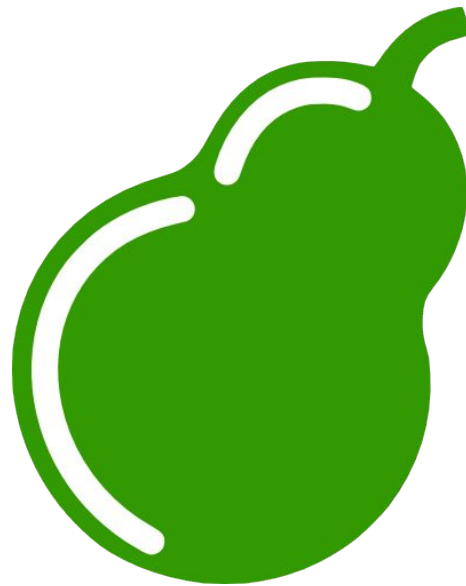


***Feel the power ...***

---

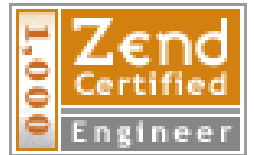
News from



**PEAR**

# About me

- ◆ Stefan Neufeind, from Neuss (close to Düsseldorf)
- ◆ Zend-certified PHP-developer
- ◆ PEAR-maintainer for 3 years
- ◆ „Believes“ in open community-projects
- ◆ Working for SpeedPartner GmbH
  - ◆ „Fullservice internet-solutions“
  - ◆ Consulting
  - ◆ Development, including PHP with PEAR :-)

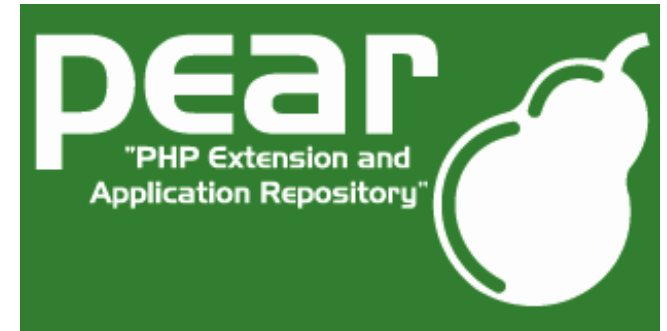


# Agenda

- ◆ What is PEAR?
- ◆ PEAR-package overview
- ◆ Getting PEAR / The installer
- ◆ New in PEAR 1.4.x, including:
  - ◆ PEAR-channels
  - ◆ PHAR („php archive“)
- ◆ Package news / examples
- ◆ Links

# What is PEAR?

## PHP Extension and Application Repository



- ◆ THE repository for PHP
- ◆ Founded 1999 by Stig S. Bakken
- ◆ Collection of high quality PHP-components
- ◆ ~370 packages in 36 categories
- ◆ ~350 package-maintainers, ~630 developers
- ◆ 100% free (licenses: PHP, Apache, BSD, LGPL)
- ◆ Tools for communication and collaboration

# What is PECL?

## PHP Extension Community Library



- ◆ Sister-repository of / split-off from PEAR
- ◆ Sharing many resources but separate website, commandline-installer, ...
- ◆ Components in C-language
  - ◆ Compiled using PEAR installer for specific environment
  - ◆ Not “that” easily installable (especially on shared host)
- ◆ ~140 packages in 28 categories

# What is PEAR?

Consists of:

- ◆ Website (presentation, installer-backend, ...)
- ◆ Installer (commandline, web, GUI)
- ◆ Packages
- ◆ Documentation
- ◆ Support
- ◆ Quality assurance (PEAR-QA)

# PEAR-package overview

- ◆ No applications (almost :-) )  
but components for dedicated needs
  - ◆ Database abstraction
  - ◆ Webservices
  - ◆ Internationalization
  - ◆ Network / Internet
  - ◆ Graphic generation / modification
  - ◆ Validation (email-addresses, IBAN, credit cards, ...)
  - ◆ ...

# PEAR-package overview

## [Authentication](#) (8)

[Auth](#), [Auth\\_HTTP](#), [Auth\\_PrefManager](#), [Auth\\_PrefManager2](#) »

## [Caching](#) (2)

[Cache](#), [Cache\\_Lite](#)

## [Console](#) (6)

[Console\\_Color](#), [Console\\_Getargs](#), [Console\\_Getopt](#), [Console\\_ProgressBar](#) »

## [Date and Time](#) (3)

[Calendar](#), [Date](#), [Date\\_Holidays](#)

## [Event](#) (1)

[Event\\_Dispatcher](#)

## [File System](#) (4)

[File](#), [File\\_Find](#), [File\\_SearchReplace](#), [VFS](#) »

## [Gtk2 Components](#) (6)

[Gtk2\\_EntryDialog](#), [Gtk2\\_FileDrop](#), [Gtk2\\_IndexedComboBox](#), [Gtk2\\_PHPConfig](#) »

## [HTTP](#) (12)

[HTTP](#), [HTTP\\_Client](#), [HTTP\\_Download](#), [HTTP\\_Header](#) »

## [Internationalization](#) (5)

[I18N](#), [I18Nv2](#), [I18N\\_UnicodeString](#), [Translation](#) »

## [Mail](#) (8)

[Mail](#), [Mail\\_IMAP](#), [Mail\\_IMAPv2](#), [Mail\\_Mbox](#) »

## [Benchmarking](#) (1)

[Benchmark](#)

## [Configuration](#) (1)

[Config](#)

## [Database](#) (29)

[DB](#), [DBA](#), [DBA\\_Relational](#), [DB\\_ado](#) »

## [Encryption](#) (8)

[Crypt\\_Blowfish](#), [Crypt\\_CBC](#), [Crypt\\_CHAP](#), [Crypt\\_HMAC](#) »

## [File Formats](#) (23)

[Archive\\_Tar](#), [Archive\\_Zip](#), [Contact\\_AddressBook](#), [Contact\\_Vcard\\_Build](#) »

## [Gtk Components](#) (4)

[Gtk\\_FileDrop](#), [Gtk\\_ScrollingLabel](#), [Gtk\\_Styled](#), [Gtk\\_VarDump](#) »

## [HTML](#) (31)

[HTML\\_AJAX](#), [HTML\\_BBCodeParser](#), [HTML\\_Common](#), [HTML\\_Common2](#) »

## [Images](#) (17)

[Image\\_3D](#), [Image\\_Barcode](#), [Image\\_Canvas](#), [Image\\_Color](#) »

## [Logging](#) (1)

[Log](#)

## [Math](#) (17)

[Math\\_Basex](#), [Math\\_BinaryUtils](#), [Math\\_Complex](#), [Math\\_Derivative](#) »



# PEAR-package overview

- ◆ Components universally usable
- ◆ Cooperation instead of competing packages
- ◆ „Clean“ design
- ◆ Set of rules for quality-assurance and serviceability
- ◆ Used in large variety of applications
- ◆ Business-ready

# Getting PEAR / The installer

- ◆ PEAR-installer bundled since PHP 4.3.0
  - ◆ Automatically available, even on Windows :-)
  - ◆ By default also included when self-compiling
- ◆ For earlier PHP-versions / distributions without PEAR simply use <http://go-pear.org>
  - ◆ e.g. `lynx -source http://go-pear.org | php -q`
  - ◆ or save go-pear-source and open in browser
- ◆ Note: PEAR-path should be in your `include_path` (php.ini)

# Getting PEAR / The installer

- ◆ Common installation-environment for PHP-code (PEAR) and C-extensions (PECL)
- ◆ Runnable on many platforms (Linux, Windows, ...)
- ◆ Various “tastes”: commandline, web, GUI
- ◆ Taking care of package-dependencies etc.
- ◆ Tools for developers
- ◆ Wide adoptions of the installer
  - ◆ pearified, eZ, prado, and other tools/frameworks

# Getting PEAR / The installer

- ◆ Installation of packages (e.g. commandline) with dependency-checks, conflict-checks etc.:

```
# pear install Validate_Finance-alpha
Did not download dependencies: pear/Validate, use --alldeps or --
onlyreqdeps to download automatically
pear/Validate_Finance requires package "pear/Validate" (version >=
0.5.0)
No valid packages found
install failed
```

- ◆ Also allows for optional components, version-dependencies, ...

# Getting PEAR / The installer

- ◆ Useful commands to get you started:
  - ◆ \$ pear help [command]      general or command-help
  - ◆ \$ pear config-show      show settings like paths
  - ◆ \$ pear list      list installed packages
  - ◆ \$ pear remote-list      list available packages
  - ◆ \$ pear install <package>      install a package
  - ◆ \$ pear uninstall <package>
  - ◆ \$ pear list-upgrades      check for package upgrade
  - ◆ \$ pear upgrade <package>      perform an upgrade

# New in PEAR 1.4.x

- ◆ Automatic dependency resolution
  - ◆ `--onlyreqdeps / -o`
  - ◆ `--alldeps / -a`
- ◆ Dependencies on external packages
  - ◆ References to packages by URL
- ◆ Channels
  - ◆ Using multiple channels
  - ◆ Possibility to run your own channel
  - ◆ Cross-channel dependencies

# New in PEAR 1.4.x

- ◆ New package.xml-format
  - ◆ Improved flexibility
- ◆ Mirroring
- ◆ Multiple modules in one package
  - ◆ Bundling packages with an application, ...
- ◆ Post-install-scripts (including interaction), e.g. for
  - ◆ Initializing databases
  - ◆ Moving files to the webroot
- ◆ ReST-webservice

# New in PEAR 1.4.x

- ◆ PHAR-support
  - ◆ Only one PHP-file incl. archive, dynamic unpacking, ...
- ◆ Remote-installation
  - ◆ Installing PEAR without shell access (ftp, ftps, sftp)
  - ◆ Needs additional separate package PEAR\_RemoteInstaller
- ◆ Installation of binary PECL-packages (e.g. on Windows)



# New in PEAR 1.4.x: ReST-webservice

- ◆ PEAR <=1.3 used XML-RPC for repository-access
  - ◆ Widely used, but partially complex
- ◆ PEAR 1.4 uses ReST
  - ◆ text-based, static files possible
  - ◆ Less traffic
  - ◆ Faster response-times
  - ◆ Simplified API
  - ◆ More flexible

# New in PEAR 1.4.x: Post-install scripts

- ◆ Allow quite powerful actions after installation
  - ◆ Flexible API
  - ◆ Script-calls defined in package2.xml
  - ◆ Actions can be bound to specific files
  - ◆ Predefined actions available
  - ◆ Self-implemented actions possible with PHP-scripts
  - ◆ Execution structured in single steps
  - ◆ Interaction between user / PEAR installer

# New in PEAR 1.4.x: PHAR support

- ◆ PHAR = PHP archive
  - ◆ One single PHP-file for an entire application
  - ◆ “PHP's answer to .jar”
  - ◆ Valid tar/php-files
  - ◆ Works on PHP 4.3.0+ and PHP 5
  - ◆ PEAR-installer shipped as a PHAR
  - ◆ Access to files using streams-API:

```
<?php
require_once('example.phar');
require_once('phar://example.phar/myscript.php');
?>
```

# New in PEAR 1.4.x: PEAR-channels

```
# pear list-channels
Registered Channels:
=====
Channel          Summary
pear.php.net     PHP Extension and Application Repository
pecl.php.net     PHP Extension Community Library
__uri           Pseudo-channel for static packages
```

- ◆ More commands related to channels:
  - ◆ `pear channel-discover http://example.com/channel.xml`
  - ◆ `pear channel-alias channel://... example`
  - ◆ `pear update-channels`
- ◆ Channel-descriptions via `channel.xml`

# New in PEAR 1.4.x: PEAR-channels

## ◆ channel.xml-example (1/2):

```
<channel version="1.0"
  xsi:schemaLocation="http://pear.php.net/channel-1.0
                    http://pear.php.net/dtd/channel-1.0.xsd">
  <name>pear.example.com</name>
  <suggestedalias>foo</suggestedalias>
  <summary>Example channel.xml</summary>
  <validatepackage version="1.3.4">Foo_Validate</validatepackage>
  <servers>
    <primary port="8080" ssl="yes">
      <xmlrpc> <!-- default path is xmlrpc.php -->
        <function version="1.0">logintest</function>
        <function version="1.0">package.listLatestReleases</function>
      [...]
    </xmlrpc>
  [...]
</channel>
```

# New in PEAR 1.4.x: PEAR-channels

## ◆ channel.xml-example (2/2):

```
<rest> <!-- no default path, all must be defined in baseurl -->
  <baseurl type="package">
http://pear.example.com/rest/1.0/package</baseurl>
  <baseurl type="category">
http://pear.example.com/rest/1.0/category</baseurl>
</rest>
<soap path="soapy.php"> <!-- default path is soap.php -->
  <function version="1.0">package.listAll</function>
</soap>
</primary>
<mirror server="foo2.example.com/pearmirror">
  <rest>
    <baseurl type="package">
http://foo2.example.com/rest/1.0/package</baseurl>
  </rest>
</mirror> </servers> </channel>
```

# New in PEAR 1.4.x: PEAR-channels

pearadise.net:  
Portal for  
PEAR-  
channels

pearadise.net BETA The pear channel portal.  
Search Browse Register Help Credits

» Start » Search

## Search Pearadise

**Search the pearadise.net crop for different items.**

Please select the type of item you want to search for. The search performs a (subpattern sensitive) fulltext search on several attributes of the element type you selected. Search terms are considered additive. To restrict your search to a specific channel or category you can use the following keyword-syntax:

```
[channel:<channel_uri> [category:<category_name>]]
```

Package

« ← 1 → »

**patPortal**  
MVC-driven framework  
*Last updated on 2005-11-15, 03:01.*

« ← 1 → »

PEAR POWERED PHP POWERED W3C XHTML 1.0 W3C CSS [Impressum](#)

# New in PEAR 1.4.x: PEAR-channels

- ◆ Famous channels include:
  - ◆ eZ
  - ◆ Horde
  - ◆ Gnope.org
  - ◆ Pearified
  - ◆ PHP Application Tools
  - ◆ Seagull / PHPkitchen
  - ◆ Solar
  - ◆ Symfony Project



# PEAR channel server

- ♦ Various ways/tools to set up a channel-server
- ♦ One implementation is Chiara\_PEAR\_Server
  - ♦ PEAR-package, but developed outside of PEAR
  - ♦ <http://pear.chiaraquartet.net/>
  - ♦ Status still alpha
    - ♦ API may change
    - ♦ Still many features to come

# PEAR channel server

- ◆ Discovering Greg Beaver's development-channel

```
$ pear channel-discover pear.chiaraquartet.net
```

- ◆ Installing package from the channel (and required deps)

```
$ pear install -o chiara/Chiara_PEAR_Server-alpha
```

- ◆ Running installation-scripts

- ◆ Asks for settings

- ◆ Creates database

- ◆ Generates channel.xml and REST

```
$ pear run-scripts chiara/Chiara_PEAR_Server
```

# PEAR channel server

## Channel Administration

Logged in as **neufeind** | [My Account](#) | [Logout](#)  
Managing Channel **demo.example.com**

[Create a Category](#)  
[Create a Package](#)  
[Upload a Release](#)  
[Add a Maintainer](#)

### Manage Categories

[Default](#)

### Manage Packages

### Manage Maintainers

[Stefan Neufeind \(neufeind\)](#)

### Add a New Package

**Channel** demo.example.com

**\*Name**

**Category**

**\*License**

**License URL**

**\*Summary**

**\*Description**

**Parent Package**

**Web CVS URL**

**Bug Tracker URL**

# Package news / examples

- ♦ Lately updated packages (small and incomplete selection):
  - ♦ HTML\_AJAX
  - ♦ Image\_Puzzle
  - ♦ Image\_3D
  - ♦ MDB2, MDB2\_Schema and drivers, ...
  - ♦ Net\_CDDB
  - ♦ Services\_Delicious
  - ♦ Validate-packages, e.g. Validate\_Finance
  - ♦ VFS (Virtual File System API)

# Package news: HTML\_AJAX

- ◆ Easy to work with (on client and server)
- ◆ OO JavaScript interface
  - ◆ Class are exposed to Javascript as classes
- ◆ POST used for requests
- ◆ Async and sync calls
- ◆ Proxied calls: through stub objects for PHP classes in JS  
Proxyless calls: utility-functions like `HTML_AJAX.replace()`

# Package news: HTML\_AJAX

- ♦ JavaScript error handling
- ♦ Browser-compatibility:
  - ♦ Firefox 1.0+
  - ♦ Internet Explorer 5.5+ (maybe 5.0+)
  - ♦ Opera 8.5+
  - ♦ Safari 2+ (Most things)

# Package news: HTML\_AJAX

## ♦ server-example (1/2):

```
<?php
class search {
    public function __construct() {
        $this->pdo = new PDO('mysql:host=127.0.0.1;dbname=ajaxdata',
                            'ajaxuser', 'ajaxpw');
    }
    public function get_records($search='') {
        $sql = "SELECT * FROM books ";
        if (trim($search) != '') {
            $sql .= "WHERE title LIKE ";
                $this->pdo->quote('%' . $search . '%');
        }
        return $this->pdo->query($sql)->fetchAll(PDO_FETCH_ASSOC);
    }
}
[...]
```

# Package news: HTML\_AJAX

- ♦ server-example (1/2):

```
[...]
include 'HTML/AJAX/Server.php';
class MyServer extends HTML_AJAX_Server {
    public $initMethods = true;
    public function initSearch() {
        $this->registerClass(new search());
    }
}
$server = new MyServer();
$server->clientJsLocation = '/usr/share/pear/data/HTML_AJAX/js/';
$server->handleRequest();
```

- ♦ Loading generated JavaScript-interface on client:

```
<script src="xul_search.php?client=all" />
<script src="xul_search.php?stub=all" />
```



# Package news: Image\_3D

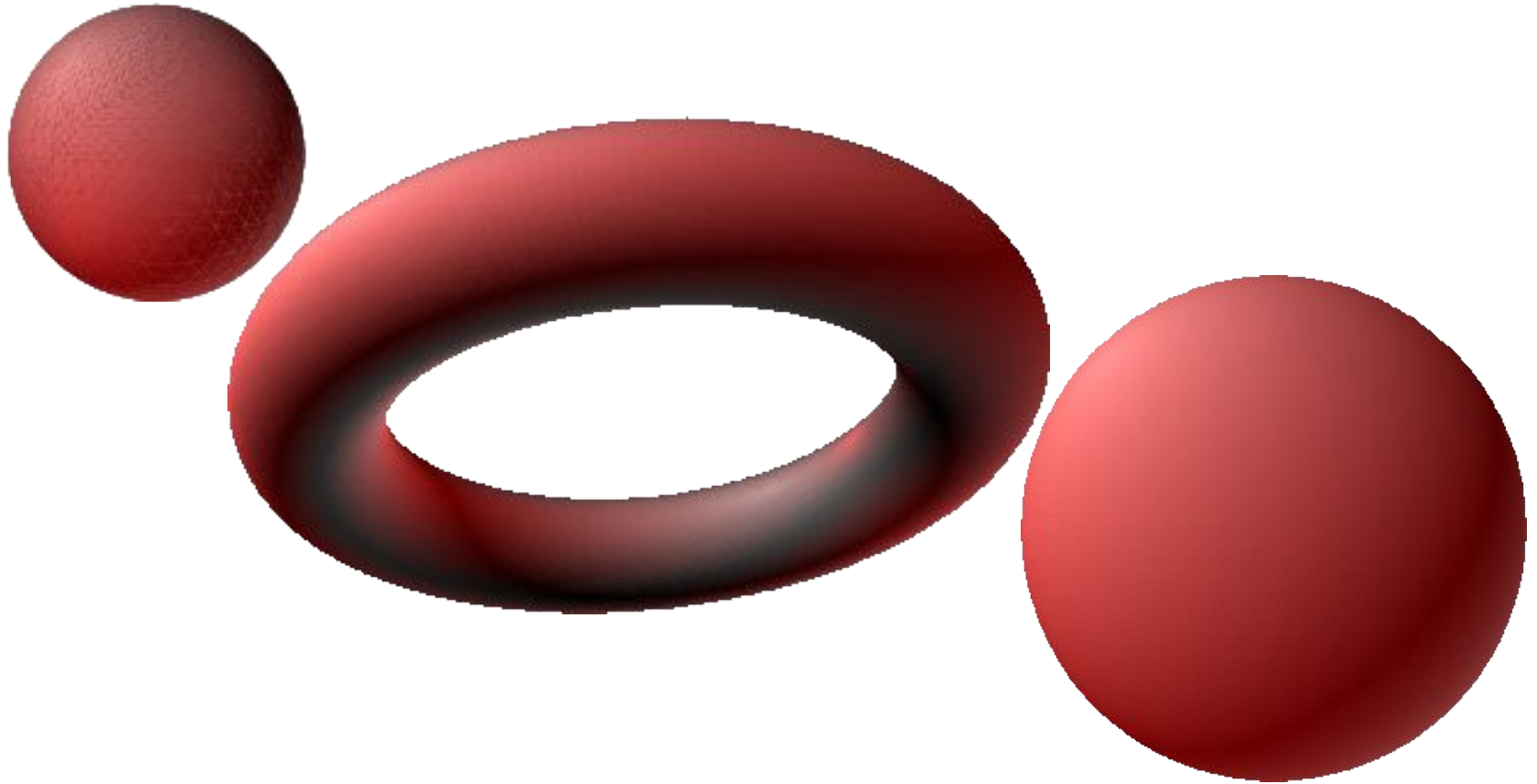
- ◆ Creation of 3 dimensional images using PHP
- ◆ highly object oriented, PHP5
- ◆ Creation of 3D objects like cubes, spheres, maps, text, pie, torus, cone, ...
- ◆ Your own object definitions
- ◆ Your own material definitions
- ◆ Import of 3DSMax files
- ◆ Unlimited number of light sources
- ◆ Output drivers: Rendering via GD, SVG or ASCII

# Package news: Image\_3D

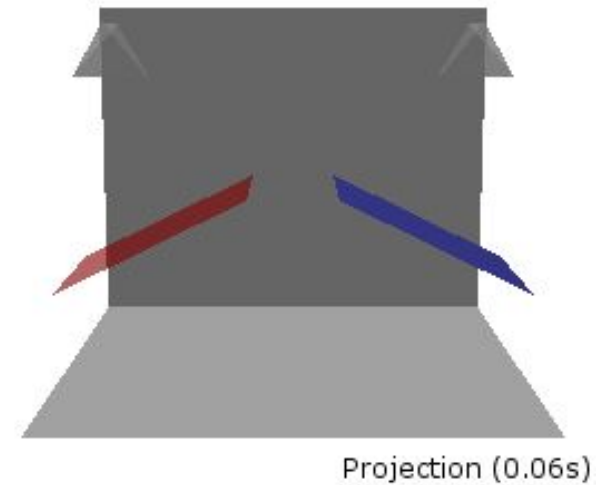
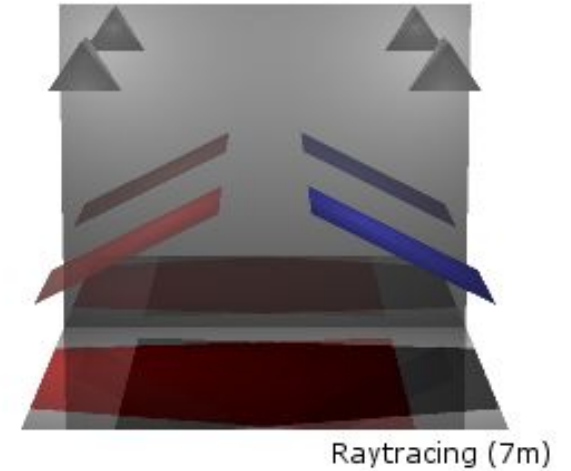
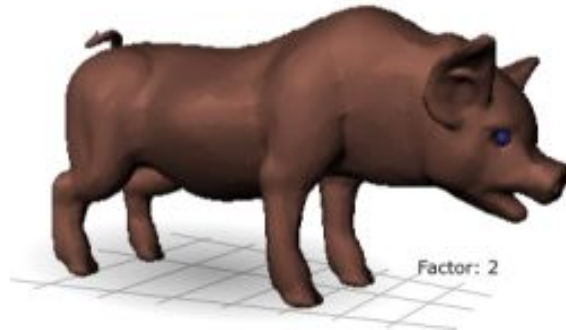
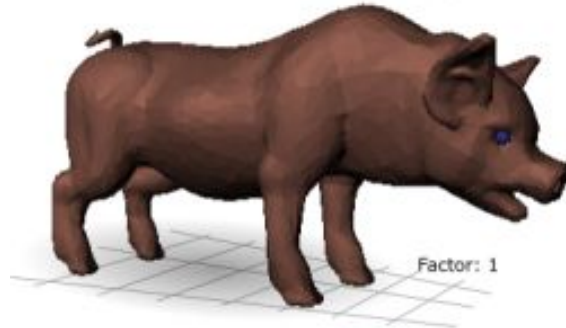
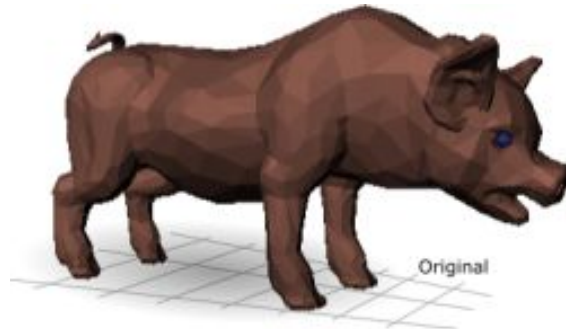
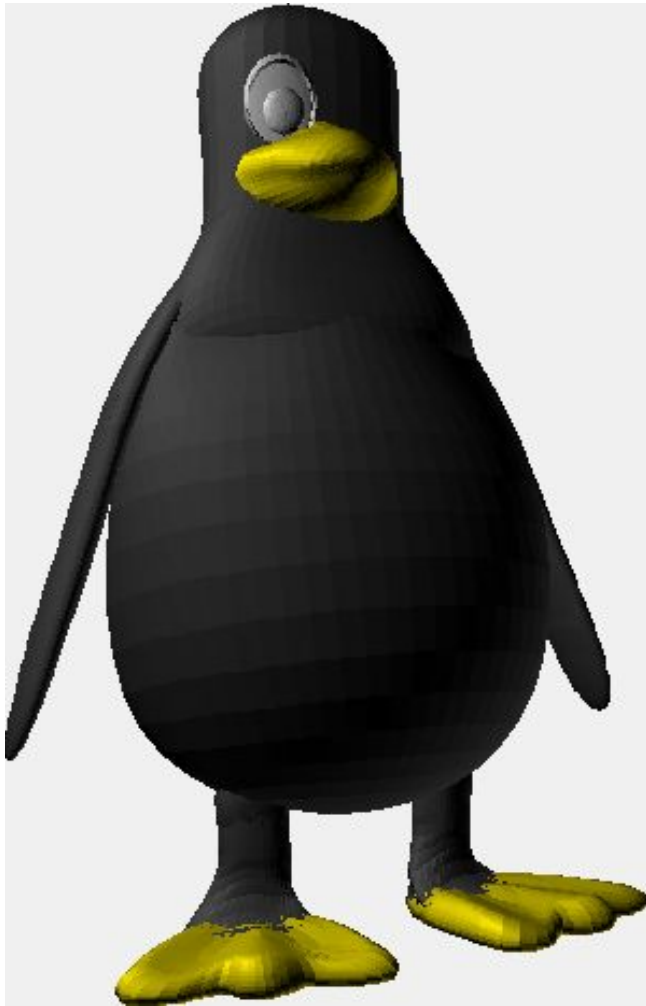
- ◆ Latest enhancements:
  - ◆ Spotlights, pointlights and ambient lights
  - ◆ Bezier curves
  - ◆ Torus and cone
  - ◆ Improved speed
  - ◆ Driver for ASCII-output (including animation)
  
- ◆ Best of all: It's really cool :-)

# Package news: Image\_3D

Image\_3D

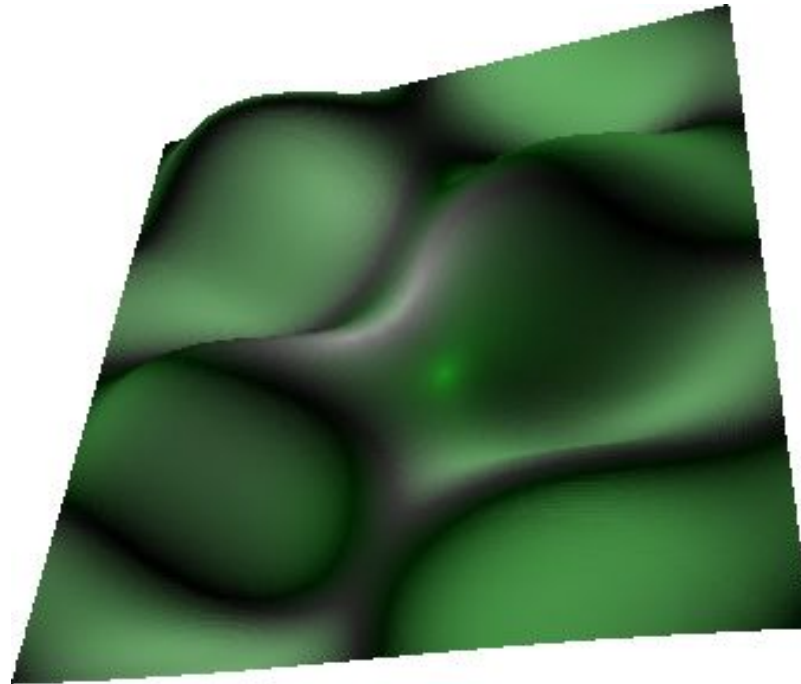


# Package news: Image\_3D



# Package news: Image\_3D

Still pictures ...



... or even animated – hope it works here :-)

# Package news: MDB2

- ◆ “stable” since beginning 2006
- ◆ Supports everything “DB” offers (and more)
- ◆ API-mix of DB and PDO
- ◆ MDB2 for “high abstraction”, PDO for “lower layers”
  - ◆ Working with joined forces; PDO-driver in the works
- ◆ Faster than DB (at least in “micro”-benchmarks)
- ◆ Uses modules
- ◆ Flexible and feature-rich (because of modules)

# Package news: MDB2

- ◆ Most drivers use native prepared-API of RDBMS
- ◆ Full datatype-abstraction (Datatype module)
- ◆ LOBs are provided as regular stream resources
  - ◆ Direct access to data (if RDBMS supports it)
  - ◆ Allows using functions like fread() on LOBs from DB
  - ◆ Allows INSERTs directly from files
- ◆ Debug callback support
- ◆ PHP5 iterator support
- ◆ Lazy connect support (connects when necessary)

# Package news: MDB2

- ◆ Abstraction of functions (Function module)

```
if (!$this->methodExists($this->db->function, 'functionTable')) {  
    return;  
}  
$functionTable_clause = $this->db->function->functionTable();  
$query = 'SELECT 1 '.$functionTable_clause;
```

- ◆ Many methods for administrating database-objects (Manager and reverse module)
  - ◆ Creating tables, sequences, indexes, constraints
  - ◆ Listing tables, indexes, ...
  - ◆ Reverse engineering (table-field details)



# Package news: MDB2

- ◆ Various emulations, where necessary (REPLACE, non correlated sub queries, ...)
- ◆ Functions like “lastInsertId” to work with autoincrements

```
$mdb2->loadModule('Extended');  
// fetch the next ID in the sequence or return php null  
$sid = $mdb2->extended->getBeforeID('foo');  
$query = 'INSERT INTO foo (id, bar) VALUES ('. $mdb2->quote($sid,  
        'integer').', '. $mdb2->quote('xxx', 'text').')';  
$mdb2->exec($query);  
// return $sid or fetch the last inserted id via autoincrement  
$sid = $mdb2->extended->getAfterID($sid, 'foo');
```

- ◆ Several portability modes
- ◆ Buffered and unbuffered resultsets

# Package news: MDB2\_Schema

- ◆ Creation of databases
- ◆ Reverse engineering into XML-format
- ◆ Comparing XML-schema-files
- ◆ Automatic modification of schema according to compare
- ◆ Working directly in DB or as SQL dump

# Package news: MDB2\_Schema

```
$dsn = $_REQUEST['type'].'://'.
    $_REQUEST['user'].':'. $_REQUEST['pass'].'@'.
    $_REQUEST['host'].'/'. $_REQUEST['name'];

$schema =& MDB2_Schema::factory($dsn, $options);

$dump_what = MDB2_SCHEMA_DUMP_STRUCTURE; // ..._CONTENT, ..._ALL
$dump_config = array(
    'output_mode' => 'file',
    'output' => $_REQUEST['file']
);

$definition = $schema->getDefinitionFromDatabase();

$operation = $schema->dumpDatabase($definition, $dump_config,
    $dump_what);
```

# Links

- ▶ PEAR/PECL-website: <http://pear.php.net/>, <http://pecl.php.net/>
- ▶ PEAR-channel aggregator: <http://www.pearadise.net/>
- ▶ Greg Beaver's blog (incl. many examples for installer etc.): <http://greg.chiaraquartet.net/>
- ▶ HTML\_AJAX: [http://wiki.bluga.net/HTML\\_AJAX/HomePage](http://wiki.bluga.net/HTML_AJAX/HomePage)
- ▶ Image\_3D: <http://kore-nordmann.de/>
- ▶ MDB2 homepage with lots of examples: <http://oss.backendmedia.com/MDB2/>
- ▶ MDB2\_Schema-docs and examples: [http://cvs.php.net/viewcvs.cgi/pear/MDB2\\_Schema/docs/](http://cvs.php.net/viewcvs.cgi/pear/MDB2_Schema/docs/)

# Thanks go to ...

- ◆ Kore
- ◆ Lukas
- ◆ Markus
- ◆ Michael
- ◆ Pierre
- ◆ Toby
  
- ◆ All PHP / PEAR-colleagues :-)

# Thank you!

Up-to-date slides available at:

<http://talks.speedpartner.de/>

Questions?

neufeind (at) speedpartner.de