

Welcome!

eZ components

Vancouver PHP Conference - Vancouver, Canada

Derick Rethans - dr@ez.no

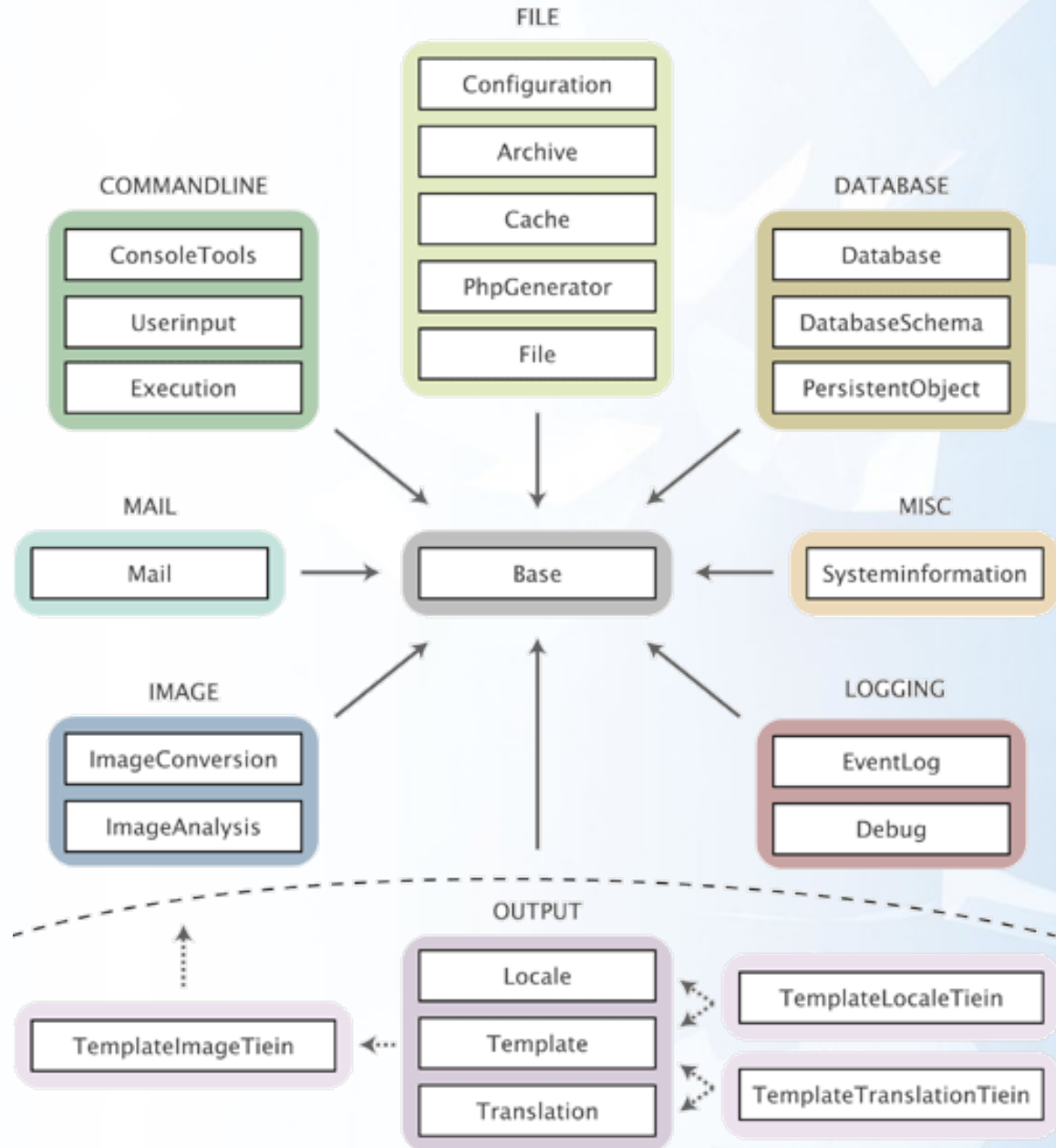
<http://files.derickrethans.nl/ezc-vancouver7.pdf>

eZ components Goals

- 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

Overview

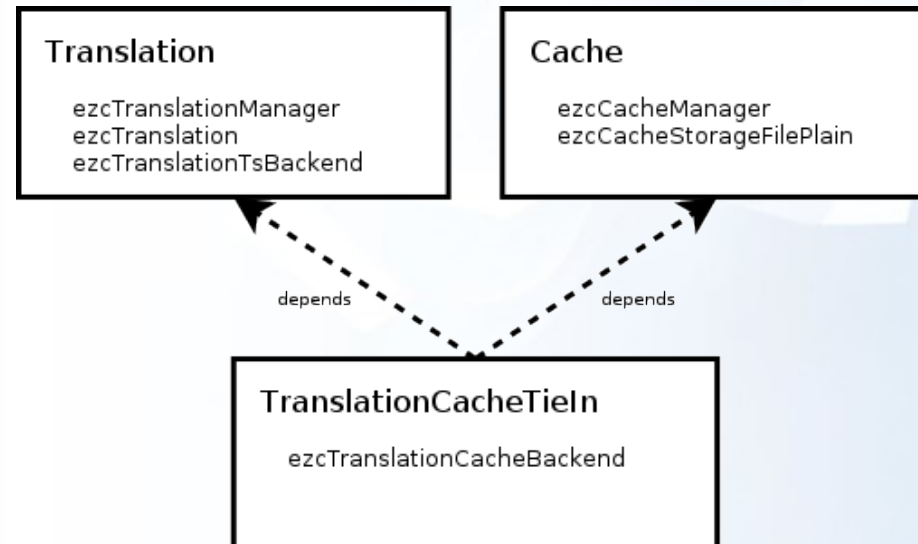
Share your information



Dependencies

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

Tie-Ins:



Installation:

```
// download the bundle from http://ez.no/download/ez\_components  
tar -xjf ezcomponents-2006.2.tar.bz2  
pwd
```

Setup:

```
<?php  
ini_set( 'include_path', '/home/httpd/ezcomponents-2006.2:.' );  
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. |
+-----+-----+

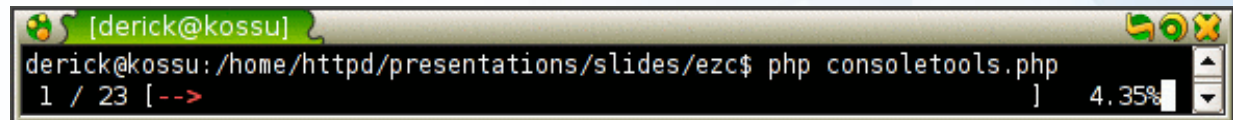
```

```
<?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();
?>
```




```
<?php
require 'ezc-setup.php';
$log = ezcLog::getInstance();
$log->source = $log->category = NULL;

// Create writers
$warningWriter = new ezcLogUnixFileWriter(
    "/tmp", "ezc-pres-warning.log", 128
);
$errorWriter = new ezcLogUnixFileWriter(
    "/tmp", "ezc-pres-error.log", 256
);

// Create filters
$warningFilter = new ezcLogFilter;
$warningFilter->severity = ezcLog::WARNING;
$log->getmapper()->appendRule(
    new ezcLogFilterRule( $warningFilter, $warningWriter, true )
);

$errorFilter = new ezcLogFilter;
$errorFilter->severity = ezcLog::ERROR;
$log->getmapper()->appendRule(
    new ezcLogFilterRule( $errorFilter, $errorWriter, true )
);

// Log messages
$log->log( "Oops, this was unexpected.", ezcLog::WARNING );
$log->log( "Oh no, major problem!", ezcLog::ERROR, array(
    "category" => "SQL" )
);

echo '<font size="6">';
echo 'Warnings<br/>', nl2br( htmlspecialchars( file_get_contents( '/tmp/ezc-
pres-warning.log' ) ) ), "<br/>";
echo 'Errors<br/>', nl2br( htmlspecialchars( file_get_contents( '/tmp/ezc-
pres-error.log' ) ) );
?>
```

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( 'vanco%' )
        )
    )
    ->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" ) );
}
?>
```

```
<?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__ );
// Setup
$filters = array();
$settings = new ezcImageConverterSettings(
array( new ezcImageHandlerSettings( 'GD', 'ezcImageGdHandler' ) )
);
$converter = new ezcImageConverter( $settings );

// Create transformations
$filters[] = new ezcImageFilter( 'scale',
array(
    'width' => 320, 'height' => 240,
    'direction' => ezcImageGeometryFilters::SCALE_DOWN,
)
);
$converter->createTransformation(
'preview', $filters, array( 'image/jpeg' )
);

$filters[] = new ezcImageFilter( 'colorspace',
array( 'space' => ezcImageColorspaceFilters::COLORSPACE_GREY )
);
$converter->createTransformation(
'prevgrey', $filters, array( 'image/jpeg' )
);
?>
```

ImageConversion

```
<?php
require 'imageconversion.php';

// Apply transformations
$converter->transform(
    'preview', "$dir/nacreous.jpg", "$dir/nacreous-small.jpg"
);
$converter->transform(
    'prevgrey', "$dir/nacreous.jpg", "$dir/nacreous-grey.jpg"
);
?>
```

Result:



```
<?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';
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] ) );

?>
```


Parsing from:

- POP3 Servers
- IMAP Servers
- MBOX Files
- Single Mail Files
- Variables

Supported formats:

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

```
<?php
require '../include/ezc-setup.php';
require '../include/template_function_call.php';

$tpl = new ezcTemplate();
$tpl->configuration->addExtension( "TemplateFunctionCall" );
$tpl->send->listing = array( 1 => 'Item 1', 2 => 'Item 2' );

echo $tpl->process( 'oneonly.ezt' );
?>

{use $listing}
<ul>
{foreach $listing as $id => $entry}
<li><a href='index.php?mailNr={$id}' class="nowrap">{$entry}</a></li>
{/foreach}
</ul>
```

- Variable scoping per file
- Output contexts
- Custom extensions: functions, blocks
- Compiled to PHP code

```
<?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;

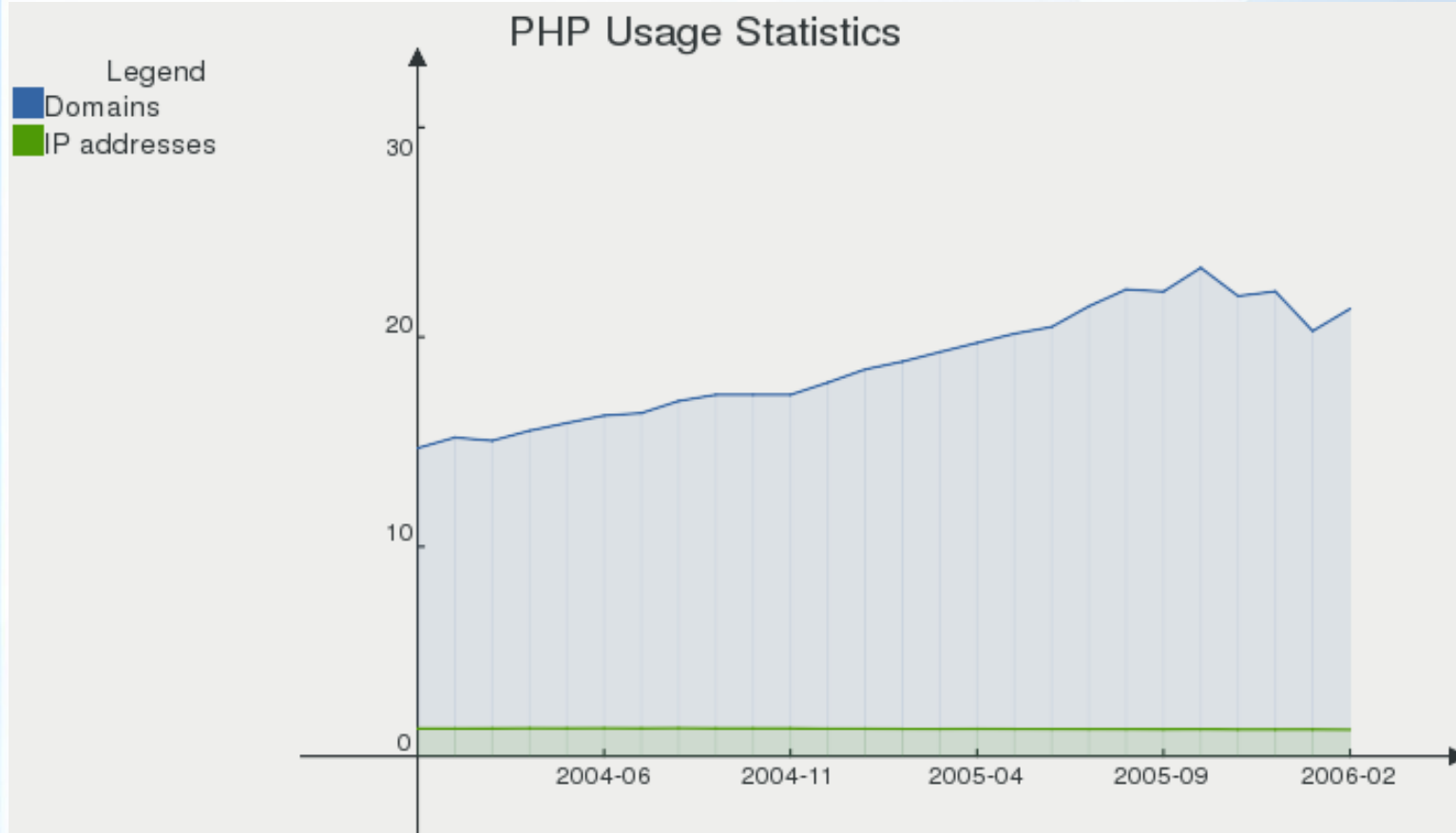
$chart->legend->title = "Legend";

$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( 700, 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;

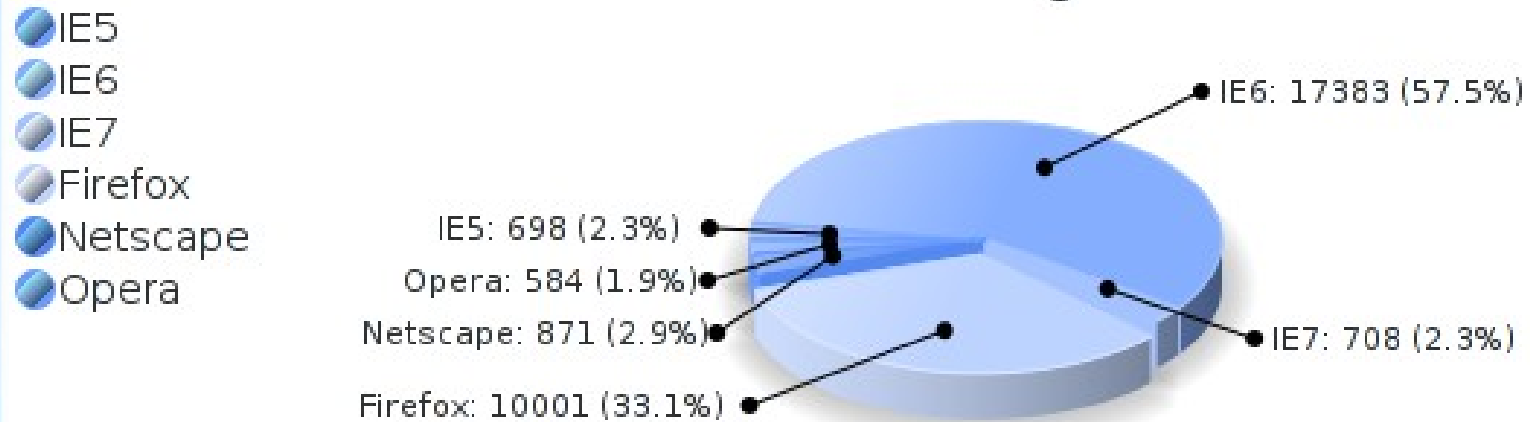
$chart->driver = new ezcGraphSvgDriver();

$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( 560, 200, 'php://output' );
?>
```

Browser Partitioning



The Graph component outputs:

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

```
<pre style="font-size: 11px;">
<?php
require 'ezc-setup.php';

// setup feed and content module
$feed = new ezcFeed( '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() );
?>
```

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

$feed = new ezcFeed( 'rss2' );
try {
    $feed = ezcFeed::parse( 'http://components.ez.no/rss/rss2.xml' );
} catch ( Exception $e ) {
    $feed = ezcFeed::parse( dirname( __FILE__ ) . '/rss2.xml' );
}

echo "<b>{$feed->title}</b><br/><br/>\n";
foreach( $feed->items as $item )
{
    echo "<a href='{$item->link}'>{$item->title}</a><br/>";
    echo $item->Content->encoded, "<br/>";
}
?>
```


What's Next?

New components:

- Authentication: Graphs and diagrams
- Feed: Reading and writing RSS/Atom feeds
- Workflow: Workflow engine

Additions to components:

- Database: SQL Server driver, Debugging support
- DatabaseSchema: Foreign Key support
- Mail: Caching, other additions
- PersistentObject: Class stub generation
- Template: Caching, Character set support and Translation Tie-in

Roadmap: <http://issues.ez.no/RoadMap.php?Id=487>

License and Contributing

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

Questions and Resources

Questions anybody?

Resources:

- Download: <http://ez.no/ezcomponents/download>
- Documentation: <http://ez.no/doc/components/overview>
- Mailinglist: <http://lists.ez.no/mailman/listinfo/components>
- IRC: #ezcomponents @ Freenode
- These Slides: <http://files.derickrethans.nl/ezc-vancouver7.pdf>