

Welcome!

eZ Components

Welcome!

OpenWeb Vancouver - Vancouver, Canada

Derick Rethans - dr@ez.no

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

- 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

- 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

Documentation:

- Makes a library usable
- API documentation
- Tutorials
- Example Applications
- <http://ezcomponents.org/docs>

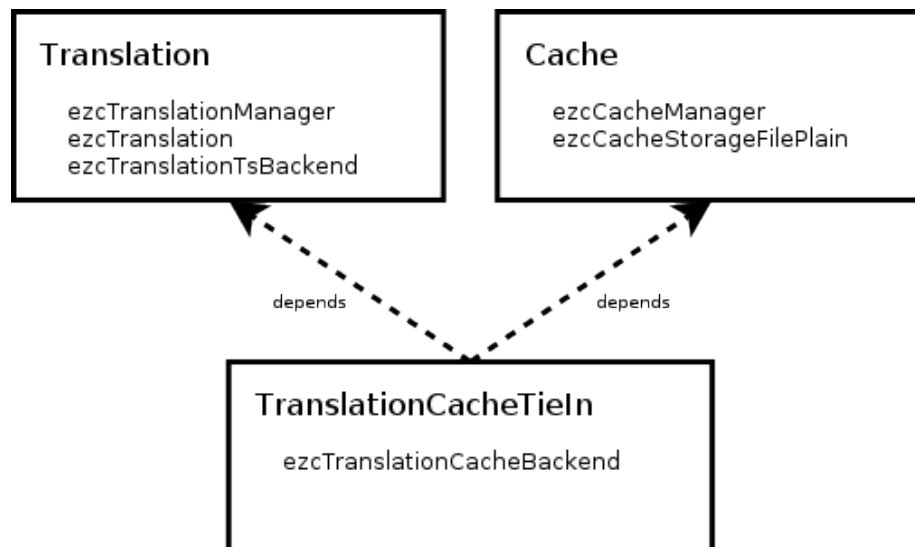
PHP Documentor:

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

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

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

Tie-Ins:



- 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 ezcMailSmtptTransport( '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:

```
// download the bundle from  
http://ezcomponents.org/download  
tar -xjf ezcomponents-2007.2.1.tar.bz2  
pwd
```

Setup:

```
<?php  
ini_set( 'include_path',  
' /home/httpd/ezcomponents-2007.2.1:.' );  
require 'Base/src/base.php';  
  
function __autoload( $className )  
{  
    ezcBase::autoload( $className );  
}  
?>
```

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 );  
}  
?>
```

Supported Handlers:

- MS SQL
- 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/trip' );
ezcDbInstance::set( $strunk, 'ezt' );

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

Allows writing SQL that works for all supported databases, without having to deal with differences in SQL dialects.

Supported SQL categories:

- SELECT: `$db->createSelectQuery();`
- INSERT: `$db->createInsertQuery();`
- UPDATE: `$db->createUpdateQuery();`
- DELETE: `$db->createDeleteQuery();`

Query object:

- Uses a "fluent-interface"
- Exposes an expression object through `$db->e`.

Works by returning the object that a method is called on after the method does its work.

```
<?php
public function from()
{
    if ( $this->fromString == '' )
    {
        $this->fromString = 'FROM ';
    }
    // ...
    $this->fromString .= join( ', ', $tables );
    return $this;
}
?>
```

Allows for chaining of calls:

```
<?php
$q->select( '*' )->from( 'quotes' )
->where( $e->eq( 'author', $q->bindValue( 'Robert
Foster' ) ) );
?>
```

- Safer: No manual quoting required
- Faster: The query is only compiled once

```
<?php
$country = $normalized_name = $name = $province = $lat =
$lon = null;
$sql = $db->createInsertQuery();
$sql->insertInto( 'city' )
->set( 'country', $sql->bindParam( $country ) )
->set( 'normalized_name', $sql-
>bindParam( $normalized_name ) )
->set( 'name', $sql->bindParam( $name ) )
->set( 'province', $sql->bindParam( $province ) )
->set( 'lat', $sql->bindParam( $lat ) )
->set( 'lon', $sql->bindParam( $lon ) );
$stmt = $sql->prepare();

do {
    $line = fgets( $fd ); if ( feof( $fd ) ) break;
```

- 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

- Provide an interface to Search backends
- Interfaces to: Solr (Lucene)

```
<?php
$solr = new ezcSearchSolrHandler;
$session = new ezcSearchSession( $solr, new
ezcSearchEmbeddedManager() );
$q = $session->createFindQuery( 'ezcSearchSimpleArticle'
);
$searchWords = split( ' ', $searchWord );
foreach( $searchWords as $searchWord )
{
    $q->where( $q->eq( 'body', $searchWord ) )
    ->where( $q->eq( 'title', $searchWord ) );
}
$q->facet( 'type' )
->limit( 10, $begin );
$r = $session->find( $q );
?>
```

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

$archive = ezcArchive::open( dirname( __FILE__ ) .
'/test-archive.zip' );
foreach( $archive as $entry )
{
    if ( $entry->getPath() == 'archive-logo.jpg' )
    {
        $dir = dirname( __FILE__ );
        echo "Extracting {$entry->getPath()} to
$dir.<br/><br/>\n";
        $archive->extract( $dir );
    }

    echo $entry, "\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 Creation Diagram

```
<div id="graph"> <iframe id="graphIn"  
src="/presentations/slides/ezc/mail-diagram.svg"/> </div>
```

```
<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' );

?>
```

```
<div id="graph"> <iframe id="graphIn"  
src="/presentations/slides/ezc/graph1.php"/> </div>
```

```
<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:

- 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>

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>