

PEAR – Neue Birnen braucht das Land



1 / 26

Willkommen!

PHP-Entwicklung mit PEAR

Stefan Neufeind <neufeind@php.net>

LinuxWorld Conference & Expo
Frankfurt, 2004-06-27

Creative Commons Attribution-ShareAlike License

Agenda



2 / 26

- Vorstellung
- Was ist PEAR?
- Was bietet mir PEAR?
- Standards und Regeln
- PEAR Installer
- PEAR Website
- Paketbeispiele
- Kontakte
- Überblick
- Fragen

Vorstellung



3 / 26

- Stefan Neufeind, aus Neuss (bei Düsseldorf)
- Consultant und Entwickler für SpeedPartner
- Seit einigen Jahren “überzeugter” PHP-Entwickler
- PEAR-Pakete:
 - Image_Graph
 - Net_Traceroute
 - HTTP_Session
 - Validate
- PEAR DEV / QA / DOC

Was ist PEAR?



4 / 26

- "PHP Extension and Application Repository"
- Umfangreiche Sammlung von PHP-Komponenten
 - Objektorientiert
 - Flexibel
 - Modular
 - Hohe Qualität
- Gegründet 1999 von Stig S. Bakken
- Mehr als 350 Pakete, 700 registrierte Entwickler
- Community-Struktur

Was ist PEAR?



5 / 26

- Einheitliche PHP-Paket-Infrastruktur
- Vermeidung von redundantem Code (über Abhängigkeiten)
- Definierte Standards
- Flexible Verwendung der Komponenten

Was bietet mir PEAR?



6 / 26

- Vielzahl fertiger Komponenten
- Einfache Installation und Verwaltung
- Installer für PEAR-Pakete, eigene Pakete und “3rd party”-Pakete (z.B. phpOpenTracker, Horde-Framework)
- “Framework” für weitere Entwicklungen
- Einheitliche Basis (z.B. error-handling)

Standards und Regeln



7 / 26

- Einheitlicher Coding-Style
- Definierte Paket-Stati (devel, alpha, beta, stable)
- Standard für Versionsionierung
- Dokumentations-Standard (PHPDoc)
- Standard für Unit-Tests (PHP_Unit)

Standards und Regeln



8 / 26

Wofür?

- Strukturierte Entwicklung
- Qualität, Wartbarkeit
- Einfachere Handhabung
- Flexibles und “generisches” API-Design gefördert

- Installiert PHP- und C-Pakete
- Kommunikation per XML_RPC
- Mitgeliefert seit PHP 4.3.0
- Auch einfache Nach-Installation:
 - <http://go-pear.org/>
 - Linux: `lynx -source http://go-pear.org/ | php -q`
 - Browser-Installation möglich
- Oberflächen für Konsole, Web, GTK

PEAR Installer



10 / 26

Live-Demo

- Neue Version in Arbeit:
 - Channel-Support
 - Erweiterte Paket-Mechanismen
 - Automatische Abhängigkeitsauflösung
- Pläne:
 - Datenbank-Unterstützung
 - Webroot-Installation
 - Paket-Signaturen
 - Weitere, alternative Frontends

- Katalogisiertes Paket-Archiv
- Programmierer-Kontakte
- Bugtracker
- Dokumentation
- Support-Links
(Mailinglisten, Chat, Tutorials, Präsentationen)
- Installer-Backend (XML_RPC)
- PEAR Proposal-System (PEPr)
- Entwickler-Funktionen

PEAR Website



13 / 26

Live-Demo

Paketbeispiel: Mail_Mime



14 / 26

```
<?php
include('Mail.php');
include('Mail/mime.php');

$text = 'Text version of email';
$html = '<html><body>HTML version of email</body></html>';
$file = '/home/richard/example.php';
$CrLf = "\n";
$headers = array('From' => 'you@yourdomain.com',
                 'Subject' => 'Test mime message' );

$mime = new Mail_mime($CrLf);
$mime->setTXTBody($text);
$mime->setHTMLBody($html);
$mime->addAttachment($file, 'text/plain');

$body = $mime->get();
$headers = $mime->headers($headers);
$mail =& Mail::factory('mail');

$mail->send('postmaster@localhost', $headers, $body);
?>
```

Paketbeispiel: MDB2



15 / 26

```
/* Import eines XML-Schema */

MDB2::loadFile('Tools/Manager');
$manager =& new MDB2_Tools_Manager;
$input_file = 'metapear_test_db.schema';
$manager->connect($db);
$manager->updateDatabase($input_file, $input_file.'.before');

/* Einfache Selects */
$db->setDatabase($db_name);
$query = 'SELECT * FROM test';
$result = $db->query($query);
$array = $result->fetchRow();
$result->free();

$array = $db->queryAll($query);
```

Paketbeispiel: MDB2



16 / 26

```
/* Prepared statements */
$alldata = array(
    array(1, 'one', 'un'),
    array(2, 'two', 'deux'),
    array(3, 'three', 'trois'),
    array(4, 'four', 'quatre')
);

$prepared_query = $db->prepare('INSERT INTO numbers VALUES(?,?,?)',
    array('integer', 'text', 'text'));

foreach ($alldata as $row) {
    echo('running execute<br>');
    $db->executeParams($prepared_query, null, $row);
}

/* Subselects */
$sub_select = $db->subSelect('SELECT test_name from test WHERE test_name = '.
    $db->quote('gummihuhn', 'text'), 'text');
$query_with_subselect = 'SELECT * FROM test WHERE test_name IN '.
    '('.$sub_select.')';
$result = $db->query($query_with_subselect);
$array = $result->fetchAll();
$result->free();
```


Paketbeispiel: HTML_QuickForm



17 / 26

```
require_once 'HTML/QuickForm.php';
$form = new HTML_QuickForm('firstForm');

$form->setDefaults(array('name' => 'Joe User'));

$form->addElement('header', null, 'QuickForm tutorial example');
$form->addElement('text', 'name', 'Enter your name:',
    array('size' => 50, 'maxlength' => 255));
$form->addElement('submit', null, 'Send');

$form->applyFilter('name', 'trim');
$form->addRule('name', 'Please enter your name', 'required',
    null, 'client');

if ($form->validate()) {
    echo '<h1>Hello, ' . htmlspecialchars($form->exportValue('name')).
        '!</h1>';
    exit;
}

$form->display();
```

Paketbeispiel: HTML_QuickForm



18 / 26

QuickForm tutorial example

*Enter your name:

* denotes required field

Paketbeispiel: Image_Graph (ab 0.3.0dev1)



19 / 26

```
$Graph =& Image_Graph::factory('graph', array(800, 600));

$Arial =& $Graph->addNew('ttf_font', 'arial.ttf');
$Arial->setSize(18);
$Arial1 =& $Graph->addNew('ttf_font', 'arial.ttf');
$Arial1->setSize(14);
$Arial1->setAngle(90);
$Arial2 =& $Graph->addNew('ttf_font', 'arial.ttf');
$Arial2->setSize(14);
$Arial2->setAngle(270);
$Arial3 =& $Graph->addNew('ttf_font', 'arial.ttf');
$Arial3->setSize(12);

$Graph->add(
    Image_Graph::vertical(
        Image_Graph::factory('title', array('German Car Popularity', &$Arial)),
        Image_Graph::horizontal(
            Image_Graph::factory('title', array('Popularity', &$Arial1)),
            Image_Graph::horizontal(
                Image_Graph::vertical(
                    $Plotarea = Image_Graph::factory('plotarea'),
                    $Legend = Image_Graph::factory('legend'), 95
                ),
                Image_Graph::factory('title', array('Defects / 1000 units', &$Arial2)), 95
            ), 7
        ), 5
    );
```

Paketbeispiel: Image_Graph (ab 0.3.0dev1)



20 / 26

```
$Legend->setPlotArea($Plotarea);
$Legend->setFont($Arial3);

$Dataset =& Image_Graph::factory('dataset');
$Dataset->addPoint('Audi', 100);
$Dataset->addPoint('Mercedes', 41);
$Dataset->addPoint('Porsche', 78);
$Dataset->addPoint('BMW', 12);

$Dataset2 =& Image_Graph::factory('dataset');
$Dataset2->addPoint('Audi', 10);
$Dataset2->addPoint('Mercedes', 17);
$Dataset2->addPoint('Porsche', 12);
$Dataset2->addPoint('BMW', 21);

$GridY =& $Plotarea->addNew('bar_grid', null, IMAGE_GRAPH_AXIS_Y);
$GridY->setFillStyle(Image_Graph::factory('gradient', array(IMAGE_GRAPH_GRAD_VERTICAL,
'white', 'lightgrey')));
$Plot =& $Plotarea->addNew('bar', array(&$Dataset, 'normal', 'Popularity'));
$FillArray =& Image_Graph::factory('Image_Graph_Fill_Array');
$Plot->setFillStyle($FillArray);
$FillArray->addNew('gradient', array(IMAGE_GRAPH_GRAD_VERTICAL, 'green', 'white'));
$FillArray->addNew('gradient', array(IMAGE_GRAPH_GRAD_VERTICAL, 'blue', 'white'));
$FillArray->addNew('gradient', array(IMAGE_GRAPH_GRAD_VERTICAL, 'yellow', 'white'));
$FillArray->addNew('gradient', array(IMAGE_GRAPH_GRAD_VERTICAL, 'red', 'white'));
$FillArray->addNew('gradient', array(IMAGE_GRAPH_GRAD_VERTICAL, 'orange', 'white'));
```

Paketbeispiel: Image_Graph (ab 0.3.0dev1)



21 / 26

```
$Marker =& $Graph->addNew('Image_Graph_Marker_Array');
$Marker->addNew('Image_Graph_Marker_Icon', './images/audi.png');
$Marker->addNew('Image_Graph_Marker_Icon', './images/mercedes.png');
$Marker->addNew('Image_Graph_Marker_Icon', './images/porsche.png');
$Marker->addNew('Image_Graph_Marker_Icon', './images/bmw.png');
$Plot->setMarker($Marker);

$Plot2 =& $Plotarea->addNew('line', array(&$Dataset2, 'normal', 'Defects'),
                           IMAGE_GRAPH_AXIS_Y_SECONDARY);
$Plot2->setLineColor('blue@0.4');

$Marker =& $Graph->addNew('Image_Graph_Marker_Value', IMAGE_GRAPH_VALUE_Y);
$Plot2->setMarker($Marker);
$Marker->setFont($Arial3);
$Marker->setPadding(7);

$AxisX =& $Plotarea->getAxis(IMAGE_GRAPH_AXIS_X);
$AxisX->setFont($Arial3);

$AxisY =& $Plotarea->getAxis(IMAGE_GRAPH_AXIS_Y);
$AxisY->setDataPreprocessor(Image_Graph::factory
                           ('Image_Graph_DataPreprocessor_Formatted', '%0.0f%%'));

$AxisY->forceMaximum(105);
$AxisY->setFont($Arial3);

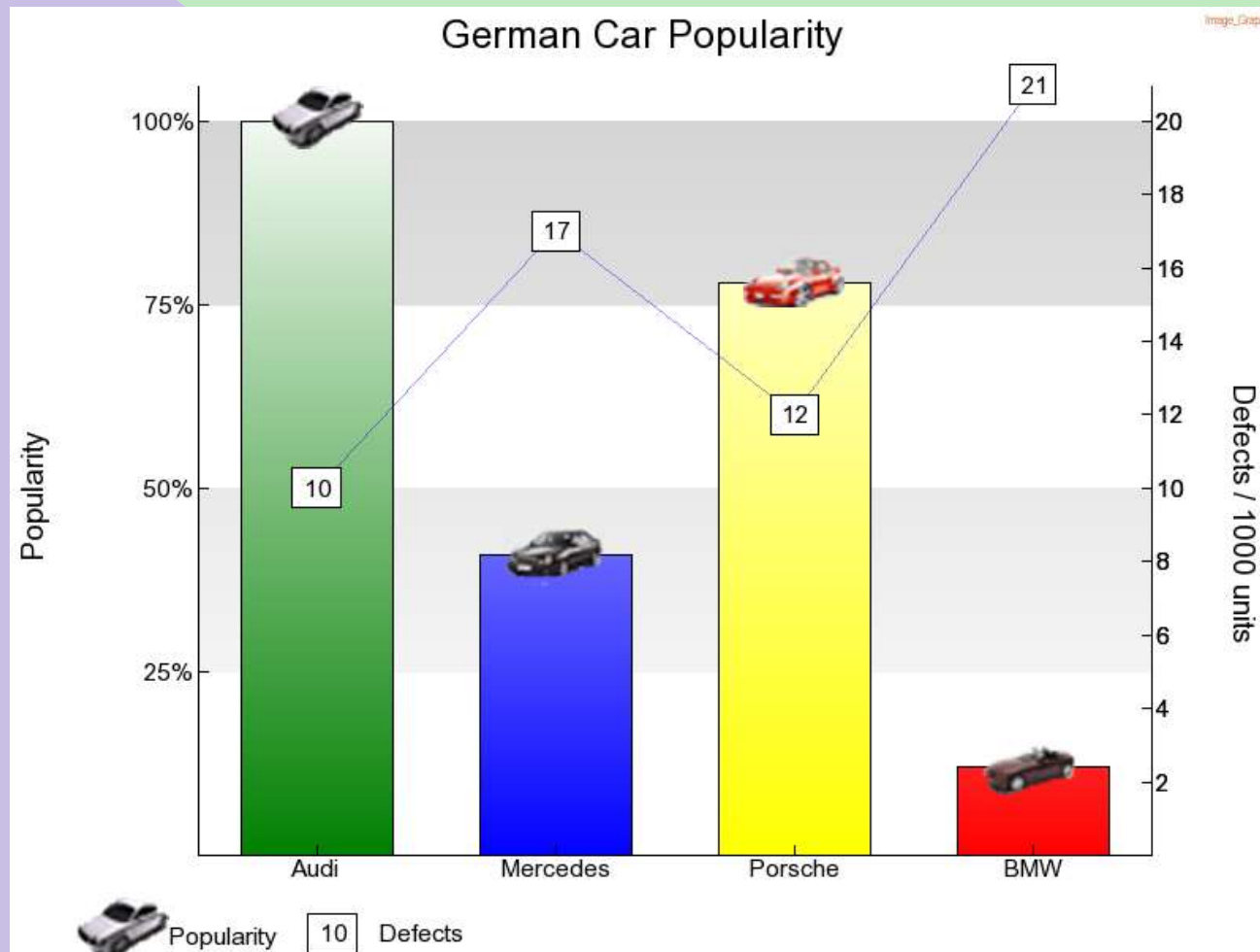
$AxisY =& $Plotarea->getAxis(IMAGE_GRAPH_AXIS_Y_SECONDARY);
$AxisY->setFont($Arial3);

$Graph->done();
```

Paketbeispiel: Image_Graph (ab 0.3.0dev1)



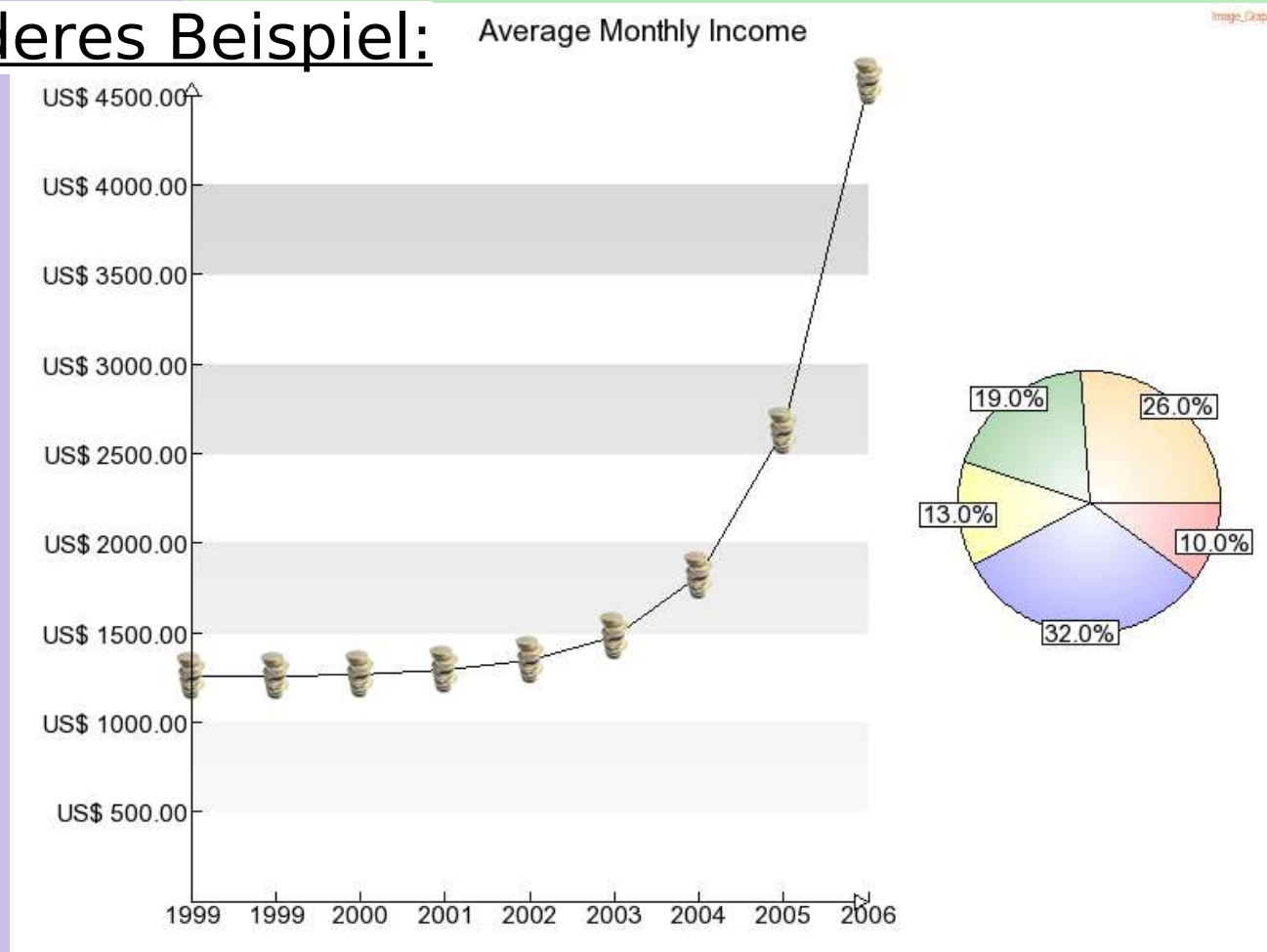
22 / 26



Paketbeispiel: Image_Graph (ab 0.3.0dev1)



Anderes Beispiel: Average Monthly Income



Kontakte



24 / 26

- Mailinglisten
 - pear-general@php.net
 - pear-dev@php.net
- IRC (#pear@EFnet)
- Direkte Kontakte zu den Entwicklern
- Dokumentations-Team (pear-doc@lists.php.net)
- QA-Team (pear-qa@lists.php.net)
- PEAR group (pear-group@php.net)
für Öffentlichkeitsarbeit etc.

Wofür steht PEAR?

- “Enterprise-ready” / Hohe Qualität
- Verbesserte Sicherheit und Flexibilität durch “100 Augen sehen mehr als 2“-Effekt
- Support durch die Community
- Einfache Paket-Verwaltung
- 100% freie Lizenzen (PHP license, LGPL, ... - keine GPL)
- Mitarbeit erwünscht (!)

Fragen?



26 / 26

Danke für's Zuhören!

Interesse geweckt?
Fragen? Kritik?

Stefan Neufeind <neufeind@php.net>

Vortrag online:
<http://pear.speedpartner.de/>