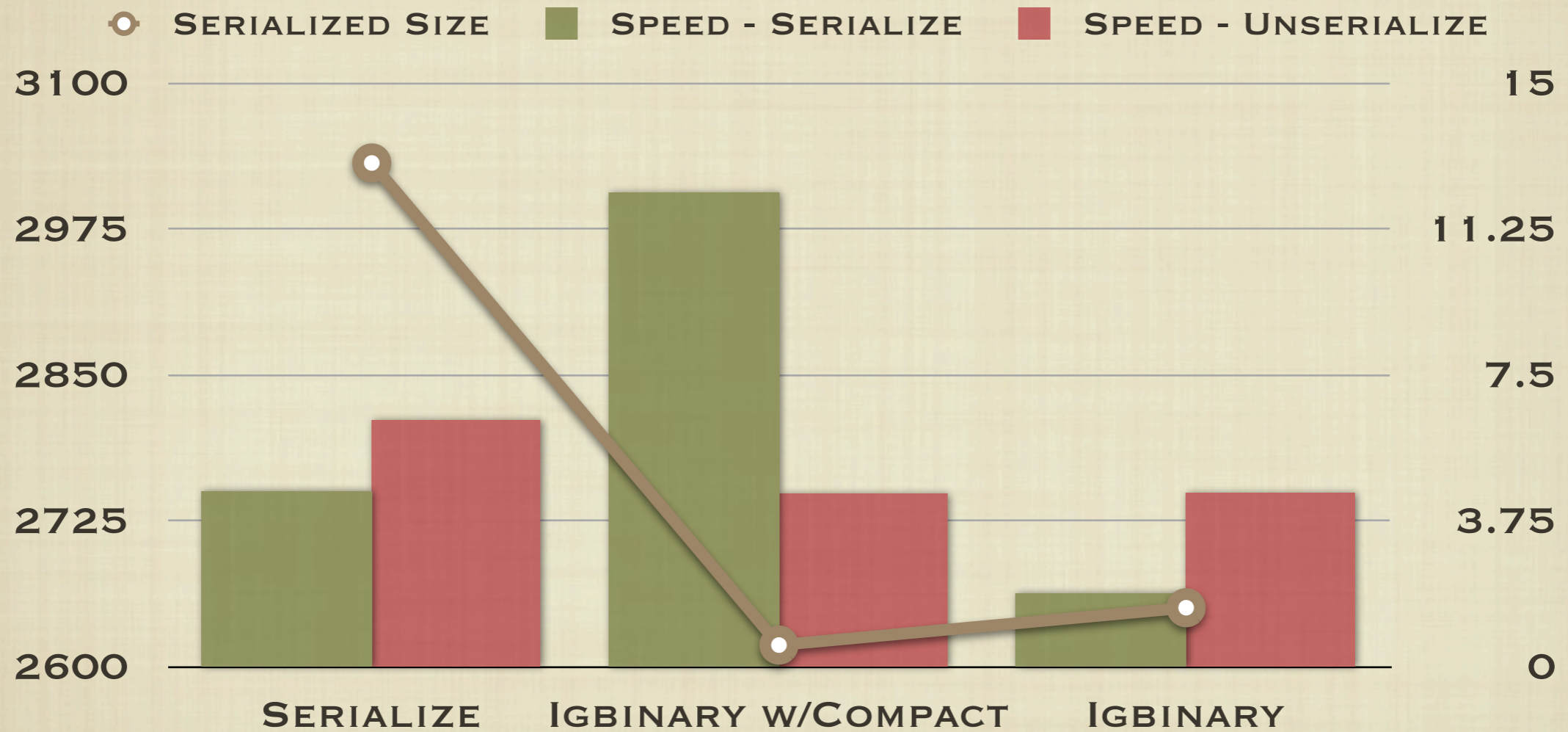
```
INI_SET("IGBINARY.COMPACT_STRINGS", 1);  
IGBINARY_SERIALIZE($_SERVER);
```

```
// UN-SERIALIZE
```

```
IGBINARY_UNSERIALIZE($X);
```



# IGBINARY SPEED TEST



  
**OPTIMAL**



# XHPROF

- LIGHT WEIGHT PHP PROFILER DESIGNED FOR IN PRODUCTION USE.
- AGGREGATE RUN DATA
- WEB INTERFACE
- IN-PRODUCTION “SAMPLING” MODE

[HTTP://PECL.PHP.NET/PACKAGE/XHPROF](http://pecl.php.net/package/xhprof)

[HTTP://GITHUB.COM/PREINHEIMER/XHPROF](http://github.com/preinheimer/xhprof)



# PROFILING

**:: PRE-PENDED TO EVERY PHP SCRIPT (INIT)**

**AUTO\_PREPEND\_FILE = /XHPROF/EXTERNAL/HEADER.PHP**

```
INCLUDE_ONCE __DIR__ . '/XHPROF_LIB/CONFIG.PHP');  
INCLUDE_ONCE __DIR__ . '/XHPROF_LIB/UTILS/XHPROF_LIB.PHP';  
INCLUDE_ONCE __DIR__ . '/XHPROF_LIB/UTILS/XHPROF_RUNS.PHP';  
XHPROF_ENABLE(  
    XHPROF_FLAGS_CPU + XHPROF_FLAGS_MEMORY  
);
```

**:: APPENDED TO EVERY PHP SCRIPT (STORE)**

**AUTO\_APPEND\_FILE = /XHPROF/EXTERNAL/FOOTER.PHP**

```
$XHPROF_DATA = XHPROF_DISABLE();  
$XHPROF_RUNS = NEW XHPROFRUNS_DEFAULT();  
$XHPROF_RUNS->SAVE_RUN($XHPROF_DATA,  
'APPNAME', NULL, $_XHPROF);
```

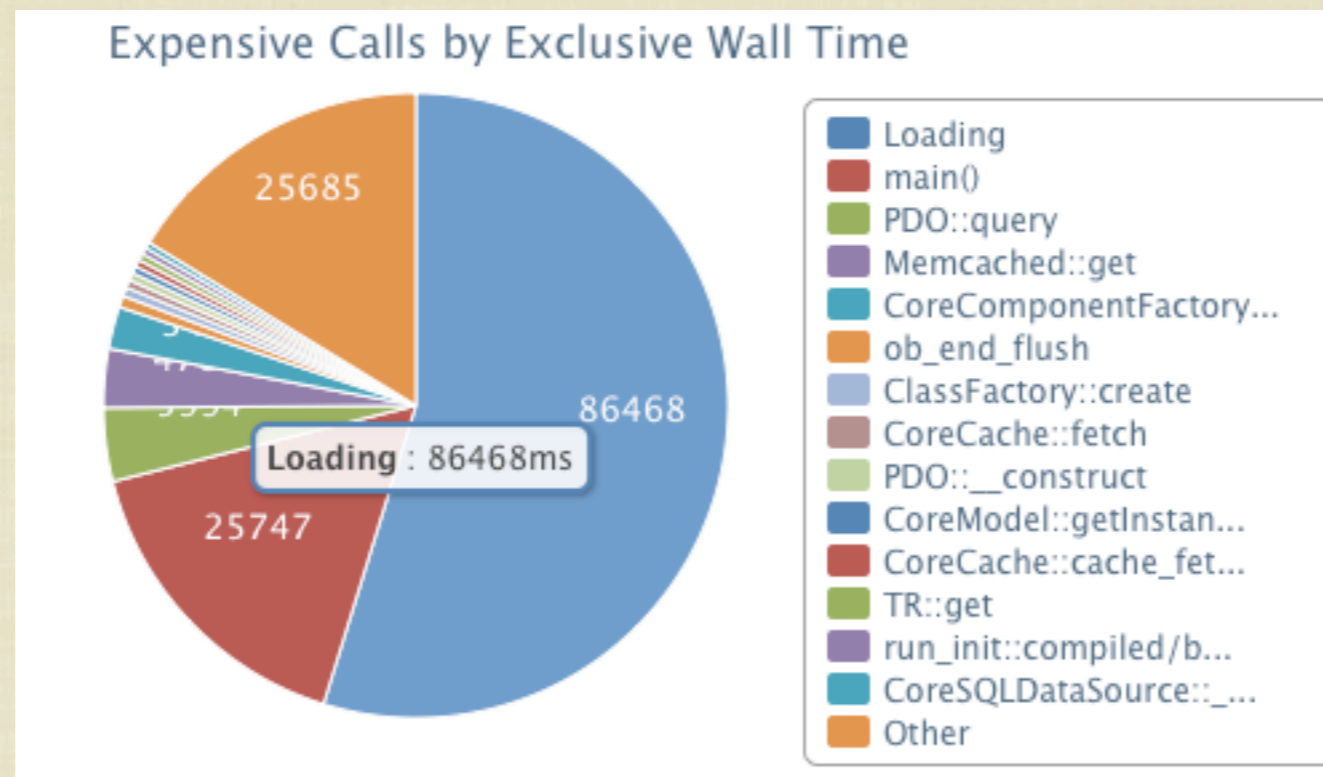


# PROFILE OUTPUT

Stat	Exact URL	Similar URLs
Count	<a href="#">248</a>	<a href="#">248</a>
Min Wall Time	128.3070 ms	128.3070 ms
Max Wall Time	357.8990 ms	357.8990 ms
Avg Wall Time	148.6592 ms	148.6592 ms
95% Wall Time	211.8830 ms	211.8830 ms
<b>Display run Incl. Wall Time (microsec)</b>	<b>157,848 microsecs</b>	
Min CPU Ticks	124.9810 ms	124.9810 ms
Max CPU Ticks	225.9650 ms	225.9650 ms
Avg CPU Ticks	136.6728 ms	136.6728 ms
95% CPU Ticks	152.9780 ms	152.9780 ms
<b>Display run Incl. CPU (microsecs)</b>	<b>149,978 microsecs</b>	
Min Peak Memory Usage	43,985,208 bytes	43,985,208 bytes
Max Peak Memory Usage	47,683,560 bytes	47,683,560 bytes
Avg Peak Memory Usage	44,452,830 bytes	44,452,830 bytes
95% Peak Memory Usage	45,279,104 bytes	45,279,104 bytes
<b>Display run Incl. PeakMemUse (bytes)</b>	<b>44,909,544 bytes</b>	
<b>Number of Function Calls:</b>	<b>1,897</b>	

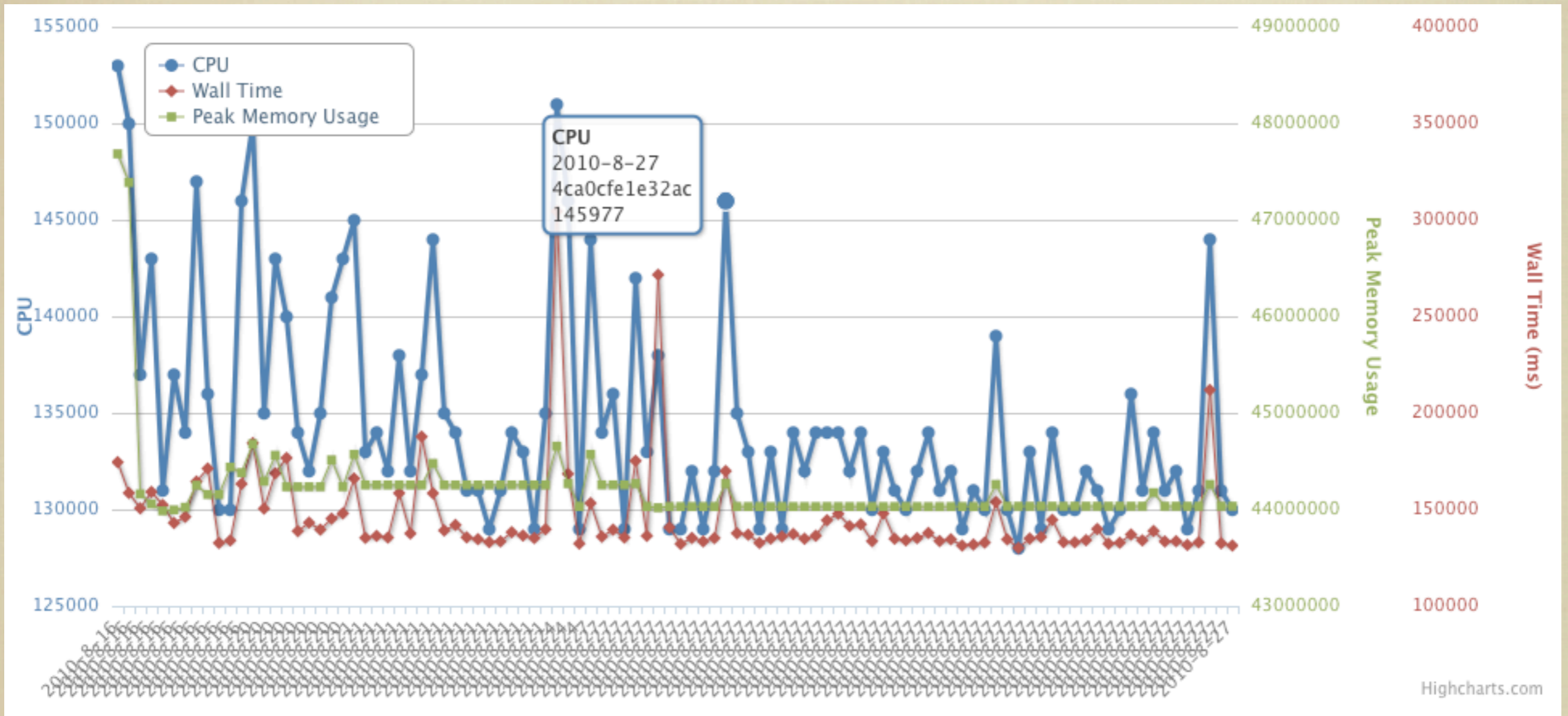


# PROFILE OUTPUT





# PROFILE OUTPUT





# PROFILE OUTPUT

◆ Call Count	◆ Wall Time	◆ CPU	◆ Memory Usage	◆ Peak Memory Usage	◆ Exclusive Wall Time	◆ Exclusive CPU	◆ Exclusive Memory Usage	◆ Exclusive Peak Memory Usage
1	174482	152978	47510112	47683560	20727	20996	4896	5832
1	73589	53993	5102248	5279096	97	0	4744	4064
1	72072	70990	38679368	38679144	72072	70990	38679368	38679144
1	69975	49993	4701800	4813208	179	0	4136	1720
2	43401	27996	4198312	4180048	50	0	-464	1008
2	43351	27996	4198776	4179040	319	0	10320	9304
2	39588	23997	3776408	3778320	125	0	4432	3376
2	39206	23997	3753400	3754384	78	0	-304	1112
5	38915	22998	3745776	3745624	403	0	10536	13808
5	38275	22998	3710200	3713040	2391	3000	29400	19160



# FILEINFO

- A RELIABLE MECHANISM FOR IDENTIFYING FILES
- NOT DEPENDANT ON FILE EXTENSION
- CAN PROVIDE MIME TYPES
- IDENTIFIES HUNDREDS OF FILE TYPES





# FILEINFO HOW-TO

```
$FINFO = FINFO_OPEN();
```

```
$FILE = __FILE__;
```

```
// MIME DESCRIPTION -- PHP SCRIPT TEXT
```

```
FINFO_FILE($FINFO, $FILE);
```

```
// MIME TYPE -- TEXT/X-PHP
```

```
FINFO_FILE($FINFO, $FILE, FILEINFO_MIME_TYPE);
```

```
// MIME -- TEXT/X-PHP; CHARSET=US-ASCII
```

```
FINFO_FILE($FINFO, $FILE, FILEINFO_MIME);
```

```
// MIME ENCODING -- US-ASCII
```

```
FINFO_FILE($FINFO, $FILE, FILEINFO_MIME_ENCODING);
```



# PHP-EXCEL



- AN INTERFACE TO LIBXL LIBRARY
- ALLOWS GENERATION OF EXCEL BIFF8 & XML DOCUMENTS
- CAN PARSE EXCEL BIFF (5-8) AND XML DOCUMENTS
- WICKEDLY FAST! **200K ROWS IN < 1 SECOND**



# CREATING EXCEL DOCS

```
$X = NEW EXCELBOOK();
```

```
$S = $X->ADDSHEET("SHEET 1");
```

```
$S->WRITE(1, 1, 'TEST');
```

```
$S->WRITE(2, 2, 123);
```

```
$X->SAVE("FILE.XLS");
```

	A	B	C
1			
2		Test	
3			123
4			
5			



# READING EXCEL DOCS

```
$X = NEW EXCELBOOK();
```

```
$X->LOADFILE("FILE.XLS");
```

```
$S = $X->GETSHEET();
```

```
FOR ($I = 0, $E = $S->LASTROW(); $I < $E; $I++) {  
    PRINT_R(ARRAY_FILTER($S->READROW($I)));  
}
```

```
ARRAY ( [1] => TEST)
```

```
ARRAY ( [2] => 123 )
```



**THANK YOU FOR  
LISTENING**

**SLIDES WILL BE AVAILABLE  
AT [HTTP://ILIA.WS](http://ilia.ws)**

**PLEASE GIVE ME YOUR FEEDBACK  
[HTTP://JOIND.IN/2250](http://joind.in/2250)**