

Welcome!

# eZ components

Welcome!

webtech2006 - Sofia, Bulgaria

Derick Rethans - dr@ez.no

<http://files.derickrethans.nl/ezc-webtech2006.pdf>

# eZ components Goals

- Provide a solid platform for PHP application development
- Clean and simple API
- Keep backward compatibility for longer periods of time
- Stable and few regressions
- Clean IP, Open Source friendly

# Development Goals

- Move to PHP 5.1.1
- Thoroughly plan the product to get a top notch API for the library
- Thoroughly tested. All code should be covered by unit tests prior to implementation (PHPUnit)
- Make sure we keep the product open enough for future development
- Do proper documentation during the development

## Documentation:

- Makes a library usable
- API documentation
- Tutorials
- Example Applications
- <http://ez.no/doc/components/overview>

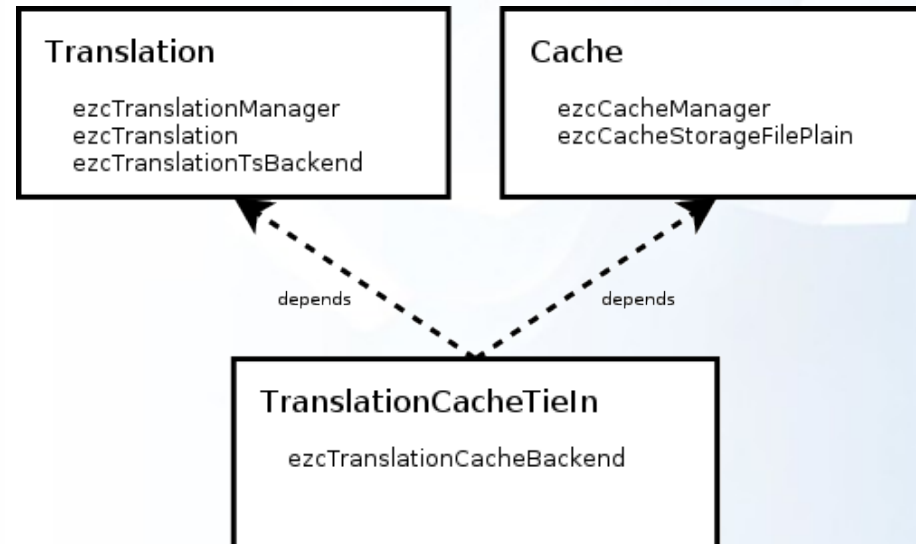
## PHP Documentor:

- Made for PHP
- In use by PEAR
- Supported by PHP IDEs

# Dependencies

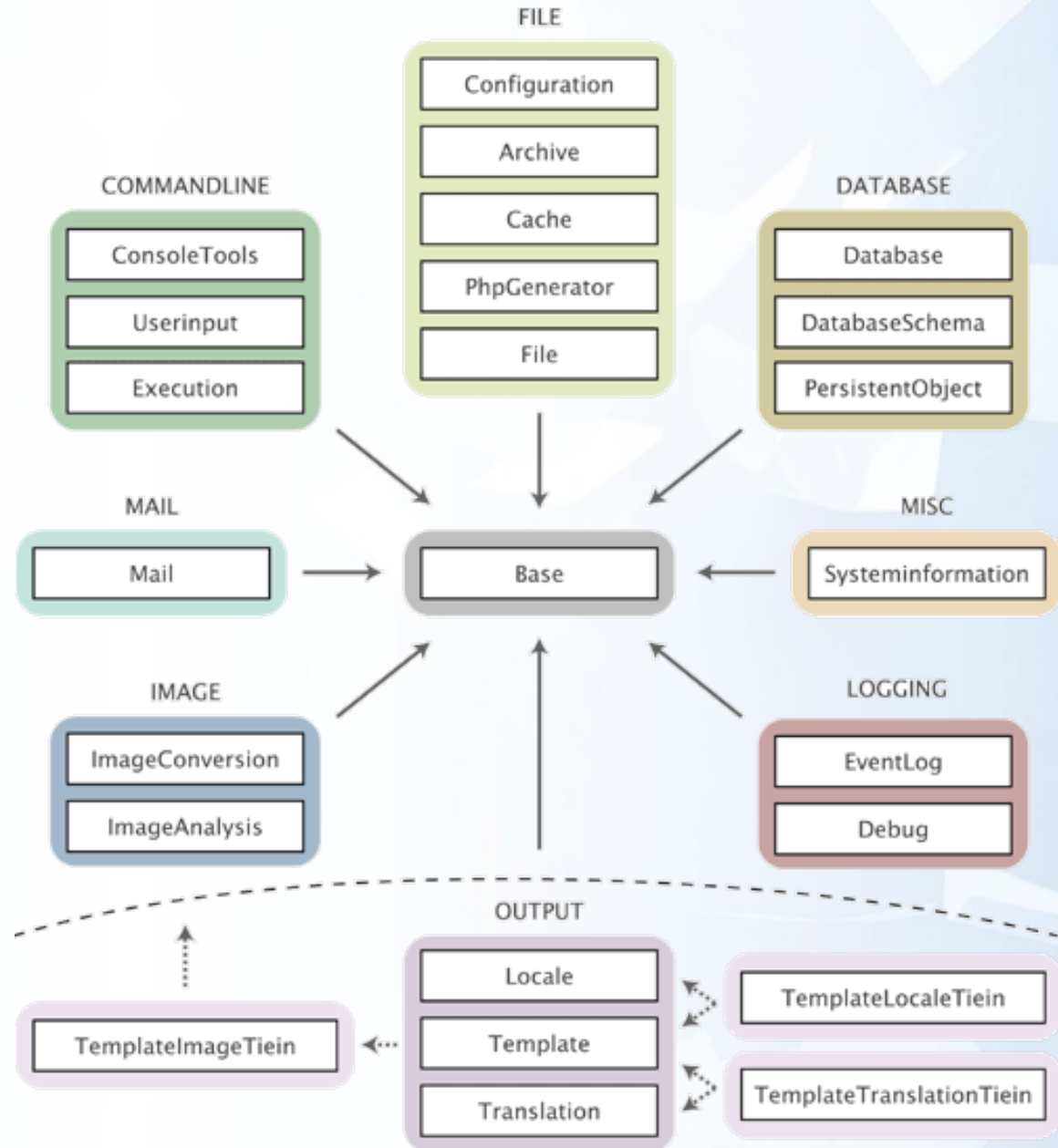
- The less the better
- Only if really necessary
- Dependency-only packages

Tie-Ins:



# Overview

Share your information



## Installation:

```
// download the bundle from http://ez.no/download/ez\_components  
tar -xjf ezcomponents-1.1.tar.bz2  
pwd
```

## Setup:

```
<?php  
ini_set( 'include_path', '/home/httpd/ezcomponents-1.1:.' );  
require 'Base/src/base.php';  
  
function __autoload( $className )  
{  
    ezcBase::autoload( $className );  
}  
  
ezcBase::checkDependency( 'Test', ezcBase::DEP_PHP_VERSION, '5.3.0' );  
?>
```

## Installation:

```
pear channel-discover components.ez.no
pear remote-list -c ezc
pear install ezc/ezcomponents
```

## Setup:

```
<?php
require 'ezc/Base/base.php';

function __autoload( $className )
{
    ezcBase::autoload( $className );
}

ezcBase::checkDependency( 'Test', ezcBase::DEP_PHP_VERSION, '5.3.0' );
?>
```



```

<?php
require 'ezc-setup.php';
$out = new ezcConsoleOutput;
$out->formats->headline->color = 'red';
$out->formats->headline->style = array( 'bold' );

$table = new ezcConsoleTable( $out, 60 );
$table[0]->format = 'headline';
$table[0]->borderFormat = 'headline';

$table[0][0]->content = 'Headline 1';
$table[0][1]->content = 'Headline 2';
$table[1][0]->content = "The next cell will wrap:";
$table[1][1]->content = <<<END
If there is a lot of text in a specific cell then the
ezcConsoleTable will correctly wrap around.
END;
$table[1][1]->align = ezcConsoleTable::ALIGN_RIGHT;

$table->outputTable();
?>

```

```

+-----+-----+
| Headline 1 | Headline 2 |
+-----+-----+
| The next cell will wrap: | If there is a lot of text in a |
|                           | specific cell then the |
|                           | ezcConsoleTable will correctly |
|                           | wrap around. |
+-----+-----+

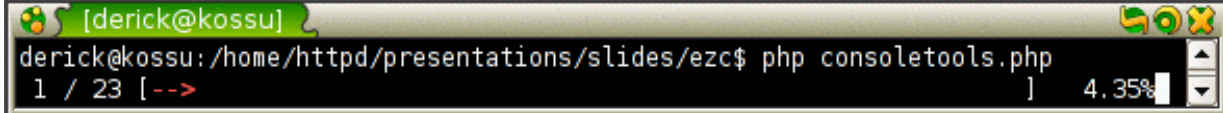
```

```
<?php
require 'ezc-setup.php';

$max = 23;
$output = new ezcConsoleOutput;
$output->formats->bar->color = 'red';
$output->formats->bar->style = array( 'bold' );
$options = array(
    'emptyChar'      => ' ',
    'barChar'        => '-',
    'formatString'   =>
        '%act% / %max% [' .
        $output->formatText( '%bar%', 'bar' ) .
        '] %fraction%%',
);
$progress = new ezcConsoleProgressbar( $output, $max, $options );

for ( $i = 0; $i < $max; $i++ )
{
    usleep( 20000 );
    $progress->advance();
}

$progress->finish();
?>
```



```
[derick@kossu]
derick@kossu: /home/httpd/presentations/slides/ezc$ php consoletools.php
1 / 23 [ --> ] 4.35%
```

## Supported Handlers:

- MySQL
- PostgreSQL
- Oracle
- SQLite

## Using Instances:

```
<?php
require 'ezc-setup.php';

$db1 = ezcDbFactory::create( 'mysql://root@localhost/ezc' );
ezcDbInstance::set( $db1, 'ezc' );

$strunk = ezcDbFactory::create( 'mysql://root@localhost/eztrunk' );
ezcDbInstance::set( $strunk, 'ezt' );

$db = ezcDbInstance::get( 'ezt' );
echo $db->getName();
?>
```

```
<?php
require 'ezc-setup.php';
$db = ezcdbFactory::create( 'mysql://root@localhost/geolocation' );
$sq = $db->createSelectQuery();

$stmt = $sq->select( 'name', 'country', 'lat', 'lon' )
    ->from( 'city' )
    ->where(
        $sq->expr->like(
            'normalized_name', $sq->bindValue( 'sofia%' )
        )
    )
    ->orderBy( 'country', 'name' )
    ->prepare();
$stmt->execute();

foreach ( $stmt as $entry )
{
    list( $name, $country, $lat, $lon ) = $entry;
    printf( '%s, %s is @ %.2f%s %.2f%s<br/>',
        $name, $country,
        abs( $lat ), ( $lat > 0 ? "N" : "S" ),
        abs( $lon ), ( $lon > 0 ? "E" : "W" ) );
}
?>
```

# Persistent Object

```
<?php
require 'ezc-setup.php';
$session = new ezcPersistentSession(
    ezcDbInstance::get(),
    new ezcPersistentCodeManager( "path/to/definitions" )
);

// Creating New Objects
$object = new City();
$object->normalized_name = "dieren";
$object->name = 'Dieren';
$object->country = 'NL';
$session->save( $object );

// Finding Objects
$q = $session->createFindQuery( 'City' );
$q->where(
    $sq->expr->like( 'name', $sq->bindValue( 'oslo%' ) )
)
->orderBy( 'country', 'name' )
->limit( 10 );
$objects = $session->findIterator( $q, 'City' );

foreach ( $objects as $object )
{
    // ...
}
?>
```

```
<?php
require 'ezc-setup.php';
$dir = dirname( __FILE__ );

$mail = new ezcMailComposer();
$mail->from = new ezcMailAddress( 'john@doe.com', 'John Doe' );
$mail->addTo( new ezcMailAddress( 'dr@ez.no', 'Derick Rethans' ) );

$mail->subject = "Example of an HTML email with attachments";
$mail->plainText = "Here is the text version of the mail.
This is displayed if the client can not understand HTML";

$mail->htmlText = <<<ENDHTML
<html>
Here is the HTML version of your mail
with an image: <img src='file://$dir/consoletools-table.png' />
</html>
ENDHTML;

$mail->addAttachment( "$dir/mail.php" );
$mail->build();

echo "<pre><font size='4'>",
    htmlspecialchars(
        $mail->generateHeaders() . "\r\n" . $mail->generateBody() );

$transport = new ezcMailTransportSmtplib( 'localhost', null, null, 2525 );
//$transport->send( $mail );
?>
```

```
<pre><font size="4"><?php
require 'ezc-setup.php';

$mailbox = new ezcMailMboxTransport( dirname( __FILE__ ) . "/mbox-
example.mbox" );
$set = $mailbox->fetchAll();
$parser = new ezcMailParser();
$mail = $parser->parseMail( $set );
showParts( $mail[0], 0 );

function showParts( $mail, $level )
{
    switch ( get_class( $mail ) )
    {
        case 'ezcMail':
            printf( "%s%s\n", str_repeat( ' ', $level ),
                'Mail' );
            showParts( $mail->body, $level + 1);
            break;

        case 'ezcMailText':
            printf( "%s%s (%s, %s)\n", str_repeat( ' ', $level ),
                'Text', $mail->subType, $mail->charset );
            echo "---\n", htmlspecialchars( $mail->text ), "\n---\n";
            break;

        case 'ezcMailFile':
            printf( "%s%s (%s)\n", str_repeat( ' ', $level ),
                'File', $mail->fileName );
            showParts( $mail->file, $level + 1);
            break;
    }
}
```



```
<?php
require 'ezc-setup.php';
list( $domains, $ips ) = require 'data-graph1.php';
$chart = ezcGraph::create( 'Line' );

$chart->title = 'PHP Usage Statistics';
$chart->palette = 'Tango';
$chart->options->padding = 10;
$chart->options->fillLines = 230;

$chart->legend->symbolSize = 7;
$chart->legend->title = "Legend";
$chart->legend->padding = 3;

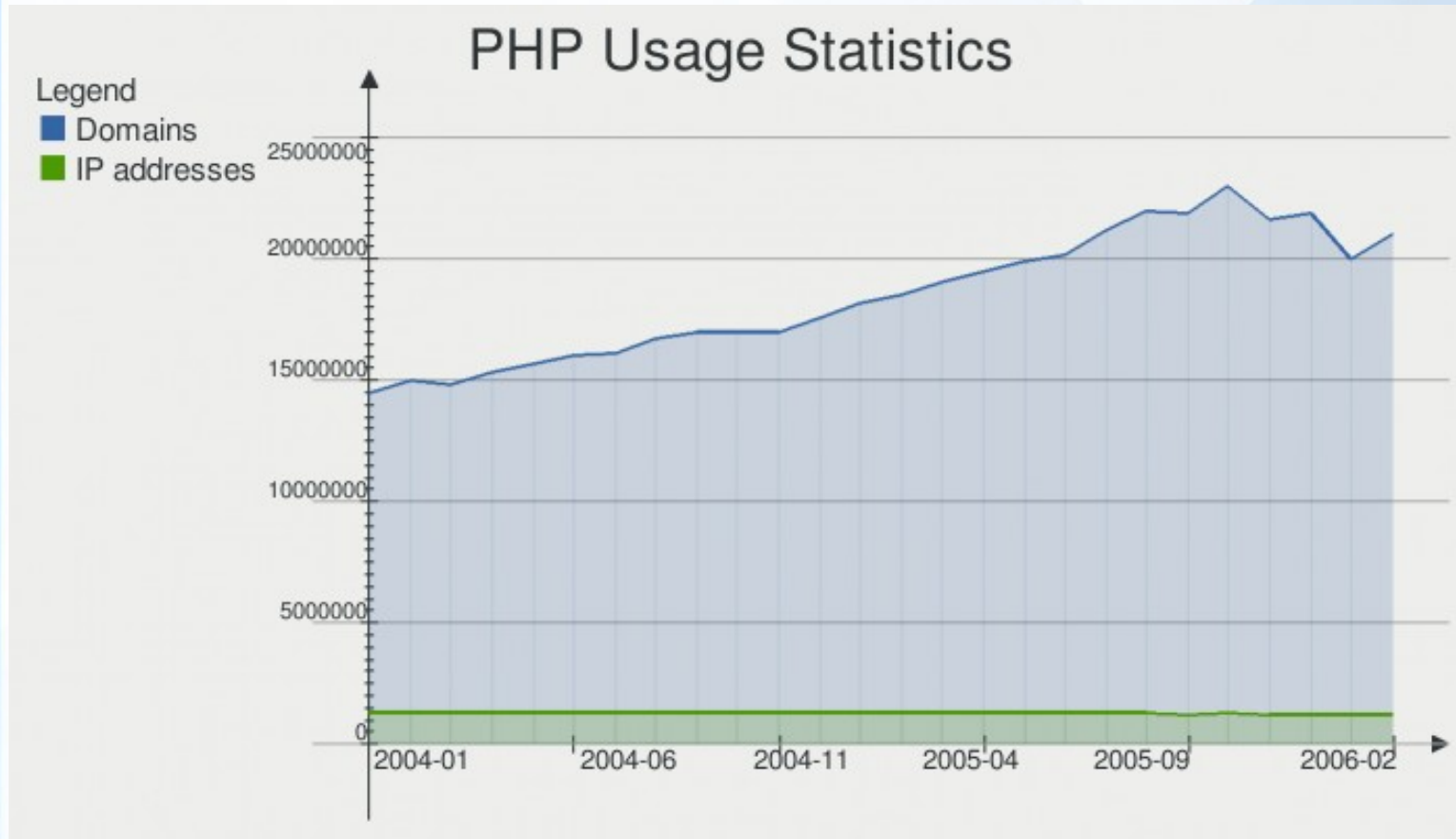
$chart->options->font = dirname(__FILE__) . '/font.ttf';
$chart->xAxis->font->maxFontSize = 10;
$chart->yAxis->grid = '#aaaaaa';
$chart->yAxis->axisSpace = 0.2;
$chart->title->font->maxFontSize = 20;

$chart['domains'] = $domains;
$chart['domains']->label = 'Domains';
$chart['ips'] = $ips;
$chart['ips']->label = 'IP addresses';

$chart->driver = new ezcGraphGdDriver();
$chart->render( 700, 400, '/tmp/graph1.png' );

?>
```





```
<?php
require 'ezc-setup.php';

$chart = ezcGraph::create( 'Pie' );
$chart->title = 'Browser Partitioning';
$chart->palette = 'Tango';

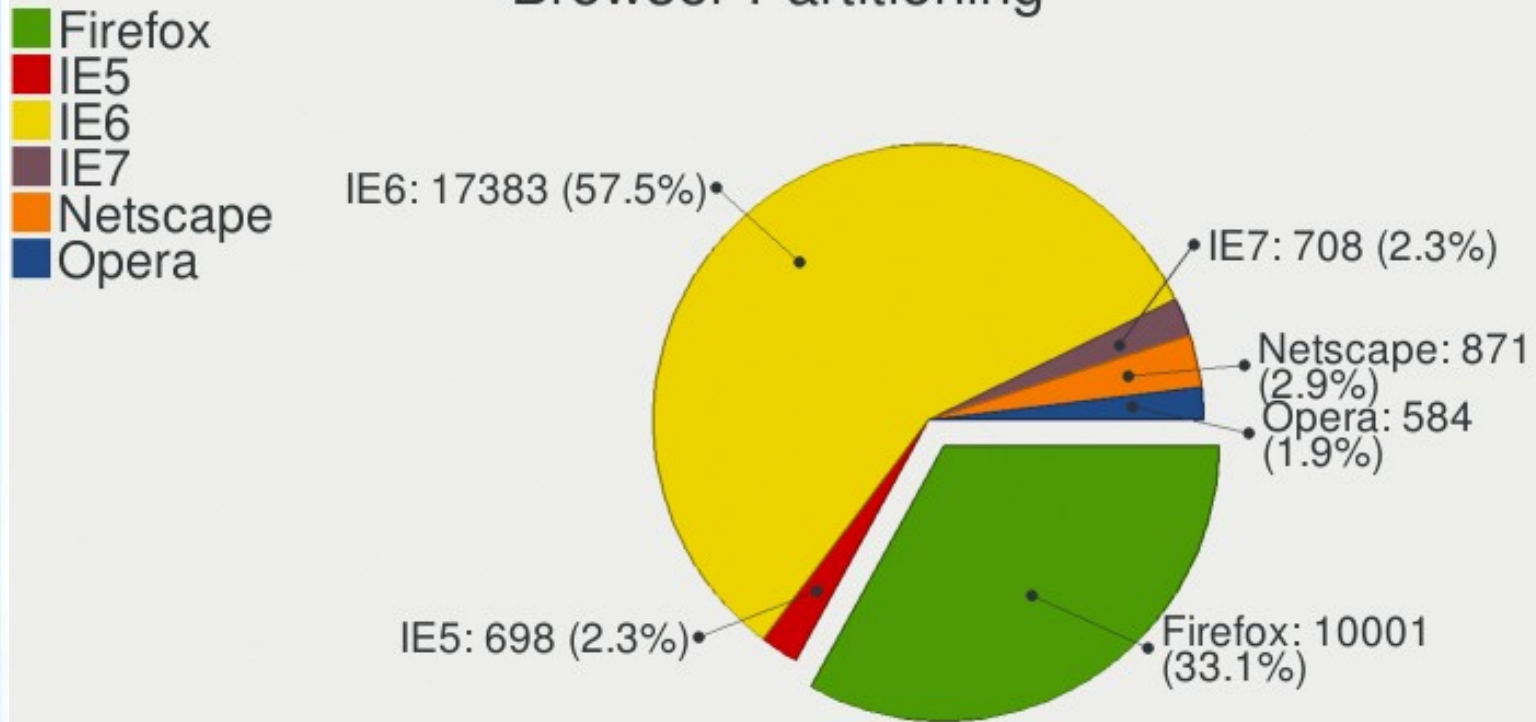
$chart['browser'] = array(
    'Firefox' => 10001,
    'IE5' => 698,
    'IE6' => 17383,
    'IE7' => 708,
    'Netscape' => 871,
    'Opera' => 584,
);
$chart['browser']->highlight['Firefox'] = true;

$chart->title->font = dirname(__FILE__) . '/font.ttf';
$chart->title->font->maxFontSize = 20;
$chart->driver = new ezcGraphGdDriver();
$chart->options->font = dirname(__FILE__) . '/font.ttf';
$chart->options->maxLabelHeight = 0.08;
$chart->render( 700, 400, '/tmp/graph2.png' );
?>
```

# Graph

Share your information

## Browser Partitioning



```
<?php
require 'ezc-setup.php';

// setup feed and content module
$feed = ezcFeed::create( 'rss2' );
$feed->title = 'eZ components release feed';
$feed->link = 'http://components.ez.no/';
$feed->description = <<<ENDL
This feed shows all the latest components releases.
ENDL;
$feed->copyright = "eZ systems A.S.";
$feed->language = 'en-us';

// load data
$stmt = require 'feed-data.php';
foreach( $stmt as $release )
{
    $item = $feed->newItem();
    $item->title = "{$release['package']} {$release['version']}";
    $item->link =
"http://ez.no/doc/components/view/latest/(file)/changelog_{$release['package']}
}.html";
    $item->description = $release['releasenotes'];
    $item->published = $release['releasedate'];
    $item->guid = md5( $item->title );
}

echo htmlspecialchars( $feed->generate() );
?>
```

# What's Next?

## New components:

- ContentDiff
- Feed: RSS/ATOM feeds
- Graph: Graphs and diagrams
- SignalObserver: IPC for PHP applications
- URL: Obtaining parameters from URLs and creating them

## Additions to components:

- Database: Oracle support
- DatabaseSchema: Support for SQLite, PostgreSQL and Oracle
- Mail: More authentication mechanisms
- PersistentObject: Class relations
- Template: Custom functions and character set support

# Questions and Resources

Questions anybody?

Resources:

- Download: [http://ez.no/download/ez\\_components](http://ez.no/download/ez_components)
- Documentation: <http://ez.no/doc/components/overview>
- Mailinglist: <http://lists.ez.no/mailman/listinfo/components>
- Forum: [http://ez.no/community/forum/ez\\_components](http://ez.no/community/forum/ez_components)
- These Slides: <http://files.derickrethans.nl/ezc-webtech2006.pdf>