

Welcome!

# eZ Components

Welcome!

**Dutch PHP Conference - Amsterdam, Netherlands**

**Derick Rethans - dr@ez.no**

**<http://derickrethans.nl/talks.php>**

- Provide a solid platform for PHP application development
- Don't force a structure: no "framework"
- Clean and simple API
- Excellent documentation
- Keep backward compatibility for longer periods of time
- Stable and few regressions
- Clean IP, Open Source friendly

- New BSD license: Open Source, very permissive, compatible with GPL
- CLA: To accept contributions to the components we need a signed "Contributor Licensing Agreement"

- Pre-fixed: for namespacing
- Readable: for sanity
- Slightly Mangled: for clarity

ezcMail vs. Mail

ezcTestSuite vs. PHPUnit2\_Framework\_TestSuite

ezcMailSmtptTransport vs. ezcMailSMTPTransport

```
<?php
require 'ezc/Base/base.php';
function __autoload( $className )
{
    ezcBase::autoload( $className );
}

$a = new ezcMailSmtTransport( 'localhost' );
var_dump( $a );
?>
```

Using the classname's part as path elements makes ugly paths:

```
ezcMailParserSet      => Mail/parser/set.php,  
ezcMailSmtptTransport => Mail/smtpt/transport.php,  
ezcMailTransportOptions => Mail/transport/options.php,  
ezcMailAddress        => Mail/mail/address.php,
```

More logical names (mail\_autoload.php):

```
ezcMailParserSet      => Mail/parser/interfaces/parser_set.php,  
ezcMailSmtptTransport => Mail/transports/smtpt/transport_smtpt.php,  
ezcMailTransportOptions => Mail/options/transport_options.php,  
ezcMailAddress        => Mail/structs/mail_address.php,
```

Some problems:

- Clashes in first part of the classname
- Needs installation into correct place for development

```
<div id="graph"> <iframe id="graphIn" src="http://kossu/test/  
allyourbase.svg"/> </div>
```

## 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 );
}
?>
```



- Sits on top of Database
- Object Relational Model
- Based on Hibernate - not Active Record
- Uses definition files, which can be auto generated by a script as part of the  
PersistentObjectDatabaseSchemaTiein:

```
php rungenerator.php  
-s mysql://root@localhost/geolocation -f mysql /tmp
```

- The classes itself can also be generated by this script
- Work is being done to allow definitions to be generated on the fly from annotated source code

- Support for multiple backends
- Abstract documents
- Support for datatypes
- Rich searching API, including faceted search
- Easy query interface

```
<?xml version="1.0"?>
<document>
  <field type="id">id</field>
  <field type="text" boost="2">title</field>
  <field type="text">summary</field>
  <field inResult="false" type="html">body</field>
  <field type="date">published</field>
  <field type="string" multi="true">author</field>
</document>
```

## Setting up the manager:

```
<?php
$backend = new ezcSearchSolrHandler;

$session = new ezcSearchSession(
    $backend,
    new ezcSearchXmlManager( $testFilesDir )
);
?>
```

```
<?php
$session = new ezcSearchSession(
    $backend,
    new ezcSearchXmlManager( $testFilesDir )
);

$a = new Article(
    null, // id
    'Test Article', // title
    'This is an article to test', // description
    'the body of the article', // body
    time() // published
);
$session->index( $a );
?>
```

```
<?php
$session = new ezcSearchSession(
    $backend,
    new ezcSearchXmlManager( $testFilesDir )
);

$q = $session->createFindQuery( 'Article' );
$q->where( $q->eq( 'title', 'Article' ) );
->limit( 5 );
->orderBy( 'id' );

$r = $session->find( $q );

foreach ( $r->documents as $document )
{
    echo $document['document']->title, "\n";
}
?>
```

```
<?php
$session = new ezcSearchSession(
    $backend,
    new ezcSearchXmlManager( $testFilesDir )
);

$q = $session->createFindQuery( 'Article' );

new ezcSearchQueryBuilder(
    $q,
    'thunderball',
    array( 'fieldOne', 'fieldTwo' )
);

$q->facet( 'title' ); // keyword data field

$r = $session->find( $q );
foreach ( $r->documents as $document )
{
    echo $document['document']->title, "\n";
}
?>
```

```
<?php
require 'ezc-setup.php';
$mail = new ezcMailComposer();

// from and to addresses, and subject
$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";

// body: plain
$mail->plainText = "Here is the text version of the mail.";

// body: 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;

// add an attachment
$mail->addAttachment( "$dir/mail.php" );

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

# Mail

## Retrieving and Parsing Mail

```
<pre><font size="4"><?php
require 'ezc-setup.php';
require 'mail-parse2b.php';

$set = new ezcMailFileSet( array( dirname( __FILE__ ) . "/mail-example.mail" ) );
$parser = new ezcMailParser();
$mail = $parser->parseMail( $set );
echo htmlspecialchars( formatMail( $mail[0] ) );

?>
```



Sending with:

- SMTP
- "MTA" (PHP's mail())

Reading from:

- POP3, IMAP, MBOX, Files, Variables

Supported formats:

- multipart/alternative: text/html mail
- multipart/mixed: attachments
- multipart/related: HTML mail with inline images
- multipart/digest: mailinglist digests

Overview:

Creates different kinds of diagrams:

- Line charts
- Bar charts
- Pie charts
- Radar charts

Drivers: GD, SVG, Flash

Renderers: 2D, 3D

Datasets: Array, Numeric, PDO, Average (Polynomial)

Axis: Date, Numeric, labeled, Logarithmic

Palettes: Black, eZ, Tango, User Defined

```
<?php
require 'ezc-setup.php';
header( 'Content-Type: image/svg+xml' );
list( $domains, $ips ) = require 'data-graph1.php';

$chart = new ezcGraphLineChart();

$chart->title = 'PHP Usage Statistics';
$chart->palette = new ezcGraphPaletteTango();
$chart->options->fillLines = 230;

$chart->legend->title = "Legend";

$chart->xAxis->font->maxFontSize = 12;
$chart->yAxis->font->maxFontSize = 12;
$chart->title->font->maxFontSize = 20;

$chart->data['domains'] = new ezcGraphArrayDataSet( $domains );
$chart->data['domains']->label = 'Domains';
$chart->data['ips'] = new ezcGraphArrayDataSet( $ips );
$chart->data['ips']->label = 'IP addresses';

$chart->driver = new ezcGraphSvgDriver();
$chart->render( 600, 400, 'php://output' );

?>
```

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

$chart = new ezcGraphPieChart();
$chart->palette = new ezcGraphPaletteEzBlue();

$chart->title = 'Browser Partitioning';

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

$chart->driver = new ezcGraphSvgDriver();

$chart->renderer = new ezcGraphRenderer3d();
$chart->renderer->options->pieChartShadowSize = 12;
$chart->renderer->options->pieChartGleam = .5;
$chart->renderer->options->dataBorder = false;
$chart->renderer->options->pieChartHeight = 16;
$chart->renderer->options->legendSymbolGleam = .5;

$chart->renderer->options->pieChartOffset = 180;

header( 'Content-Type: image/svg+xml' );
$chart->render( 900, 500, 'php://output' );
?>
```

```
<div id="graph"> <iframe id="graphIn"  
src="/presentations/slides/ezc/graph2.php"/> </div>
```

The Graph component outputs:

- PNG/JPG, through the `ezcGraphGdDriver()`
- SVG, through the `ezcGraphSvgDriver()`
- Flash, through the `ezcGraphFlashDriver()`

### New components:

- Document: Document conversion tool
- Feed: RDF/RSS/ATOM feed parsing and generating, with support for modules
- Search: Search API towards Solr, Sphinx, Google, etc.

### Additions to components:

- Cache: Hierarchical caching
- Template: Translation and character set support

Roadmap: <http://issues.ez.no/RoadMap.php?Id=729>

This is a tentative roadmap

New components:

- MvcTools: Light weight MVC layer with Tieins

Additions to components:

- Base: Different development modes
- Document: Support for many more formats
- Search: More backends

Questions anybody?

## Resources:

- Download: <http://ezcomponents.org/download>
- Documentation: <http://ezcomponents.org/docs>
- Mailinglist: <http://lists.ez.no/mailman/listinfo/components>
- IRC: #ezcomponents @ Freenode
- These Slides: <http://derickrethans.nl/talks.php>