

# Getting Rich with PHP 5

Apachecon Europe 2006

June 29, 2006. Dublin, Ireland

Rasmus Lerdorf <rasmus@php.net>

<http://talks.php.net/show/apecu06>

## A Rich Web Application...

- o makes better use of browser UI capabilities
- o makes less roundtrips to the server
- o brings together diverse backend data in interesting ways
- o sometimes emulates the look and feel of a desktop app
- o requires a rich presentation layer language

To build one we need...

- o server-side language - PHP
- o to understand backend requests
- o a wire data format - XML and/or JSON
- o to figure out web services
- o some sort of server-side datastore
- o some sort of file organization (MVC?)
- o to at least pretend to be secure
- o plenty of magic smoke

1. Build tagged, socially networked, Web 2.0ish, Ajaxy thing
2. Get 500,000 users quickly (and have it actually work)
3. Profit

500,000 users at an average of 100 front/backend requests daily

= 578 requests/second.

x 3 (because your traffic won't be evenly distributed)

= ~1700 requests/second.

We'll benchmark a representative request to get an initial idea.

## index.php

```
<?php
require_once "config.inc";
require_once "model.inc";
require_once "utils.inc";
require_once "head.inc";
head();
foreach(articles() as $row) {
    $name = upper($row['name']);
    echo <<<EOB
<tr>
<th>$name</th><td>{$row['age']}</td><td>{$row['entry']}</td>
</tr>
EOB;
}
boo();
foot();
?>
```

## config.inc

```
<?php
$config = array(
    'db' => 'pgsql',
    'db_user' => 'nobody',
    'db_pwd' => 'foobar',
    'db_host' => 'localhost',
    'db_db' => 'users',
    'db_opts' => array(PDO::ERRMODE_EXCEPTION => true)
);
?>
```

## model.inc

```
<?php
function articles() {
    global $config;
    try {
        $dbh = new
PDO($config['db'].':host='.$config['db_host'].';dbname='.$config['db_db'],
        $config['db_user'],$config['db_pwd'],
        $config['db_opts']);
        $q = "select users.name, users.age, data.entry
        from users,data where data.name = users.name";
        $result = $dbh->query($q,PDO::FETCH_ASSOC);
    } catch (PDOException $e) {
        echo "Error!: " . $e->getMessage();
        die();
    }
    return $result;
}
?>
```

## head.inc

```
<?php
function head() {
?>
<html><head><title>Index</title></head>
<body>
<table>
<?php
}
```

```
function foot() {  
?>  
</table></body></html>  
<?php  
}  
?>
```

## utils.inc

```
<?php  
function upper($arg) {  
    return strtoupper($arg);  
}  
function boo() {  
    echo "boo";  
}  
?>
```

# Results

Hardware: 1.8GHz AMD Athlon, w/ 1GByte RAM and 3ware 7200rpm drives under a 3ware 7500-4 RAID card using a full-duplex 100MBit dedicated segment with no other traffic

Software: Ubuntu Linux 2.6.15, Apache-1.3.34, PHP 5.1.3-dev, PostgreSQL 8.1

## http\_load

```
7:27pm shiny:~/bm> http_load -parallel 5 -fetches 1000 bml.txt
1000 fetches, 5 max parallel, 6e+06 bytes, in 58.1026 seconds
6000 mean bytes/connection
17.2109 fetches/sec, 103266 bytes/sec
msecs/connect: 0.403263 mean, 68.603 max, 0.194 min
msecs/first-response: 284.133 mean, 5410.13 max, 55.735 min
HTTP response codes:
  code 200 -- 1000
```

## CPU Bound?

```
8:02pm ubuntu:~> vmstat 1
procs -----memory----- ---swap-- -----io-----
--system-- ----cpu----
 r  b   swpd   free   buff   cache    si    so    bi    bo    in
cs us sy id wa
10  0    148  83636 213480 387212     0     0     0     0 1613
1065 90 10  0  0
 5  0    148  87164 213480 387212     0     0     0     0 1460
1013 90 10  0  0
 5  0    148  87288 213480 387216     0     0     0     0 1460
966 91  9  0  0
 7  0    148  84932 213480 387216     0     0     0     0 1459
956 90 10  0  0
 7  0    148  86792 213484 387220     0     0     0    136 1534
1051 93  7  0  0
 7  0    148  86156 213484 387220     0     0     0     0 1484
1003 90 10  0  0
 5  0    148  87768 213484 387224     0     0     0     0 1493
1067 91  9  0  0
```

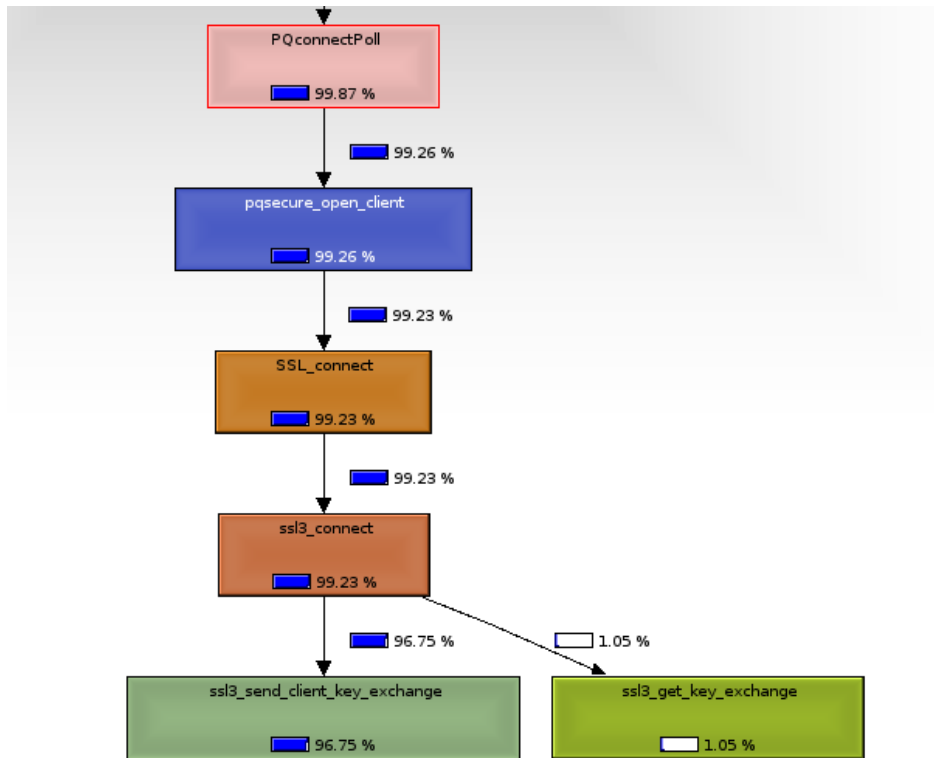
We need 1700 req/sec and we are at 17 req/sec. We need 100 servers!

# Why is it so slow?

## Let's ask Callgrind

```
callgrind --dump-instr=yes --trace-jump=yes -v
/usr/sbin/apache -X
```

## Callgraph [pgssl.out]



So, let's turn off ssl in the postgresql.conf file and try again.

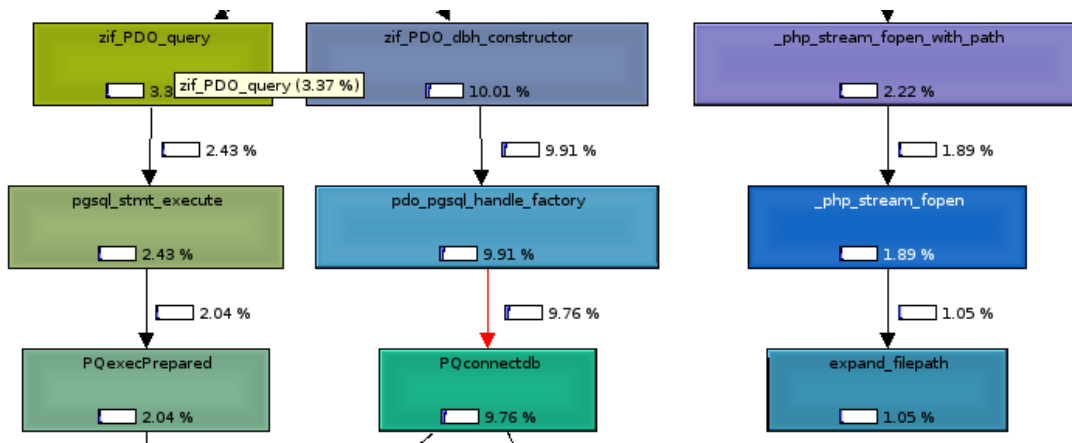
## http\_load

```
1000 fetches, 5 max parallel, 6e+06 bytes, in 9.19585 seconds
6000 mean bytes/connection
108.745 fetches/sec, 652468 bytes/sec
msecs/connect: 0.321531 mean, 23.129 max, 0.219 min
msecs/first-response: 44.8532 mean, 412.765 max, 8.065 min
HTTP response codes:
code 200 -- 1000
```

Better, but another look at the callgraph shows we are still spending 10% of our time connecting to the database and more time tearing the connection back down.

## Callgraph [pgconnect.out]





Turn on persistent connections

```

<?php
$config = array(
    'db' => 'pgsql',
    'db_user' => 'nobody',
    'db_pwd' => 'foobar',
    'db_host' => 'localhost',
    'db_db' => 'users',
    'db_opts' => array(PDO::ERRMODE_EXCEPTION => true,
                      PDO::ATTR_PERSISTENT => true)
);
?>

```

http\_load

```

5000 fetches, 5 max parallel, 3e+07 bytes, in 14.7368 seconds
6000 mean bytes/connection
339.286 fetches/sec, 2.03572e+06 bytes/sec
msecs/connect: 0.404772 mean, 34.54 max, 0.168 min
msecs/first-response: 10.3008 mean, 94.533 max, 2.332 min
HTTP response codes:
code 200 -- 5000

```

Ok, we are down to only needing 5 servers now.

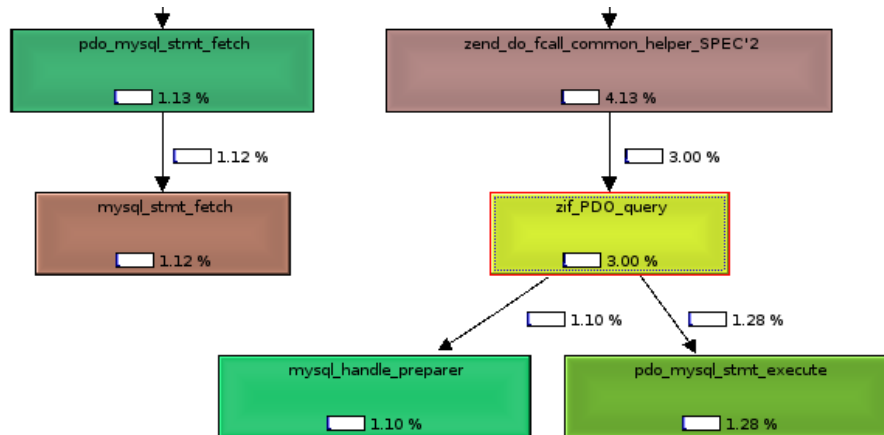
# Let's try MySQL 4.1

Non-persistent: 370 requests/second

Persistent: 525 requests/second

This is with MySQL's query cache enabled. But Callgrind shows us that we are actually calling MySQL's internal prepare/execute API.

## Callgraph [mysql.out]



## Use direct queries

```

<?php
$config = array(
    'db' => 'mysql',
    'db_user' => 'nobody',
    'db_pwd' => 'foobar',
    'db_host' => 'localhost',
    'db_db' => 'users',
    'db_opts' => array(PDO::ERRMODE_EXCEPTION => true,
                      PDO::ATTR_PERSISTENT => true,
                      PDO::ATTR_EMULATE_PREPARES=>true)
);
?>

```

PDO::ATTR\_EMULATE\_PREPARES was added in PHP 5.1.3. You can use PDO::MYSQL\_ATTR\_DIRECT\_QUERY in earlier versions.

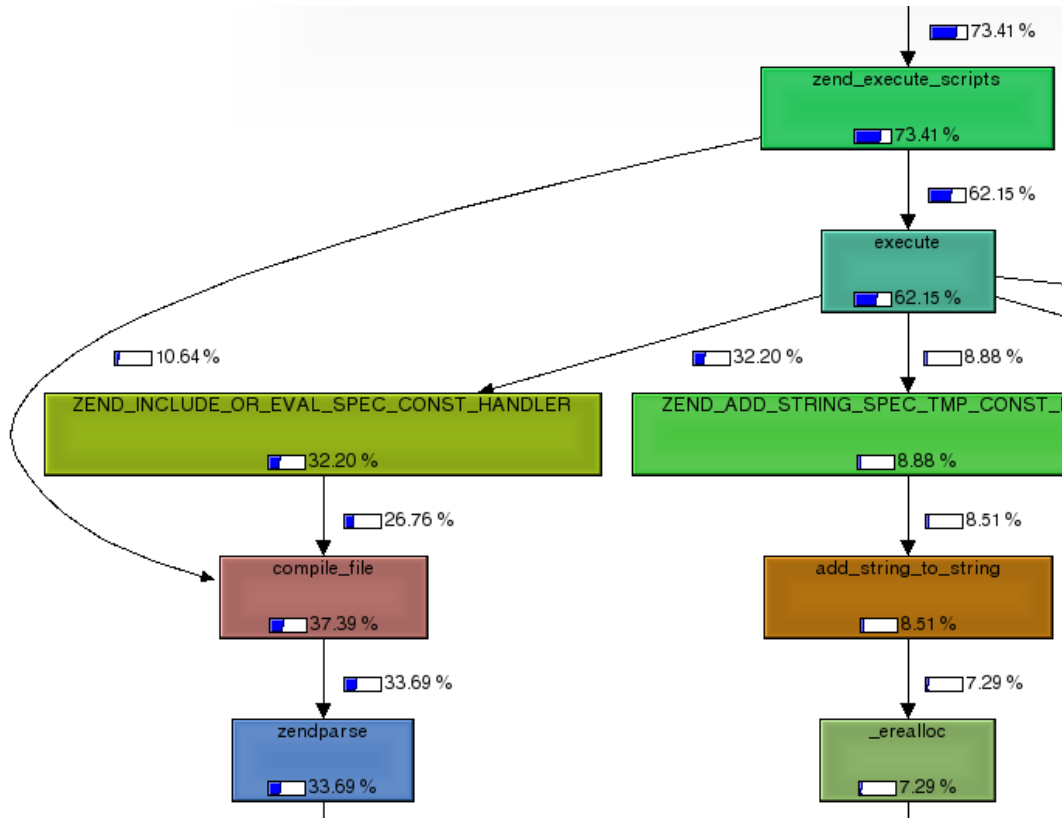
That brings us to 580 requests/second. Down to 3 servers!

The same tests with InnoDB reveal similar numbers.

Sqlite3 comes in at around 550 requests/second with persistent connections.

If we look at the last callgraph again, we see that we are spending close to 40% of our time parsing and compiling. We can eliminate that by using an opcode cache like APC.

## Callgraph [mysql.out]



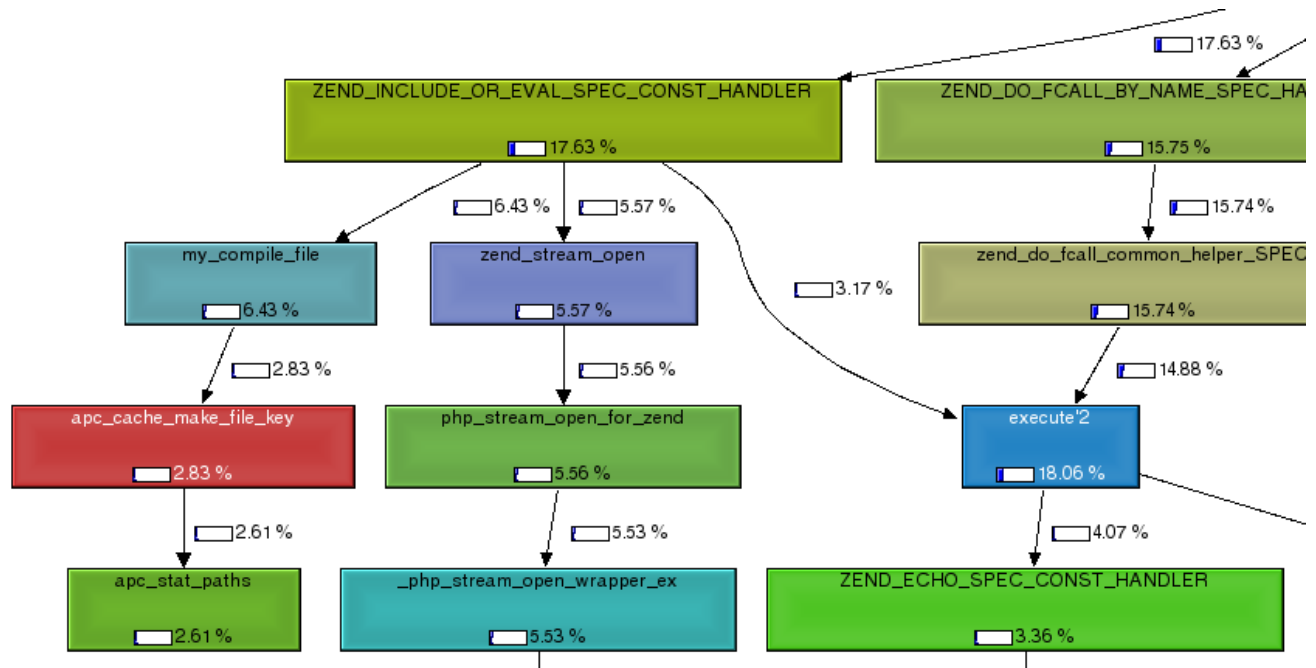
## php.ini

```

extension = "apc.so"
apc.enabled=1
apc.shm_segments=1
apc.shm_size=32
apc.num_files_hint=1024
apc.gc_ttl=3600
apc.ttl=0
apc.mmap_file_mask=/tmp/apc.XXXXXX
apc.filters=
apc.stat=1
apc.enable_cli=0
  
```

805 requests/second. Callgraph?

## Callgraph [apc.out]



If we look carefully we see that over 5000 requests we are opening 20000 files. Main script plus 4 includes on each request. This is because include\_once and require\_once don't play nice with opcode caches right now. Changing to require brings us to 875 requests/second.

APC also has a no-stat mode. If you give it absolute paths, it can skip the stat() call.

## php.ini

```
apc.stat=0
```

## config.inc

```
<?php
$config = array(
    'db'      => 'mysql',
    'db_user' => 'nobody41',
    'db_pwd'  => 'foobar',
    'db_host' => 'localhost',
    'db_db'   => 'users',
    'db_opts' => array(PDO::ERRMODE_EXCEPTION => true,
                      PDO::ATTR_PERSISTENT => true,
                      PDO::MYSQL_ATTR_DIRECT_QUERY=>1),
    'path'    => getcwd()
);
?>
```

## index.php

```
<?php
require "./config.inc";
require $config['path']."/model.inc";
require $config['path']."/utils.inc";
require $config['path']."/head.inc";
head();
foreach(articles() as $row) {
    $name = upper($row['name']);
    echo <<<EOB
    <tr><th>$name</th><td>{$row['age']}</td><td>{$row['entry']}</td></tr>
    EOB;
}
boo();
foot();
?>
```

This takes us to 885 requests/second. We are down to 2 machines!

APC has a nice feature where you can store PHP variables in shared memory. We can use that for our config array.

## index.php

```
<?php
if(!$config = apc_fetch('config')) {
    require "./config.inc";
    apc_store('config',$config);
}
...
```

This takes us to 890 requests/second.

Of course, we can take this further and also cache the result of the database query for 5 minutes.

## model.inc

```
<?php
function articles() {
    global $config;
    if(!$result = apc_fetch('result')) {
        try {
            $dbh = new
PDO($config['db'].' :host='.$config['db_host'].';dbname='.$config['db_db'],
$config['db_user'],$config['db_pwd'],$config['db_opts']);
            $q = "select users.name, users.age, data.entry
                from users,data where data.name = users.name;";
            $res = $dbh->query($q,PDO::FETCH_ASSOC);
            $result = $res->fetchAll();
            apc_store('result',$result,300);
        } catch (PDOException $e) {
            echo "Error!: " . $e->getMessage();
            die();
        }
    }
    return $result;
}
?>
```

This takes us to 1080 requests/second.

A final optimization is to get rid of an include file. Our utils.inc file was a bit useless.

## utils.inc

```
<?php
function upper($arg) {
    return strtoupper($arg);
}
function boo() {
    echo "boo";
}
?>
```

We can add those 2 functions to model.inc and remove the require call. Now we get:

## http\_load

```
15000 fetches, 5 max parallel, 9e+07 bytes, in 13.6243 seconds
6000 mean bytes/connection
1100.97 fetches/sec, 6.60585e+06 bytes/sec
msecs/connect: 0.295239 mean, 17.802 max, 0.114 min
```

msecs/first-response: 3.61389 mean, 1097.1 max, 0.739 min  
HTTP response codes:  
code 200 -- 15000

Compare with our original:

## http\_load

1000 fetches, 5 max parallel, 6e+06 bytes, in 58.1026 seconds  
6000 mean bytes/connection  
17.2109 fetches/sec, 103266 bytes/sec  
msecs/connect: 0.403263 mean, 68.603 max, 0.194 min  
msecs/first-response: 284.133 mean, 5410.13 max, 55.735 min  
HTTP response codes:  
code 200 -- 1000

## PHP 5





```

        <price>$6.95</price>

        <description>two eggs, <bacon or sausage, <toast

        <calories>950</calories>

    </food>

</breakfast_menu>

```

## XPath

```

<?php
$dom = domdocument::load('menu.xml');
$ctx = new domXPath($dom);
$result = $ctx->query('/breakfast_menu/food[@itemno >
3]/price/text()');
foreach($result as $node) {
    echo $node->nodeValue."<br />\n";
}
?>

```

### Output:

```

$4.50
$6.95

```

## XSL

```

<?php
$dom = domdocument::load('menu.xml');
$domxsl = domDocument::load('menu.xsl');
$proc = new xsltProcessor;
$proc->importStyleSheet($domxsl);
echo $proc->transformToXML($dom);
?>

```

### Output:

```

Belgian Waffles
- $5.95two of our famous Belgian Waffles with
  plenty of real maple syrup
  (650 calories per serving)
Strawberry Belgian Waffles
- $7.95light Belgian waffles covered with strawberries
  and whipped cream
  (900 calories per serving)
Berry-Berry Belgian Waffles
- $8.95light Belgian waffles covered with an
assortment of
  fresh berries and whipped cream
  (900 calories per serving)
French Toast
- $4.50thick slices made from our homemade sourdough
bread
  (600 calories per serving)
Homestyle Breakfast
- $6.95two eggs, bacon or sausage, toast, and
  our ever-popular hash browns
  (950 calories per serving)

```

## SimpleXML

```

<?php

```

```
$menu = simplexml_load_file('menu.xml');
foreach ($menu->food as $item) {
    echo $item['itemno'] . " ) ";
    echo $item->price . " ... ";
    echo $item->name . "<br />\n";
}
?>
```

### Output:

```
1) $5.95 ... Belgian Waffles
2) $7.95 ... Strawberry Belgian Waffles
3) $8.95 ... Berry-Berry Belgian Waffles
4) $4.50 ... French Toast
5) $6.95 ... Homestyle Breakfast
```

## Recent Flickr Photos

```
<?php
$url =
'http://www.flickr.com/services/feeds/photos_public.gne';
foreach(simplexml_load_file($url)->entry as $it) echo
$it->content;
?>
```

### Output:

Monkey Magic posted a photo:

lasterato posted a photo:

pflood33 posted a photo:

tobbycat posted a photo:

Karimerejames posted a photo:

monstrum\_xxx posted a photo:

ellis\_sctt posted a photo:

collard\_greens posted a photo:

zzcoyote posted a photo:

yuricheung.zihao posted a photo:

## Recent comp.lang.php messages

```
<?php
$feed =
'http://groups.google.com/group/comp.lang.php/feed/atom_v1_0_msgs.xml';
$xml = simplexml_load_file($feed);
echo "<h1>{$xml->title}</h1><h2>{$xml->subtitle}</h2>";
foreach($xml->entry as $entry) {
    $date = date("r",strtotime($entry->updated));
    echo <<<EOB
<div style="background: #000; width:54em; padding:2px;">
    <div style="background: #ffa; padding: 5px;">
        From: {$entry->author->name}<br />
        Date: $date<br />
        Subject: <a
href="{ $entry->link['href'] }">{$entry->title}</a><br />
    </div>
    <div style="background: #ffd; padding: 5px; border-top: 2px
dotted;">
        { $entry->summary}<br />
    </div>
</div>
<br clear="left"/>
EOB;
}
?>
```

## Output:

```
comp.lang.php Google Group
PHP, server side scripting.
```

```
From: Robert S robert.spam.me.sensel...@gmail.com
Date: Fri, 07 Jul 2006 06:22:35 -0700
Subject: Re: method="POST" changes '.' to '_'
```

I've gone for this approach. Works a treat. Many thanks for the help folks.

```
From: Otto J. Makela o...@iki.fi
Date: Fri, 07 Jul 2006 06:15:45 -0700
Subject: Re: Perl-1.0.0
```

To make it absolutely clear: Perl-1.0.0 (note the capital P) is a php-add-on. It makes it possible to use existing perl libraries from php. Please see [link] for more info. [...] Exactly. [...] Please check out how I posted my original question: Newsgroups: comp.lang.php,comp.lang.perl.m isc

```
From: Rik luiheidsgoe...@hotmail.com
Date: Fri, 07 Jul 2006 05:56:27 -0700
Subject: Re: Perl-1.0.0
```

Euhm, Jerry, a PECL PERL interpreter for PHP has a lot to do with PHP itself I'd imagine... There's a good chance other PHP-users thought this a usefull add-on and know the issues concerning installing it. Some, not many. The post isn't totally off-base, but I'd imagine people in for instance c.l.perl could be of more assistance in this particular case, that's true.

From: Jerry Stuckle jstuck...@attglobal.net  
Date: Fri, 07 Jul 2006 05:37:42 -0700  
Subject: Re: Perl-1.0.0

I understand the question, all right. What I don't understand is why you're posting this in a PHP newsgroup. You're trying to compile a Perl package. Or haven't you figured out there is a difference between Perl and PHP? How many of us do you think have ever even tried to compile Perl packages?

From: Michael L michaelly...@gmail.com  
Date: Fri, 07 Jul 2006 05:33:37 -0700  
Subject: Need serious help with a recursive(?) tree function

Hi Guys(I apologize for the lengthy post - Im trying to explain it as best i can) I've been cracking my head on this one for the past 24+ hours and i have tried creating the function in ten different ways and none of the versions i've made works exactly as it should. I have an array called \$PageArray which contains a sorted list of all

From: Otto J. Makela o...@iki.fi  
Date: Fri, 07 Jul 2006 05:27:07 -0700  
Subject: Re: Perl-1.0.0

Please don't post if you don't understand the question.

From: jerryyang\_...@yahoo.com  
Date: Fri, 07 Jul 2006 05:04:22 -0700  
Subject: PHP File Management with User Folders.

I'm looking for a script / scripts that will allow me to control files that users can download. I would like to create 'General Folder' that any registered user can download from and private folders that only the specified registered user can download from. This needs to be simple to manage and simple to download from.

From: Jerry Stuckle jstuck...@attglobal.net  
Date: Fri, 07 Jul 2006 04:57:37 -0700  
Subject: Re: Strategy for securing MySQL PHP application - please comment

You shouldn't need to use `html_entity_decode()` for output to non-web presentations, as you shouldn't have html entities in the database anyway. You should also validate numeric input to ensure it is numeric. It will prevent other types of SQL injection, i.e. `DELETE FROM table WHERE id=$i;` And the user keys in

From: David Haynes david.hayn...@sympatico.ca  
Date: Fri, 07 Jul 2006 04:54:32 -0700  
Subject: Re: Editing a (config) file

I would read the file into an associative array, update the value based on the key, write a new file, rename it to the original. I might also `flock()` the original configuration file if the operating system does not support atomic rename/move. -david-

From: Jerry Stuckle jstuck...@attglobal.net  
Date: Fri, 07 Jul 2006 04:50:34 -0700  
Subject: Re: Perl-1.0.0

So, what's your PHP question? Looks like you're trying to compile Perl modules - so you should start in the Perl newsgroups.

From: David Haynes david.hayn...@sympatico.ca  
Date: Fri, 07 Jul 2006 04:50:03 -0700  
Subject: Re: Time help

As others have pointed out, you could just add 2 hours (in seconds) to the offset but what about daylight saving time? A better way is to set the timezone value. e.g. `// save the current timezone $my_tz = date_default_timezone_get(); // set the new timezone date_default_timezone_set('America/New York');`

From: Jerry Stuckle jstuck...@attglobal.net  
Date: Fri, 07 Jul 2006 04:36:11 -0700  
Subject: Re: method="POST" changes '.' to '\_'

OK, no problem. Just: `<form action=test.php method=post><input type=hidden name="filename[0]" value="filename.txt"><input type=submit name="Submit[0]"><input type=hidden name="filename[1]" value="filename.pdf"><input type=submit name="Submit[1]"><input type=hidden name="filename[2]" value="filename.doc">`

From: Alexander Curvers acurv...@gmail.com  
Date: Fri, 07 Jul 2006 04:21:54 -0700  
Subject: add attributes elements to an XML tag?

Hi Whats the PHP code to add an XML tag with an attribute element in my SOAP message body. With output formatted corresponding to the following example: <PRSADRVBL soortEntiteit="R"> .....<other xml>..... </PRSADRVBL> Kind regards  
Alexander Curvers

From: Jerry Stuckle jstuck...@attglobal.net  
Date: Fri, 07 Jul 2006 04:19:00 -0700  
Subject: Re: Confused by mysqli

They are two different tables - so typically they have two different routines. In this case one size does NOT fit all. And why is that? You're accessing different data - so you need different code. Sure. But they're two different tables. I'm not surprised. Short cuts rarely are. That's right, because you're trying to mix bind and non-bind functions.

From: Manish yehaiman...@gmail.com  
Date: Fri, 07 Jul 2006 04:11:54 -0700  
Subject: Deleting duplicate entries from MySQL database table

The topic is related to MySQL database. Suppose a table "address" contains the following records

-----  
-----  
-----  
-----



With PHP5's solid XML support and improved internal OOP support, Web Services are now a natural fit for PHP. Most people seem to think of SOAP when we say Web Services, so here is the obligatory SOAP example:

```
<?php
$amazon_index = array(
    'DVD', 'Photo', 'Electronics', 'OfficeProducts',
    'HealthPersonalCare',
    'Toys', 'Baby', 'VideoGames', 'MusicTracks',
    'OutdoorLiving',
    'Blended', 'MusicalInstruments', 'Magazines',
    'DigitalMusic',
    'Jewelry', 'Video', 'Tools', 'PCHardware', 'SportingGoods',
    'Classical', 'Software', 'Books', 'VHS', 'Wireless',
    'Restaurants',
    'Music', 'GourmetFood', 'Miscellaneous', 'Kitchen',
    'WirelessAccessories',
    'Merchants', 'Beauty', 'Apparel'
);

function amazon($index, $keywords, $timeout=7200) {
    $dest_file = "/tmp/aws_{$index}_".md5($keywords);
    if(file_exists($dest_file) && filemtime($dest_file) >
(time()-$timeout)) {
        $result = unserialize(file_get_contents($dest_file));
    } else {
        $aws = new SoapClient('http://webservices.amazon.com/'.
'AWSECommerceService/US/AWSECommerceService.wsdl',
            array("trace" => 1));
        $result = $aws->ItemSearch(array(
            'SubscriptionId'=>'XXXXXXXXXXXXXXXXX',
            'AssociateTag'=>'lerdorf-20',
            'Request'=>array(array('SearchIndex'=>$index,
                                'Keywords'=>$keywords))
        ));
    }
    $tmpf = tempnam('/tmp','YWS');
    file_put_contents($tmpf, serialize($result));
    rename($tmpf, $dest_file);
}
return $result;
}
?>
```

```
<?php
function Add($x,$y) {
    return $x+$y;
}

$server = new
SoapServer(null,array('uri'=>"http://test-uri/"));
$server->addFunction("Add");
$server->handle();
?>
```

Or using a WSDL:

```
<?php
function Add($x,$y) {
    return $x+$y;
}

$server = new SoapServer("./add.wsdl");
$server->addFunction("Add");
$server->handle();
?>
```

## add.wsdl

```
<?xml version="1.0" ?>
<definitions
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:si="http://soapinterop.org/xsd"
  xmlns:tns="http://localhost/~rasmus/pecl/soap/test.wsdl"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://localhost/~rasmus/pecl/soap/test.wsdl">

  <types>
    <xsd:schema
      targetNamespace="http://localhost/~rasmus/pecl/soap/test.wsdl">
      <xsd:import
        namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
    </xsd:schema>
  </types>

  <message name="AddRequest">
    <part name="x" type="xsd:double" />
    <part name="y" type="xsd:double" />
  </message>
  <message name="AddResponse">
    <part name="result" type="xsd:double" />
  </message>

  <portType name="TestServicePortType">
    <operation name="Add">
      <input message="tns:AddRequest" />
      <output message="tns:AddResponse" />
    </operation>
  </portType>

  <binding name="TestServiceBinding"
    type="tns:TestServicePortType">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="Add">
      <soap:operation soapAction="Add" style="rpc" />
      <input>
```

```

        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
    <output>
        <soap:body use="encoded"

encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </output>
</operation>
</binding>

<service name="TestService">
    <port name="TestServicePort"
binding="tns:TestServiceBinding">
        <soap:address
location="http://localhost/~rasmus/pecl/soap/soap_server.php" />
    </port>
</service>

</definitions>

```

SOAP makes my head hurt. Let's look at some stuff we can all actually understand instead.

## Take a pinch of RSS

```
<?php
$url = 'http://buzz.yahoo.com/feeds/buzzover1.xml';
$xml = @simplexml_load_file($url);
if($xml) {
    foreach($xml->channel->item as $item) {
        $ret[(string)$item->title] = (string)$item->link;
    }
    echo "<pre>"; print_r($ret); echo "</pre>";
} else echo "Can't get to Yahoo!";
?>
```

### Output:

```
Array
(
    [1. World Cup 2006] =>
http://search.yahoo.com/search?p=world+cup+2006&cs=bz
    [2. Britney Spears] =>
http://search.yahoo.com/search?p=britney+spears&cs=bz
    [3. WWE] => http://search.yahoo.com/search?p=wwe&cs=bz
    [4. Cartoon Network] =>
http://search.yahoo.com/search?p=cartoon+network&cs=bz
    [5. Keira Knightley] =>
http://search.yahoo.com/search?p=keira+knightley&cs=bz
    [6. Anne Hathaway] =>
http://search.yahoo.com/search?p=anne+hathaway&cs=bz
    [7. Wikipedia] =>
http://search.yahoo.com/search?p=wikipedia&cs=bz
    [8. Superman Returns] =>
http://search.yahoo.com/search?p=superman+returns&cs=bz
    [9. Paris Hilton] =>
http://search.yahoo.com/search?p=paris+hilton&cs=bz
    [10. Cristiano Ronaldo] =>
http://search.yahoo.com/search?p=cristiano+ronaldo&cs=bz
    [11. FIFA] => http://search.yahoo.com/search?p=fifa&cs=bz
    [12. hilary swank] =>
http://search.yahoo.com/search?p=hilary+swank&cs=bz
    [13. Shakira] =>
http://search.yahoo.com/search?p=shakira&cs=bz
    [14. RuneScape] =>
http://search.yahoo.com/search?p=runescape&cs=bz
    [15. Jessica Simpson] =>
http://search.yahoo.com/search?p=jessica+simpson&cs=bz
)
```

## Add a spoonful of REST

```
<?php
$srvc='http://api.search.yahoo.com/ImageSearchService/V1/imageSearch';
foreach($ret as $key=>$link) {
    $url = $srvc . "?query=$key&appid=RESTDemo";
    $obj = simplexml_load_file($url);
}
?>
```

## A thimble of gradeschool math

$$x^2/a^2 + y^2/b^2 = 1$$

And we end up with something like this

<http://buzz.progphp.com>

By the way, throwing a cache layer between you and whatever remote service you are accessing tends to be a good idea. In our case we can do it like this:

```
<?php
$tmp = '/tmp/'.md5($q);
if(!file_exists($tmp) || filemtime($tmp) < (time()-7200)) {
    $stream = fopen($url,'r');
    $tmpf = tempnam('/tmp','YWS');
    file_put_contents($tmpf, $stream);
    fclose($stream);
    rename($tmpf, $tmp);
}
$obj = simplexml_load_file($tmp);
?>
```

Finally, a free geocoding API

## REST Query to look up an address

```
http://api.local.yahoo.com/MapsService/V1/geocode?appid=YahooDemo&location=-:location:
```

## Returned XML

```
<?xml version="1.0"?>
<ResultSet
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:yahoo:maps"
  xsi:schemaLocation="urn:yahoo:maps
http://api.local.yahoo.com/MapsService/V1/GeocodeResponse.xsd">
  <Result precision="city" warning="The street
could not be found. Here is the center of the city.">
    <Latitude>39.21347</Latitude>
    <Longitude>-79.637589</Longitude>
    <Address></Address>
    <City>LOCATION</City>
    <State>WV</State>
    <Zip></Zip>
    <Country>US</Country>
  </Result>
</ResultSet>

<!-- ws01.search.scd.yahoo.com uncompressed Fri Jul 7
06:57:00 PDT 2006 -->
```

## We can parse it like this

```
<?php
$url = 'http://api.local.yahoo.com/MapsService/V1/geocode';
$url .=
'?appid=YahooDemo&location='.urlencode(':-:location:-:');
$xml = simplexml_load_file($url);
$ret['precision'] = (string)$xml->Result['precision'];
$ret['warning'] = (string)$xml->Result['warning'];
foreach($xml->Result->children() as $key=>$val) {
    if(strlen($val)) $ret[(string)$key] = (string)$val;
}
echo "<pre>"; print_r($ret); echo "</pre>";
?>
```

## Output:

```
Array
(
    [precision] => city
    [warning] => The street could not be found. Here is the
center of the city.
    [Latitude] => 39.21347
    [Longitude] => -79.637589
    [City] => LOCATION
    [State] => WV
    [Country] => US
)
```

## Closest Match

```
http://api.local.yahoo.com/MapsService/V1/geocode?appid=YahooDemo&location=70120First,20
```

## Returned XML

```
<ResultSet xsi:schemaLocation="urn:yahoo:maps
http://api.local.yahoo.com/MapsService/V1/GeocodeResponse.xsd">
```

```
<Result precision="address"
      warning="The exact location could not be found,
              here is the closest match: 701 First Ave,
Sunnyvale, CA 94089">
  <Latitude>37.416384</Latitude>
  <Longitude>-122.024853</Longitude>
  <Address>701 FIRST AVE</Address>
  <City>SUNNYVALE</City>
  <State>CA</State>
  <Zip>94089-1019</Zip>
  <Country>US</Country>
</Result>
</ResultSet>
```

Docs: <http://developer.yahoo.net/maps/>

You can also ask for a single map tile of any location at any size.

## REST Query to get a tile

`http://api.local.yahoo.com/MapsService/V1/mapImage?appid=YahooDemo&location=94539`

## Returned XML

```
<Result>
http://img.maps.yahoo.com/mapimage?MAPDATA=NE6uked6wXUvMKeeO65.zg9FRNGF36TLiC8QU14ROsDkK
</Result>
```

## Example

```
<html><head><title>Map Tile Demo</title>
<script type="text/javascript">
<?php
function request_cache($url, $dest_file, $timeout=43200) {
    if(!file_exists($dest_file) || filemtime($dest_file) <
(time()-$timeout)) {
        $stream = fopen($url,'r');
        $tmpf = tempnam('/tmp','YWS');
        file_put_contents($tmpf, $stream);
        fclose($stream);
        rename($tmpf, $dest_file);
    }
}

$cities = array("San Francisco", "Los Angeles", "Seattle",
"Atlanta", "Chicago", "Dallas",
"Vancouver, Canada", "New York City",
"Toronto", "Miami, FL", "Boston");

$width = 600; $height = 300; $zoom = 7;

$url =
"http://api.local.yahoo.com/MapsService/V1/mapImage?appid=YahooDemo";

$url .= "&image_height=$height&image_width=$width&zoom=$zoom";
$url .= "&location=";
$i=0;
foreach($cities as $city) {
    $q = $url.rawurlencode($city);
    $tmp = '/tmp/yws_tile_' . md5($q);
    request_cache($q, $tmp);
    $xml = simplexml_load_file($tmp);
    if($i==0) $first = (string)$xml;
    echo <<<EOB
image$i = new Image();
image$i.src = "$xml";
EOB;
    $i++;
}
?>
</script>
</head><body>
<?php
echo <<<EOB
<div id="img_container"
    style="float:left; height:{$height}px; width:{$width}px;
padding: 0px 10px 0px 10px;">
</div>
EOB;
$i=0;
foreach($cities as $city) {
    $q = rawurlencode($city);
```



```

    echo <<<EOB
<a href="http://maps.yahoo.com/beta/index.php?#trf=0&q1=$q"
onmouseover="document.getElementById('the_img').src=image$i.src;
              this.style.backgroundColor='white';"

onmouseout="this.style.backgroundColor='#eeee33';">$city</a><br
/>
EOB;
    $i++;
}
?>
</body></html>

```

## Example

Map Tile Demo

```

image0 = new Image();
image0.src = "http://localhost/maps/0.png";image1 = new
Image();
image1.src = "http://localhost/maps/1.png";image2 = new
Image();
image2.src = "http://localhost/maps/2.png";image3 = new
Image();
image3.src = "http://localhost/maps/3.png";image4 = new
Image();
image4.src = "http://localhost/maps/4.png";image5 = new
Image();
image5.src = "http://localhost/maps/5.png";image6 = new
Image();
image6.src = "http://localhost/maps/6.png";image7 = new
Image();
image7.src = "http://localhost/maps/7.png";image8 = new
Image();
image8.src = "http://localhost/maps/8.png";image9 = new
Image();
image9.src = "http://localhost/maps/9.png";image10 = new
Image();
image10.src = "http://localhost/maps/10.png";

```

San FranciscoLos AngelesSeattleAtlantaChicagoDallasVancouver,  
CanadaNew York CityTorontoMiami, FLBoston

# Embedded Map API

There are actually 4 different APIs. Actionscript-Flash, Flex-Flash, Javascript controlling a Flash map, and Javascript controlling an AJAX/DHTML map. I know nothing about Actionscript, nor Flex. Sorry.

## Simple Embedded Flash map

```
<script type="text/javascript"

src="http://api.maps.yahoo.com/v3.0/fl/javascript/apiloader.js?appid=YahooDemo">
</script>
<div id="mapContainer" style="height: 400px; width:
930px;"></div>
<script type="text/javascript">
var mymap = new
Map("mapContainer", "YahooDemo", ":-:location:-:", 10, MapViews.HYBRID);
mymap.addTool(new PanTool(), true);
mymap.addWidget(new SatelliteControlWidget());
mymap.addWidget(new NavigatorWidget());
</script>
```

Output:

```
var mymap = new
Map("mapContainer", "YahooDemo", ":-:location:-:", 10, MapViews.HYBRID);
mymap.addTool(new PanTool(), true);
mymap.addWidget(new SatelliteControlWidget());
mymap.addWidget(new NavigatorWidget());
```

It's actually a little bit disappointing how simple it is. We don't even need PHP here. Of course, chances are you are going to put things onto the map, in which case you are likely to need PHP to fetch data from a backend to plot.

## Simple Embedded DHTML map

```
<script type="text/javascript"

src="http://api.maps.yahoo.com/ajaxymap?v=3.0&appid=YahooDemo">
</script>
<div id="jsmapContainer" style="height: 400px; width:
930px;"></div>
<script type="text/javascript">
var jsmap = new
YMap(document.getElementById('jsmapContainer'));
jsmap.addTypeControl();
jsmap.addZoomLong();
jsmap.setMapType('YAHOO_SAT');
jsmap.drawZoomAndCenter(":-:location:-:", 13);
</script>
```

Output:

```
var jsmap = new
YMap(document.getElementById('jsmapContainer'));
jsmap.addTypeControl();
jsmap.addZoomLong();
jsmap.setMapType('YAHOO_SAT');
jsmap.drawZoomAndCenter(":-:location:-:", 13);
```

# Adding Markers

You can either add markers one by one by calculating a location and adding the marker, or in the Flash API you can use an overlay. There is a very handy GeoRSS overlay that you just feed a Geocoded RSS file to. Each entry needs to have this:

## GeoRSS Sample Item

```
<item>
  <title>M 3.3, Southern California</title>
  <description>November 08, 2005 22:03:46 GMT</description>

  <link>http://earthquake.usgs.gov/recenteqsww/Quakes/ci14197132.htm</link>
  <geo:lat>33.5325</geo:lat>
  <geo:long>-116.6910</geo:long>
  <dc:subject>3</dc:subject>
  <dc:subject>pasthour</dc:subject>
  <dc:subject>11.60 km</dc:subject>
</item>
```

## Adding GeoRSS to the Flash map

```
<script type="text/javascript"

src="http://api.maps.yahoo.com/v3.0/fl/javascript/apiloader.js">
</script>
<div id="mapContainer" style="height: 500px; width:
930px;"></div>
<script type="text/javascript">
var mymap = new Map("mapContainer", "YahooDemo", "San
Francisco, CA", 12);
mymap.addTool(new PanTool(), true);
mymap.addWidget(new NavigatorWidget());

olay = new
GeoRSSOverlay('http://earthquake.usgs.gov/eqcenter/recenteqsww/catalogs/eqs7day-M2.5.xml');
mymap.addOverlay(olay);

</script>
```

### Output:

```
var mymap = new Map("mapContainer", "YahooDemo", "San
Francisco, CA", 12);
mymap.addTool(new PanTool(), true);
mymap.addWidget(new NavigatorWidget());

olay = new
GeoRSSOverlay('http://earthquake.usgs.gov/eqcenter/recenteqsww/catalogs/eqs7day-M2.5.xml');
mymap.addOverlay(olay);
```

Ok, again, disappointing that we didn't need PHP. But to do the same on the DHTML map we need some PHP help. Although there is actually a DHTML GeoRSS overlay now.

## Simple Embedded DHTML map

```
<script type="text/javascript"

src="http://api.maps.yahoo.com/ajaxymap?v=3.0&appid=YahooDemo">
</script>
<div id="jsmapContainer" style="height: 500px; width:
930px;"></div>
<?php
```

```

include 'simple_rss.php';
$url =
'http://earthquake.usgs.gov/eqcenter/recenteqsww/catalogs/eqs7day-M2.5.xml';
$feed = rss_request($url, $timeout=3600);
?>
<script type="text/javascript">
var marker = new Array();
var jsmapi = new
YMap(document.getElementById('jsmapContainer'));
jsmap.addPanControl();
jsmap.addZoomLong();
jsmap.drawZoomAndCenter("94089", 12);
<?php
$i = 0;
while(!empty($feed[$i])) {
    $info = $feed[$i]['title'][0]."<br />";
    $mag = str_replace('.', '', substr($info, 2, 3));
    $info .= $feed[$i]['description'][0]."<br />";
    $info .= '<a href="'. $feed[$i]['link'][0].'">more info</a>';
    $info = addslashes($info);
    $lat = $feed[$i]['lat'][0];
    $lon = $feed[$i]['long'][0];
    echo <<<EOB
    marker[$i] = new YMarker(new YGeoPoint($lat,$lon));
    marker[$i].addLabel("$mag");
    YEvent.Capture(marker[$i],
                    EventsList.MouseClick,
                    new
Function("marker[$i].openSmartWindow('$info');"));
    jsmapi.addOverlay(marker[$i]);
EOB;
    flush();
    $i++;
    if($i>99) break;
}
?>
</script>

```

## yajax.php

```

<?php
if(!empty($_GET['loc'])) {
    $src =
"http://api.local.yahoo.com/MapsService/V1/mapImage?appid=rlerdorf";
    $src.= "&location=".urlencode($_GET['loc']).

"&output=php&image_width=300&image_height=300&zoom=7";
    header("Content-type: application/x-json");
    echo json_encode(unserialize(file_get_contents($src)));
    exit;
}
?>
<html><head>
    <script language="javascript" src="/yui/YAHOO.js"></script>
    <script language="javascript"
src="/yui/connection.js"></script>
    <script language="javascript">
<!--
var fN = function callBack(o) {
    var resp = eval('(' + o.responseText + ')');
    img = document.createElement('img');
    img.src = resp.Result; img.width=300; img.height=300;
img.border=1;
    document.body.appendChild(img);
}
var callback = { success:fN }
function sendform(target,formName) {
    YAHOO.util.Connect.setForm(formName);
    YAHOO.util.Connect.asyncRequest('GET',target,callback);
}
// -->
</script>
</head><body>
    <form name="main"
action="javascript:sendform('yajax.php','main')">
Location: <input type="text" name="loc" />
    </form>
</body></html>

```

## yajax\_geocode.php

```

<?php
if(!empty($_REQUEST['loc'])) {
    $src =
'http://api.local.yahoo.com/MapsService/V1/geocode?appid=rlerdorf'.

'&location='.urlencode($_REQUEST['loc']).'&output=php';
    header("Content-type: application/x-json");
    echo json_encode(unserialize(file_get_contents($src)));
    exit;
}
?>
<html><head>
    <script language="javascript" src="/yui/YAHOO.js"></script>
    <script language="javascript"
src="/yui/connection.js"></script>
    <script language="javascript">
<!--
var fN = function callBack(o) {
    var resp = eval('(' + o.responseText + ')');
    var latlon = document.getElementById('latlon');
    latlon.innerHTML = "Latitude:
<b>"+resp.ResultSet.Result.Latitude+"</b> &nbsp; "+
        "Longitude:
<b>"+resp.ResultSet.Result.Longitude+"</b><br />"+
        "Precision:
<b>"+resp.ResultSet.Result.precision+"</b>";

```

```

}
var callback = { success:fN }
function sendform(target,formName) {
    YAHOO.util.Connect.setForm(formName);
    YAHOO.util.Connect.asyncRequest('POST',target,callback);
}
// -->
</script>
</head><body>
    <form name="main"
action="javascript:sendform('yajax_geocode.php','main')">
Location: <input type="text" name="loc" />
    </form>
    <div id="latlon">
    </div>
</body></html>

```

## yajax\_combined.php

```

<?php
if(!empty($_GET['loc'])) {
    $geo =
'http://api.local.yahoo.com/MapsService/V1/geocode?appid=YahooDemo'.
    '&location='.urlencode($_GET['loc']).'&output=php';
    $ll = unserialize(file_get_contents($geo));
    $lat = $ll['ResultSet']['Result']['Latitude'];
    $lon = $ll['ResultSet']['Result']['Longitude'];
    $map =
"http://api.local.yahoo.com/MapsService/V1/mapImage?appid=YahooDemo".

"&output=php&image_width=450&image_height=450&radius=1500";
    $N =
unserialize(file_get_contents("$map&latitude=$lat&longitude=$lon"));

    $lat = -$lat; $lon = $lon<0 ? 180+$lon : $lon-180;
    $S =
unserialize(file_get_contents("$map&latitude=$lat&longitude=$lon"));

    header("Content-type: application/x-json");
    echo
json_encode(array('north'=>$N['Result'],'south'=>$S['Result'],'lon'=>$lon));
    exit;
}
?>
<html><head>
    <script language="javascript" src="/yui/YAHOO.js"></script>
    <script language="javascript"
src="/yui/connection.js"></script>
    <script language="javascript">
<!--
var img1 = false;
var img2 = false;
var fN = function callBack(o) {
    var resp = eval('(' + o.responseText + ')');
    if(!img1) {
        img1 = document.createElement('img');
        img1.src = resp.north; img1.width=450; img2.height=450;
img2.border=1;
        document.body.appendChild(img1);
    } else img1.src = resp.north;
    if(!img2) {
        img2 = document.createElement('img');
        img2.src = resp.south; img2.width=450; img2.height=450;
img2.border=1;
        document.body.appendChild(img2);
    } else img2.src = resp.south;
}
var callback = { success:fN }
function sendform(target,formName) {
    YAHOO.util.Connect.setForm(formName);
    YAHOO.util.Connect.asyncRequest('GET',target,callback);
}

```

```
// -->
</script>
</head><body>
  <form name="main"
action="javascript:sendform('yajax_combined.php','main')">
Location: <input type="text" name="loc" />
  </form>
  <div id="result"></div>
</body></html>
```

<http://developer.yahoo.com/yui/>

## Connection

```
YAHOO.util.Connect.setForm(formName);
YAHOO.util.Connect.asyncRequest('GET', target, callback);
```

## Dom

```
var pos = YAHOO.util.Dom.getXY('test');
YAHOO.util.Dom.setXY('target', pos);
YAHOO.util.Dom.setStyle('test', 'opacity', 0.5);
YAHOO.util.Dom.getClientWidth(),
YAHOO.util.Dom.getClientHeight()

// Test if cursor is within 'test' region
var targetRegion = YAHOO.util.Dom.getRegion('test');
var cursorPoint = new
YAHOO.util.Point(YAHOO.util.Event.getPageX(e),

YAHOO.util.Event.getPageY(e));
var inRegion = targetRegion.contains(cursorPoint);
```

## Event

```
function fnCallback(e) { alert("click"); }
YAHOO.util.Event.addListener("elementid", "click",
fnCallback);
```

## Animation

```
var myAnim = new YAHOO.util.Anim('test', { height: {to: 10}
});
myAnim.animate();

var myAnim = new YAHOO.util.Motion('test', {points: { to:[10,
10] } });
myAnim.animate();
```

## Drag and Drop

```
var dd1 = new YAHOO.util.DD("elementid1");
var dd2 = new YAHOO.util.DDTarget("elementid2");
```

## Calendar

```
<script src = "/yui/YAHOO.js"></script>
<script src = "/yui/dom.js"></script>
<script src = "/yui/event.js"></script>
<script src = "/yui/dragdrop.js"></script>
<script src = "/yui/calendar.js"></script>
<link rel="stylesheet" href="calendar.css" type="text/css">
<div style="height:8em;">
<div id="calContainer" style="height:7em; float:left;"></div>
<a href="javascript:alert(cal.getSelectedDates())">Check</a>
</div>
<script>
dd1 = new YAHOO.util.DD("calContainer");
cal = new YAHOO.widget.Calendar("cal", "calContainer");
cal.Options.MULTI_SELECT = true;
cal.render();
</script>
```

## Output:



```
ddl = new YAHOO.util.DD("calContainer");  
cal = new YAHOO.widget.Calendar("cal","calContainer");  
cal.Options.MULTI_SELECT = true;  
cal.render();
```

## Sliders

Slides: <http://talks.php.net/show/apeu06>

APC: <http://pecl.php.net/package/APC>

YDN: <http://developer.yahoo.com/>

# Index

Rich Web Apps .....	2
Rich Web Apps .....	3
How to Get Rich .....	4
Initial Benchmark .....	5
Results .....	7
Why is it so slow? .....	8
Let's try MySQL 4.1 .....	10
APC .....	11
APC .....	14
PHP 5 .....	16
XML .....	17
5-second RSS parser .....	20
Atom parser .....	21
Web Services .....	25
SOAP Server .....	26
No more SOAP! .....	28
Geocoding API .....	30
Map Tiles API .....	32
Embedded Map API .....	34
Adding Markers .....	35
Ajax in 30 seconds .....	37
YUI Javascript Libs .....	40
Resources .....	42