



The PHP Company

# Zend Server



The PHP Company

# Course Introduction

# Why PHP on i?



**Leverage Existing Infrastructure to Develop & Run Dynamic Web Applications Easily**



- **Powerful , Flexible, & Fastest Growing Web Development Language**
- **Native on IBM i/OS**
- **Backing & Support of IBM**
- **Controlled/Driven by PHP Community**
- **Easy to Use & Fast to Deploy**
- **Supports Heterogeneous Environments**
- **Powerful Integration Capabilities**

## **Application Modernization**

Use PHP to web-enable green screen applications, utilizing existing data & business logic without rewriting application

## **Access RPG Programs & Data as Web Services**

Package programs and data as web services and make available to other web applications

## **Consolidate Data in DB2/400**

Move PHP applications that are running on Windows or Linux that access data in DB2 to run in IBM i/OS to speed up or reduce complexity of the applications

## **Access DB2/400 Data via Browser**

Easily access and interact with DB2/400 data using a standard web browser

## **New/Packaged Web Application on IBM i/OS**

Create new applications with PHP on IBM i/OS or leverage thousands of existing Open Sources applications

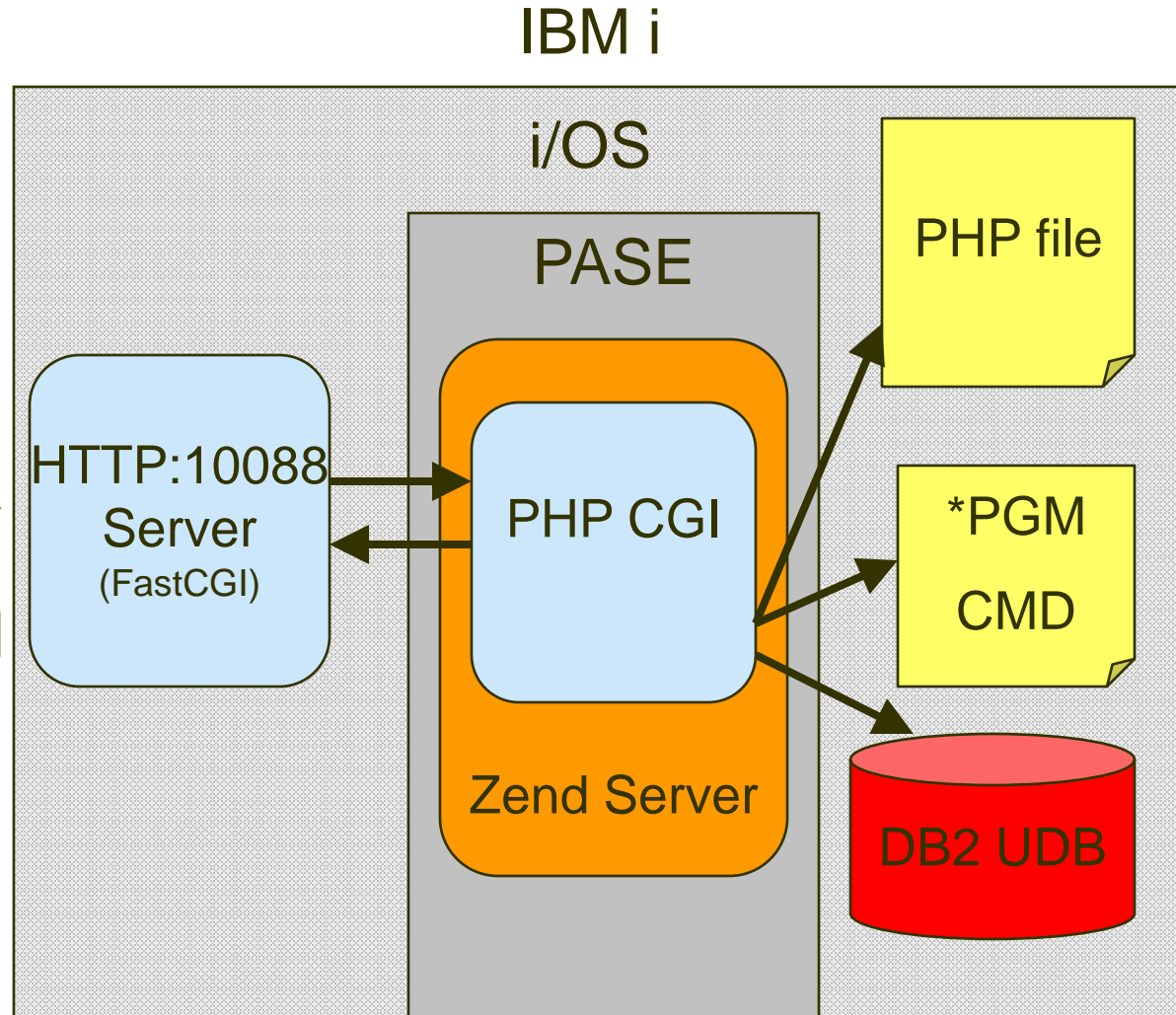
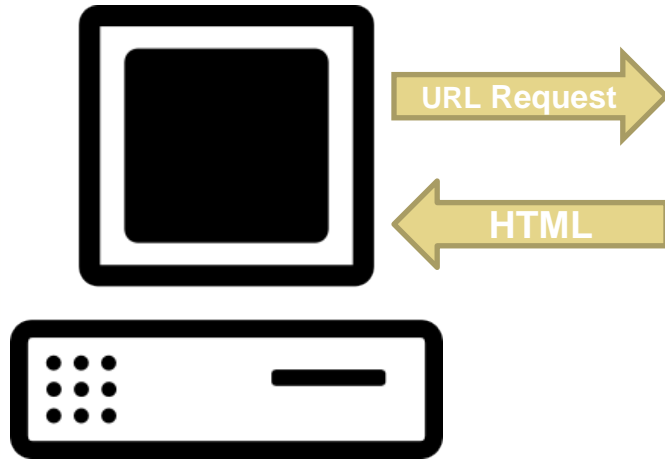
# Key value points CE vs. Enterprise

- Proactive issue monitoring
- Reduce time spent on root cause analysis
- Improve performance of
  - System
  - User
  - Developer
- Capitalize on asynchronous processing (multitasking)

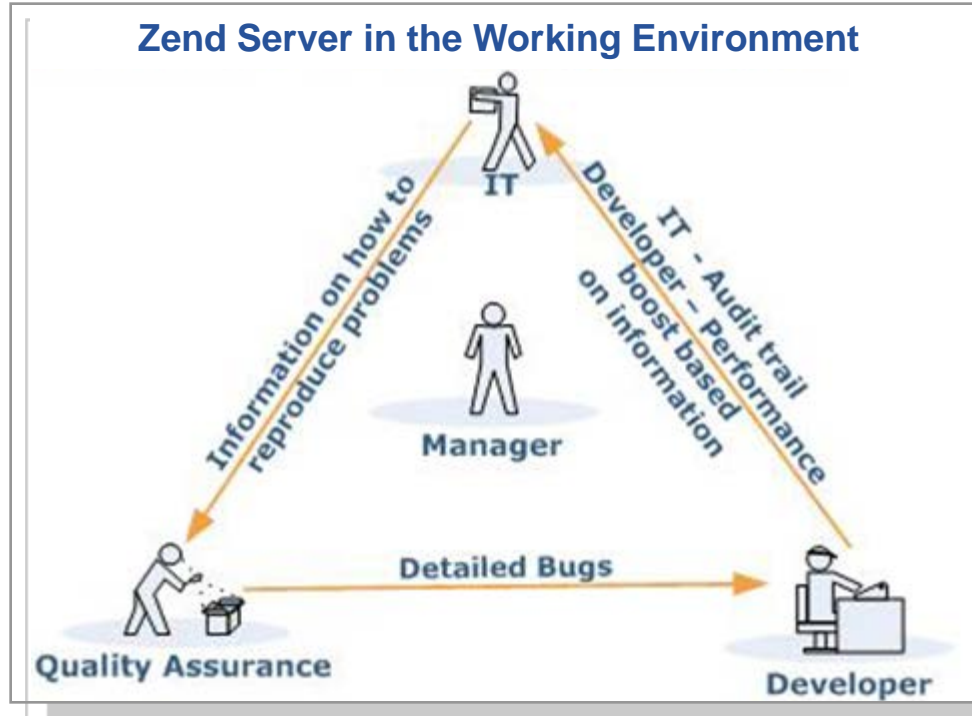
# Zend Server Under the Covers

ILE Apache:10088

- Default configuration FastCGI



# Problem Resolution Lifecycle



The Lifecycle's main goal is to help ease the strain of challenging environments and tight schedules while preventing problematic issues from falling between the cracks

# Zend Server

## What is Zend Server?

- Zend Server is a complete, enterprise-ready Web Application Server that ...
  - ... is very simple to install, configure and use
  - ... has tech support, periodical updates, security hot fixes
  - ... is basically the only reasonable choice on Windows, Linux , IBMi & Mac

Note: Currently Zend Server Community Edition (CE) is the only supported version on Mac

## Who should use Server?

- Server is designed for architects, IT managers, developers, & system administrators who want to maintain high quality of service for their Websites by:
  - Boosting application performance and throughput
  - Maintaining application reliability and security
  - Improving application management



# Zend Server: What Problems Does It Solve?

- Enhances Reliability, Improves Staff Productivity
  - Tracking, installing, configuring and testing dozens of PHP libraries and drivers is a time sink for developers, testers and administrators
  - Rapid updates and code changes that characterize today's fast-paced Web applications further aggravate the challenges of maintaining a reliable PHP runtime environment with ensured application security
- Increases Rate of Successful Deployments
  - Many of the problems encountered during application deployment or in production occur because different PHP versions and configurations are used in development, testing and production
  - Zend Server enables you to deploy your PHP applications with confidence, ensuring every member of your team uses the same, highly reliable environment consistently through each stage of the application life cycle
  - If you need to ship your PHP application to a remote customer, Zend Server's unattended installer facilitates fast and trouble-free deployments



# Zend Server: Optimized PHP Distribution

- Zend Server is a complete, pre-integrated and tested solution that frees you from the hassle of maintaining your PHP infrastructure
  - Access to Zend Technical Support
  - Hot Fixes and Security patches
- Zend Server includes a tested and certified version of PHP
  - The latest PHP 5.2 or PHP 5.3 version (5.4 coming soon)
  - PTF software updates
  - Commonly used extensions and Zend Framework are included
- An Administration Interface provides tools to set up and optimize the development environment
- Special attention is given to creating consistency across operating systems to ensure interoperability and facilitate the requirements of diverse environments that use Linux, Windows, IBMi and Mac

# Background of Development for PHP

[www.zend.com](http://www.zend.com)

**Hello World**

# Sample Script - Hello World

- Obligatory program when starting out
- Confirms correct setup of PHP
- Sample code: (Shortest PHP Macro around!)
- Save script in IFS /www/zendsvr/htdocs/helloworld.php
- Browser URL: <http://ibmisystem:10088/helloworld.php>

```
<?php
    echo "Hello World";
?>
```

This is the Hello World Script x +

← → ↻ ☆ http://15system:89/mpavlak/helloworld.php ▶ [File Icon] [Tools Icon]

New Tab Other bookmarks

Hello World

# Open Source Toolkit for PHP and IBM i

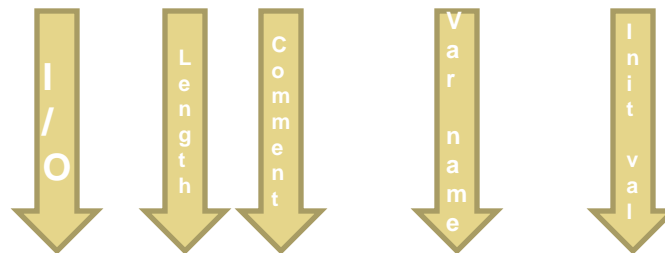
- Collaboration between IBM and Zend
- Built on ILE-RPG, Stored procedures and pure PHP
- Object Oriented, but easy to implement for Procedural developers
- Old toolkit no longer distributed, but still available from 3<sup>rd</sup> party

# Open Source Toolkit

- **Collaboration with IBM and Zend**
- **Built on ILE RPG, Stored Procedures & pure PHP**
- **Up to 10 times faster**
- **Object Oriented, but not too much!**
- **Old toolkit not going away, but support may be chargeable**

# Now the program call

- Set parameters based on function
  - Call program
  - Output results



```
37 $param[] = $ToolkitServiceObj->AddParameterChar('both', 10, 'CODE', 'CODE', $code);
38 $param[] = $ToolkitServiceObj->AddParameterChar('both', 10, 'DESC', 'DESC', $desc);
39
40 $result = $ToolkitServiceObj->PgmCall("COMMONPGM", "ZENDSVR", $param, null, null);
41
42 if($result){
43     showTable( $result['io_param']);
44 }
45 else
46     echo "Execution failed.";
```

# Background of Development for PHP

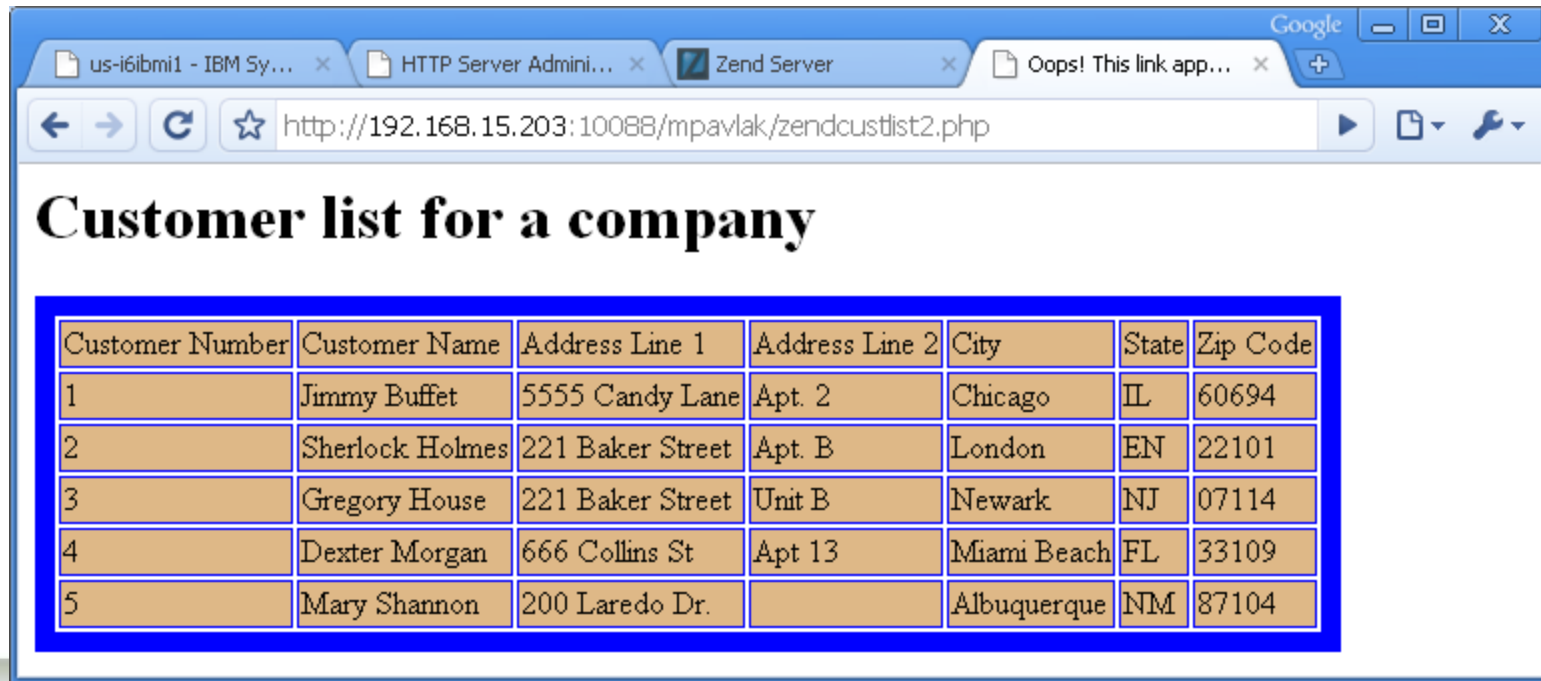
[www.zend.com](http://www.zend.com)

## PHP Data Access Example



# Basic file listing script

- Connect to DB2 i
- Select records from a DB2 table
- Load the records in an HTML table
- Display the output.



The screenshot shows a web browser window with the address bar containing the URL `http://192.168.15.203:10088/mpavlak/zendcustlist2.php`. The page title is "Customer list for a company". Below the title is a table with 7 columns: Customer Number, Customer Name, Address Line 1, Address Line 2, City, State, and Zip Code. The table contains 5 rows of data.

Customer Number	Customer Name	Address Line 1	Address Line 2	City	State	Zip Code
1	Jimmy Buffet	5555 Candy Lane	Apt. 2	Chicago	IL	60694
2	Sherlock Holmes	221 Baker Street	Apt. B	London	EN	22101
3	Gregory House	221 Baker Street	Unit B	Newark	NJ	07114
4	Dexter Morgan	666 Collins St	Apt 13	Miami Beach	FL	33109
5	Mary Shannon	200 Laredo Dr.		Albuquerque	NM	87104

# Connection to DB2 for i

- Need three things:
  - ▶ System, User ID, Password

```
// Standard DB connection to DB2...
$conn = "*LOCAL"; $name = ""; $pwd = "";

$i5link = db2_connect($conn, $name, $pwd);
if (!$i5link)
echo 'Connection failed: '.db2_stmt_error().': '.db2_stmt_errormsg();

$sql = "SELECT * from zenddata.customer";

$stmt = db2_exec($i5link,$sql)
or die("Failed query:".db2_stmt_error().": ".db2_stmt_errormsg());

?>
```

# Reading data...

## Loop through record set

```
<h1> Customer list for a company</h1>
<TABLE BORDER="10" BORDERCOLOR="blue"><tbody BGCOLOR="DEB887">
<tr><td>Customer Number</td><td>Customer Name</td><td>Address Line 1</td>
<td>Address Line 2</td><td>City</td><td>State</td><td>Zip Code</td></tr>

<?php

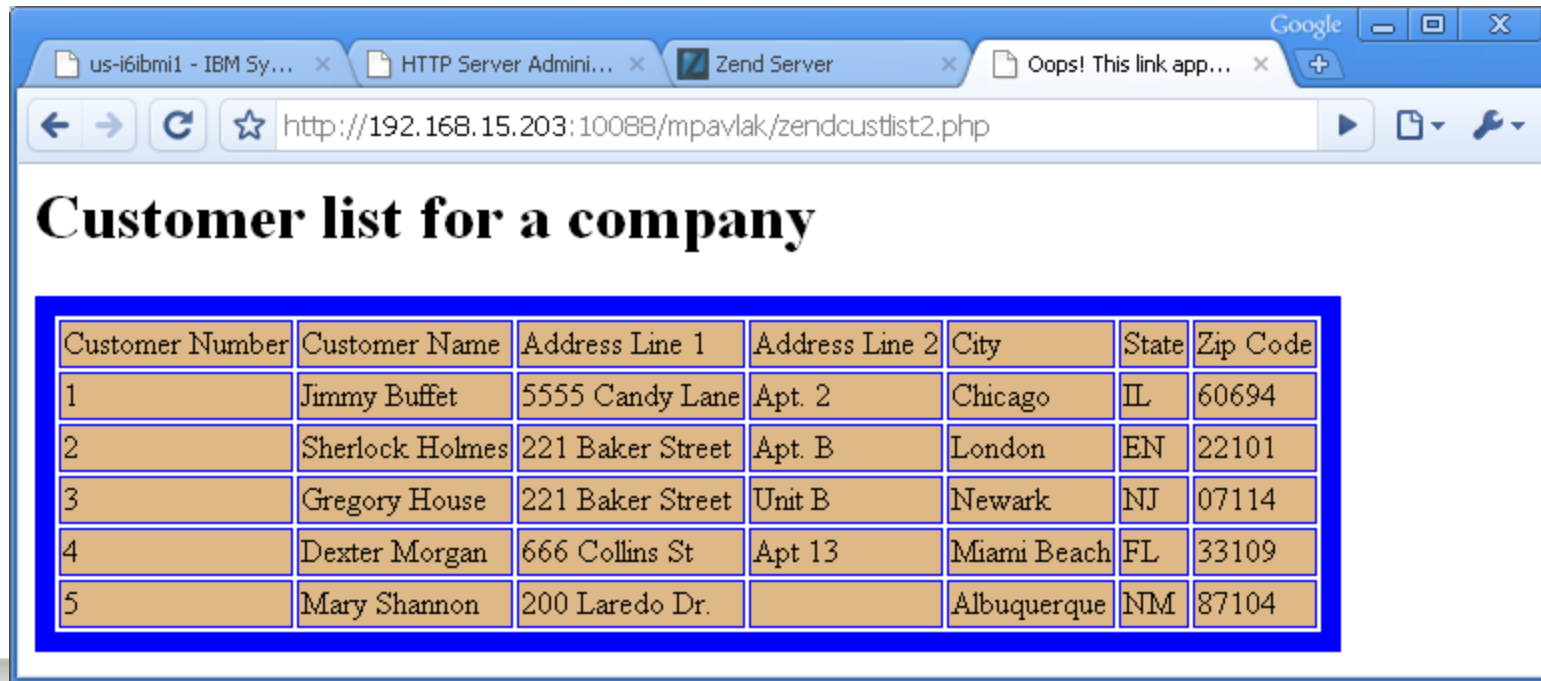
while($row=db2_fetch_array($stmt)){
list( $CUSTOMER_NUMBER, $CUSTOMER_NAME, $CUSTOMER_ADDRESS_LINE1,
$CUSTOMER_ADDRESS_LINE2, $CUSTOMER_CITY, $CUSTOMER_STATE,
$CUSTOMER_ZIP)= $row;

echo(" <tr><td> $CUSTOMER_NUMBER </td><td> $CUSTOMER_NAME </td>
<td> $CUSTOMER_ADDRESS_LINE1 </td><td>$CUSTOMER_ADDRESS_LINE2 </td>
<td> $CUSTOMER_CITY</td><td> $CUSTOMER_STATE </td>
<td> $CUSTOMER_ZIP </td></tr> ");
}
echo ' </table> ';
db2_close($i5link);

?>
```

# Basic file listing script

- Connect to DB2 i
- Select records from a DB2 table
- Load the records in an HTML table
- Display the output.



The screenshot shows a web browser window with the address bar containing the URL `http://192.168.15.203:10088/mpavлак/zendcustlist2.php`. The page title is "Customer list for a company". Below the title is a table with 7 columns: Customer Number, Customer Name, Address Line 1, Address Line 2, City, State, and Zip Code. The table contains 5 rows of data.

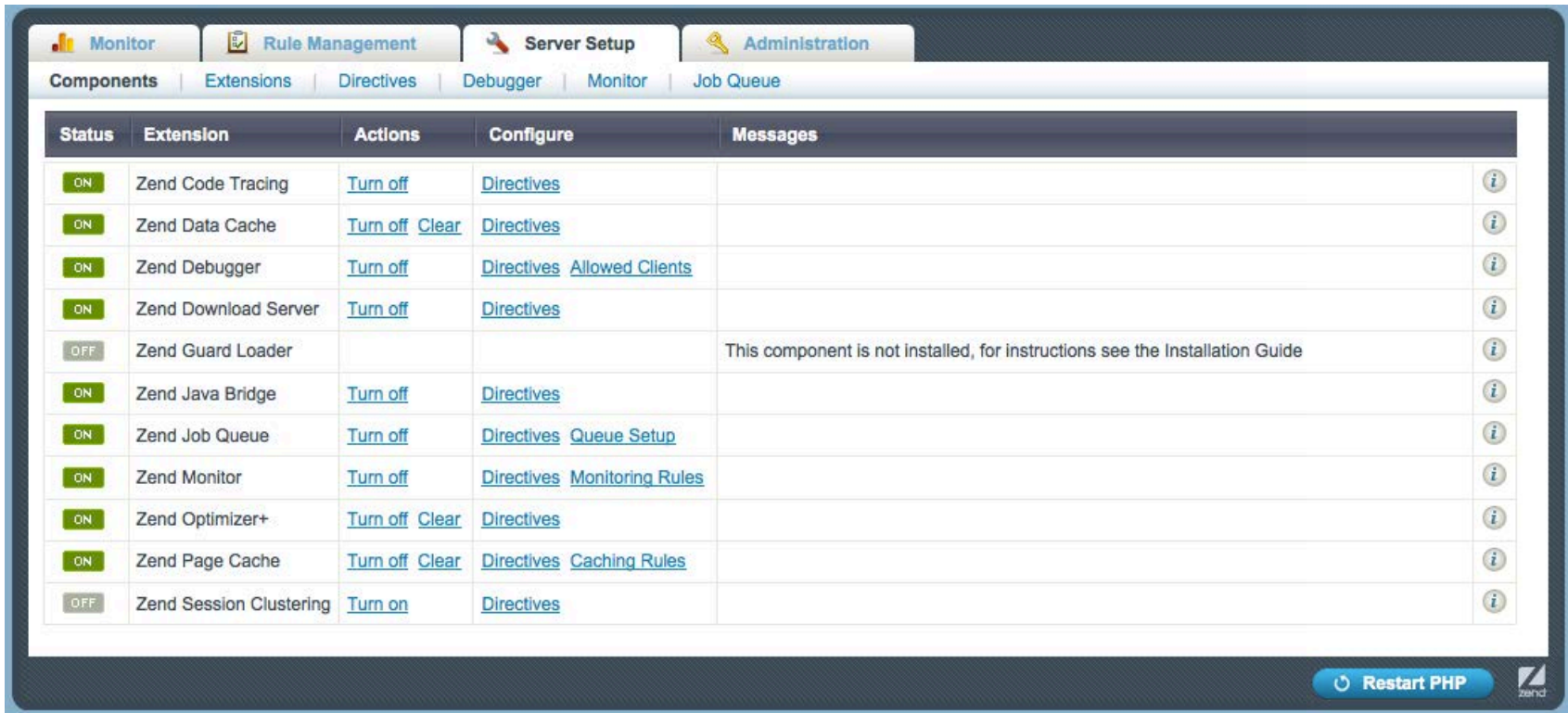
Customer Number	Customer Name	Address Line 1	Address Line 2	City	State	Zip Code
1	Jimmy Buffet	5555 Candy Lane	Apt. 2	Chicago	IL	60694
2	Sherlock Holmes	221 Baker Street	Apt. B	London	EN	22101
3	Gregory House	221 Baker Street	Unit B	Newark	NJ	07114
4	Dexter Morgan	666 Collins St	Apt 13	Miami Beach	FL	33109
5	Mary Shannon	200 Laredo Dr.		Albuquerque	NM	87104



The PHP Company

# Server Setup

# Server Setup | Components



The screenshot displays the 'Server Setup' tab in the Zend Server Administration interface. The interface includes a navigation bar with 'Monitor', 'Rule Management', 'Server Setup', and 'Administration'. Below the navigation bar, there are sub-tabs for 'Components', 'Extensions', 'Directives', 'Debugger', 'Monitor', and 'Job Queue'. The main content area features a table with the following columns: Status, Extension, Actions, Configure, and Messages. The table lists various components such as Zend Code Tracing, Zend Data Cache, Zend Debugger, Zend Download Server, Zend Guard Loader, Zend Java Bridge, Zend Job Queue, Zend Monitor, Zend Optimizer+, Zend Page Cache, and Zend Session Clustering. Each row includes a status indicator (ON/OFF), the component name, action links (Turn on/off, Clear), configuration links (Directives, Allowed Clients, Queue Setup, Monitoring Rules, Caching Rules), and a message icon.

Status	Extension	Actions	Configure	Messages
ON	Zend Code Tracing	<a href="#">Turn off</a>	<a href="#">Directives</a>	
ON	Zend Data Cache	<a href="#">Turn off</a> <a href="#">Clear</a>	<a href="#">Directives</a>	
ON	Zend Debugger	<a href="#">Turn off</a>	<a href="#">Directives</a> <a href="#">Allowed Clients</a>	
ON	Zend Download Server	<a href="#">Turn off</a>	<a href="#">Directives</a>	
OFF	Zend Guard Loader			This component is not installed, for instructions see the Installation Guide
ON	Zend Java Bridge	<a href="#">Turn off</a>	<a href="#">Directives</a>	
ON	Zend Job Queue	<a href="#">Turn off</a>	<a href="#">Directives</a> <a href="#">Queue Setup</a>	
ON	Zend Monitor	<a href="#">Turn off</a>	<a href="#">Directives</a> <a href="#">Monitoring Rules</a>	
ON	Zend Optimizer+	<a href="#">Turn off</a> <a href="#">Clear</a>	<a href="#">Directives</a>	
ON	Zend Page Cache	<a href="#">Turn off</a> <a href="#">Clear</a>	<a href="#">Directives</a> <a href="#">Caching Rules</a>	
OFF	Zend Session Clustering	<a href="#">Turn on</a>	<a href="#">Directives</a>	

Restart PHP

# Log Rotation IBM i

Consider file size and usage:

- Knowledgebase (KB)
- Manual, for now
- Php.log - PHP Messages
- Apache log - Apache web server access and errors

# Ports and Services

IBM i:

- After installation, Server's components use the following ports:
  - **10088**
    - Default web server port
    - For user & admin
    - Can be easily changed via port 2001
  - **Java Bridge:** The Java Bridge daemon listens on port 10001
    - To change this setting, go to the Administration Interface and configure the Extension called "Zend Java Bridge"
  - **MySQL**
    - Listens on port 3306 (Standard for MySQL)



# Zend Extensions

The following are some common extensions:

- `date` Built-in
- `imap` Enabled
- `pdo_sqlite` Built-in
- `tidy` Enabled
- `db2` Enabled

# .ini Files for Extensions & Directives (IBM i)

- Configuration Files related to Extensions and Directives:

```
/usr/local/zendsvr/etc/php.ini
```

- To see a list of all the `e*.ini` files used by Server, you can use the command line (in QP2TERM):

```
ls -la /usr/local/zendsvr/etc/conf.d/*.ini
```

# Server Setup | Directives

- Edit PHP configurations from the Administration Interface (Directives page) to view and configure commonly used directives
- Available directives are grouped by category in expandable lists
  - Click the arrow next to the category name to expand the list / options
- Where relevant, input fields are available to change a directive's value
- The initial display shows the most commonly used Directives ("Popular" view)
  - Click "All" for the full list of directives
  - Use the "Search" component to locate a specific directive
  - Use *ext:<extension\_name>* to find directives by extension

# Directives: Examples

The following are some directives, which we will look at later in the security best practices section:

- `disable_functions`
- `disable_classes`
- `magic_quotes_gpc`
- `allow_url_include`

# Server Setup | Debugger

- The Debugger feature enables PHP code analysis (debugging and profiling) and code repair on remote servers
- Configure Host configurations with the following two options: either *allow* or *deny* permission to initiate debugging and profiling sessions

# Server Setup | Monitor

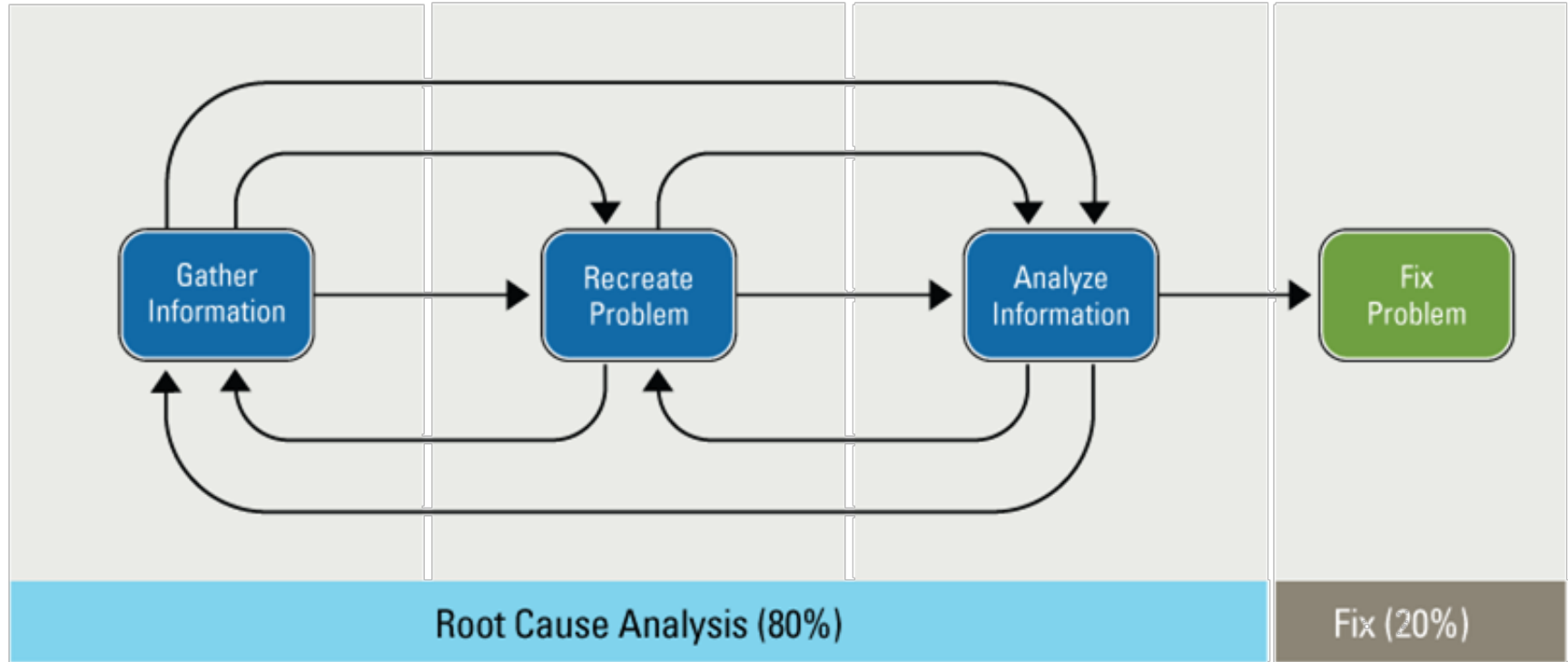
- Monitor is the feature that captures a collection of runtime-related information, called Events
  - An event is triggered according to the Monitor component's rule settings
  - We will look at Rule Management in the next module
- The Monitor page within Server Setup is the settings definition page, used to integrate the overseeing feature with Zend Studio for maximal ease-of-use
- Mail settings are also defined on this page



The PHP Company

# Using Monitor

# The Problem with Problem Resolution...



- Most problem resolution time is spent on identifying root cause
- Problem reproduction is often difficult and time-consuming
- Many possible sources: server load, input data, database state, etc.

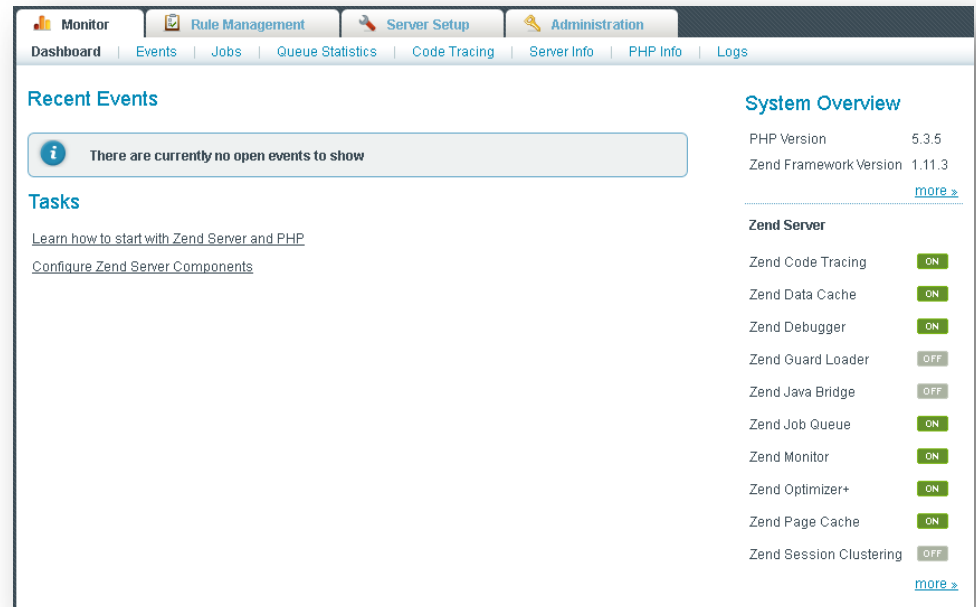


# Root Cause Analysis

- Zend Server monitors your PHP application for...
  - Slow functions or request executions
  - PHP Errors, failing functions or queries
  - High memory consumption
- You are automatically notified of issues via email alerts
- Code Tracing pinpoints the request execution that triggered the issue for fast resolution

# Monitor | Dashboard

- The Dashboard is the default page for Zend Server, displayed upon successfully logging in to the Administration Interface
- It provides quick access links and views to key information
- The information displayed is divided into three categories
  - Recent Events
  - Tasks
  - System Overview



The screenshot displays the Zend Server Administration Interface Dashboard. The top navigation bar includes tabs for Monitor, Rule Management, Server Setup, and Administration. Below this, a secondary navigation bar lists links for Dashboard, Events, Jobs, Queue Statistics, Code Tracing, Server Info, PHP Info, and Logs. The main content area is divided into three sections: Recent Events, Tasks, and System Overview. The Recent Events section shows a message: "There are currently no open events to show". The Tasks section contains links for "Learn how to start with Zend Server and PHP" and "Configure Zend Server Components". The System Overview section displays system information: PHP Version 5.3.5, Zend Framework Version 1.11.3, and a list of Zend Server components with their status (ON/OFF) indicated by green/red buttons. The components listed are: Zend Code Tracing (ON), Zend Data Cache (ON), Zend Debugger (ON), Zend Guard Loader (OFF), Zend Java Bridge (OFF), Zend Job Queue (ON), Zend Monitor (ON), Zend Optimizer+ (ON), Zend Page Cache (ON), and Zend Session Clustering (OFF). A "more" link is present at the bottom of the System Overview section.

# Monitor | Events

- Recall: You set the rules for event generation in Rules Management
- An event indicates that something happened in your environment that exceeded your definitions and standards of how the PHP code should run
- Three major functions from this page:
  - Find Events
  - View Event Details
  - Change an Event Status

## Issues vs. Events:

- Each issue is an aggregation of one or more events with shared, pre-defined characteristics
  - These shared characteristics help to identify the source of the event and the surrounding circumstances

# Monitor | Events

zend Server Help | About | Logout

Monitor Rule Management Server Setup Administration

Dashboard Events Jobs Queue Statistics Code Tracing Server Info PHP Info Logs

Filter: \*Unsaved Custom Filter\* [Hide Filter Details](#) Go to event by ID:  Go

Status  Open  Closed  Reopened  Ignored  Add Conditions:  
 Origin  
 Severity

Date Instances occurred after  and before

Rule Name Generated by one of 1 rules.

Total: 3 Last refresh time: 14-Jul-2011 13:04

ID	Last Occ.	Count	Generated by Rule	Origin	Severity	Status
<input type="checkbox"/> 000012	14-Jul 12:27	1	High Memory Usage (Absolute)	http://drupal.vm/node/5	Warning	Open
<input type="checkbox"/> 000010	14-Jul 12:27	1	High Memory Usage (Absolute)	http://drupal.vm/node/3	Warning	Open
<input type="checkbox"/> 000004	14-Jul 12:26	1	High Memory Usage (Absolute)	http://drupal.vm/node/19	Warning	Open

Change status to

# Monitor | Events

- Two types of actions available from this page - Basic and Advanced
  - Basic tasks include simple tasks, like returning to the Events page, refreshing an issue's details, and detaching an issue into a popup window
  - Advanced tasks include drilling down to detect the source and investigate the problem, which sometimes will also display the solution

# Monitor | Events

- Details are characterized as Basic, General, and Group
  - (a) Basic:
    - ID and Rule Name
    - Occurrence Info
    - Status
    - Severity
  - (b) General Details:
    - URL and Source File
    - Function Name
    - Error String and Type

# Monitor | Events

- Details are characterized as Basic, General, and Group

## (c) Group Details

- Grouping is an additional aggregation layer applied to an issue
  - Inside a single issue, events are divided into groups according to the time they occurred
- A new group is created only if there was no activity for at least five minutes
- If a new event occurs after five minutes, a new group is added to the issue
  - Drilldown: Click on a specific group and the details collected for that group display
  - Diagnose: Available options are: Debug, Profile, and Open File  
Note: server specified in Settings (default is originating server)
  - Export: Generate an XML file with the basic and general details of a particular group
  - Change Status: Used for management; filter on status

# Monitor | Events

The screenshot displays the Zend Server Monitor interface. At the top, there are navigation tabs for Monitor, Rule Management, Server Setup, and Administration. Below these are sub-tabs for Dashboard, Events, Jobs, Queue Statistics, Code Tracing, Server Info, PHP Info, and Logs. The main content area shows an event titled "12 - High Memory Usage (Absolute)" which occurred once at 14-Jul-2011 12:27. The event status is "Open" and severity is "Warning". The URL is "http://drupal.vm/node/5". A table lists the event details:

Start Time	Count	Memory (kb)
14-Jul 12:27	1	8180

Below the table, the request details are shown for a GET request to "node/5". The request includes a cookie with session and debugger information. At the bottom, there are buttons for "Debug Event", "Profile Event", and "Settings". A "Restart PHP" button is visible in the bottom right corner.



# Monitor | Server Info

- The Server Info page displays environment details, specifically:

Zend Server	Zend Guard Loader
PHP	Zend Java Bridge
Web Server	Zend Job Queue
Zend Framework	Zend Monitor
Zend Code Tracing	Zend Optimizer+
Zend Data Cache	Zend Page Cache
Zend Debugger	Zend Session Clustering*
Zend Download Server*	

\*Not available with Zend Server for IBM i

# Monitor | PHP Info

- The PHP Info screen is a read-only page that outputs information about the current state of PHP
- It provides easy access to the information contained in the `php.ini` file, including:
  - PHP compilation options and extensions
  - PHP version and environment
  - Server information and environment
  - OS version information
  - Paths, master and local values of config options
  - HTTP headers
  - PHP License

PHP Version 5.3.5 	
<b>System</b>	Windows NT KATT-PC 6.0 build 6002 (Windows Vista Ultimate Edition Service Pack 2) i586
<b>Build Date</b>	Mar 9 2011 15:46:03
<b>Compiler</b>	MSVC9 (Visual C++ 2008)
<b>Architecture</b>	x86
<b>Configure Command</b>	cscript /nologo configure.js "--disable-phar" "--disable-ipv6" "--disable-zts" "--enable-cgi" "--disable-bcmath" "--disable-calendar" "--disable-odbc" "--disable-tokenizer" "--disable-xmlreader" "--disable-xmlwriter" "--without-sqlite" "--without-wddx" "--enable-debug-pack" "--enable-cli-win32" "--enable-pdo" "--with-openssl" "--with-php-build" "--with-libxml" "--with-pdo-sqlite"
<b>Server API</b>	CGI/FastCGI
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	C:\Windows
<b>Loaded Configuration File</b>	C:\Program Files (x86)\Zend\ZendServer\etc\php.ini
<b>Scan this dir for additional .ini files</b>	(none)
<b>Additional .ini files parsed</b>	(none)
<b>PHP API</b>	20090626
<b>PHP Extension</b>	20090626
<b>Zend Extension</b>	220090626
<b>Zend Extension Build</b>	API220090626,NTS,VC9
<b>PHP Extension Build</b>	API20090626,NTS,VC9
<b>Debug Build</b>	no
<b>Thread Safety</b>	disabled

# Monitor | Logs

- Logs provide critical insight into errors and unwanted application behavior within the system environment
- The information available is directly related to Operating System
  - **Apache: 3 Logs**
    - PHP Error Log (references the `error_log` directive)
    - Apache Error Log (references the installation location)
    - Apache Access Log (references the installation location)
  - **IIS Servers**
    - PHP Error Log
- Log info can be viewed directly from the Administration Interface
- Additional logs can be added to the page, using XML
  - This requires accessing backend application files that usually should not be changed
  - Therefore, this should only be executed by someone who completely understands the concept and procedure



The PHP Company

Caching

# Break

# Caching and App Performance Optimization

- Zend Server gives you several ways to optimize applications:
  - **OpCode Optimization and Caching**
    - Reduces disk access by caching compiled bytecode
  - **Page Caching**
    - Cache entire HTTP response based on highly customizable parameters
  - **Data Caching**
    - API to cache data items (DB query results, web service calls, etc.)

# Client-Side Caching

- Zend Server offers integrated support for client-side caching
  - Utilizes the client-side caching mechanisms already built into browsers and web servers
  - Content (web pages, PDF files, media files) are cached on the end-user's machine and are only re-sent from the server when the content "expires"
  - Client-side caching is transparent to the end-user

# Page Caching (Enterprise)

- Page (Content) Caching is a process of running code once & saving output on the server for re-use, to improve performance
- Files should be cached when their content is relatively stable and does not require constant change
  - Not recommended for files whose output changes on every request
- Server supports URL-based full-page caching for Zend Framework and other MVC architectures
- If full-page caching is unavailable, developers can use a set of data caching APIs (Data Cache) to improve performance from within the code
  - Handles caching in shared memory or on disk
  - Supports putting cached items into namespaces for easier management

# Data Caching

- **The Data Cache API is used by inserting its functions into the PHP code**
- **The API caches partial PHP outputs using Memory or Disk**
- **It features:**
  - Storing Variables to the Cache
  - Fetching Variables from the Cache
  - Deleting Variables from the Cache
  - Clearing the Cache
  - Disk/Memory (SHM) storage
  - Caching using namespaces
  - Cache folder depth configuration



# Data Cache: Disk vs. Shared-Memory

- Memory caching improves server responsiveness primarily in environments that are running high-traffic applications
  - Off-load activity directed towards their Hard Disk in order to help increase performance and responsiveness
  - Options available on where to store cached variables
- Disk caching is more suitable for smaller applications and ensures cached content is available after restarting the machine
- SHM/Disk storage is implemented by using the appropriate API functions and configuring the Data Cache directives
  - Storage Options example: a simple key with no namespace stored on disk



The PHP Company

# Optimizer+

# Optimizer+

- The Optimizer+ allows you to gain a performance boost by reducing code compilation time:
  - Once PHP code is compiled for the first time, it is saved in the server's memory
  - Each time the code is called, the pre-compiled version is used instead of waiting for the code to compile causing a delay each time the code is used
- NOTE: Do not confuse Optimizer+ with Caching...

# Optimizer+ and Blacklists

- Creating a Blacklist provides an alternative, effective way to exclude code storage for particular PHP files (selectively disabling Optimizer+)
- Recommended files for Blacklisting:
  - Directories containing files that are larger than the allocated memory defined in `zend_optimizerplus.memory_consumption`
  - Directories containing more files than the allocated quantity of files as defined in `zend_optimizerplus.max_accelerated_files`
  - Large files that have high memory consumption
    - If you have exhausted all your allocated memory, select scripts that you know are the largest and slowest to execute and blacklist them
  - Files that have long execution time (makes the compilation save irrelevant)
  - Code that is modified on-the-fly (for example, an auto-generated template files)



The PHP Company

# Code Tracing

# Code Tracing

- Code Tracing records live code execution in production
- Activated automatically through extant monitoring rules, or manually
- Can replay a recorded code execution trace to quickly pinpoint root cause of functional and performance problems
- Reduce time to resolution by ~ 50%

# Code Tracing

Tracing ID: 10476.1 07-Jan-2010 21:48

Traced URL Request: http://vm-php-demo.zend.net/blog/wp-admin/index-extra.php?jax=planetnews

Trace Size: 1.01 MB

URL After Rewrites: http://vm-php-demo.zend.net/blog/wp-admin/index-extra.php

Tracing Tree

Tracing Stats

Show memory usage

Highlight the most time-consuming path

Next child in path

3 Errors

Search

Traced Functions	Running Time	(% of total)	(ms)	Called from: (line)
▼ REQUEST /blog/wp-admin/index-extra.php			11,014.91	index-extra.php
✓ HEADER Set-Cookie: ZDEDebuggerPreser		<1%	0	index-extra.php
▼ INCLUDE /usr/share/wordpress/wp-admin		100%	11,014.493	index-extra.php
▶ INCLUDE /usr/share/wordpress/wp-admin		15%	1,629.669	index-extra.php (2)
▶ INCLUDE /usr/share/wordpress/wp-admin		<1%	1(µs)	index-extra.php (3)
▶ INCLUDE /usr/share/wordpress/wp-includ		1%	70.032	index-extra.php (4)
▶ f get_option()		<1%	91(µs)	index-extra.php (6)
▶ f get_option()		<1%	73(µs)	index-extra.php (6)
✓ HEADER REPLACE Content-Type: text/t		<1%	0	index-extra.php
▼ f wp_dashboard_secondary_output()		85%	9,314.282	index-extra.php (19)
▶ f get_option()		<1%	93(µs)	ashboard_secondary_output()(379)
! ERROR [2048] Only variables should		<1%	0	ashboard_secondary_output()(380)
▼ f fetch_rss()		85%	9,314.169	ashboard_secondary_output()(381)
▶ Arguments: http://planet.wordpress		<1%	0	fetch_rss()



The PHP Company

# Job Queue



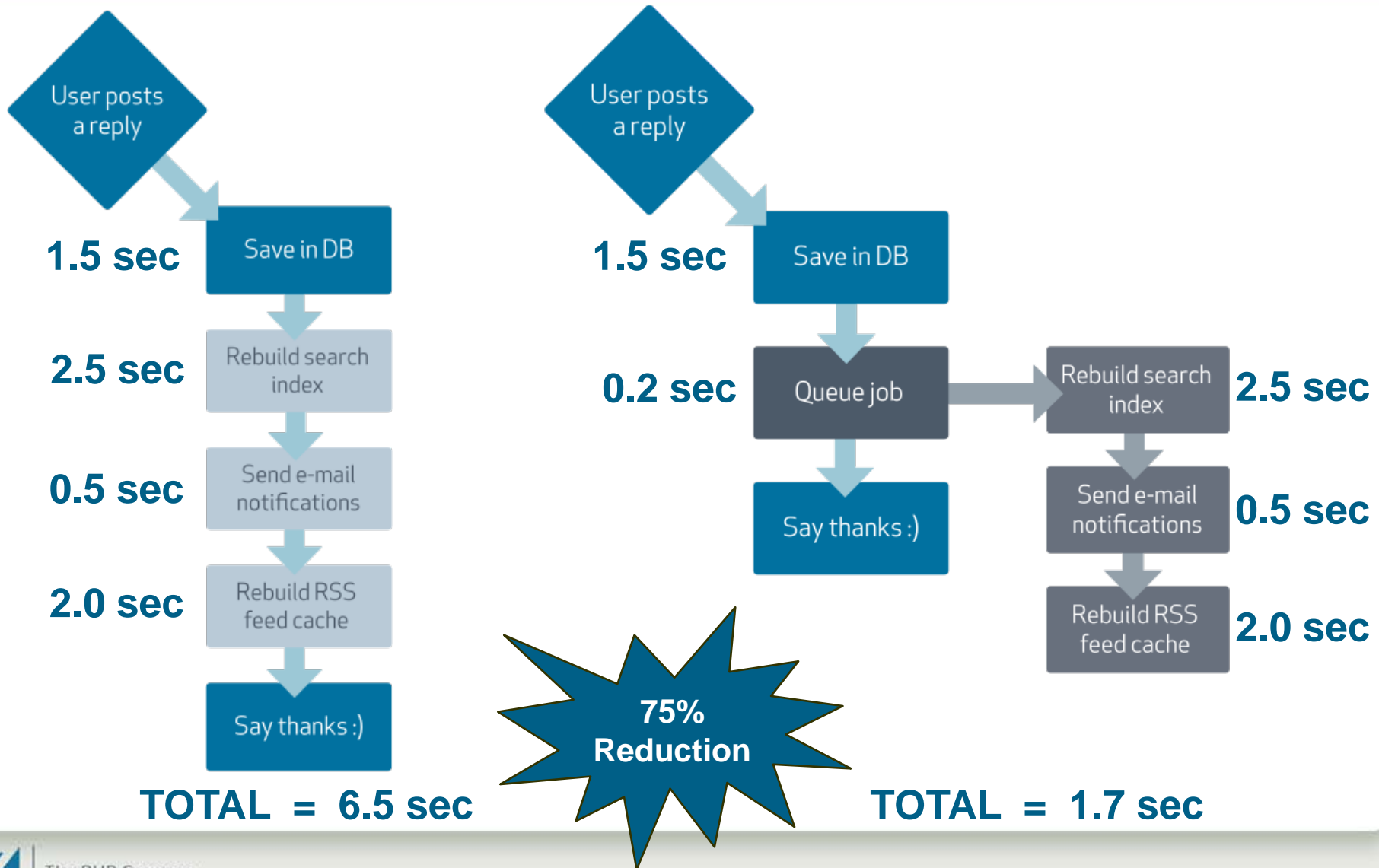
# Job Queue

- Job Queue allows for asynchronous execution of PHP scripts (Jobs)
- It streamlines offline processing and improves application performance

## Examples:

- Run a time-consuming report generation script on a different server, or at night
- Index a site for the Search feature
- Send emails for new posts
- Update RSS feeds
- Specify times for the execution of scripts (Ex: periodic database cleanup)

# Job Queue Example



# Creating Jobs

- Jobs are created using the `createHttpJob()` method

```
$queue = new ZendJobQueue();  
$queue->createHttpJob(  
    'http://backend.local/jobs/somejob.php' );
```

- Pass parameters
  - Simple ones, via query string `$_GET`
  - Complex ones, as an associative array of key => value pairs
- Set job options
  - Name, priority, schedule, etc.
  - Create deferred or recurring jobs
- Or... create from the UI



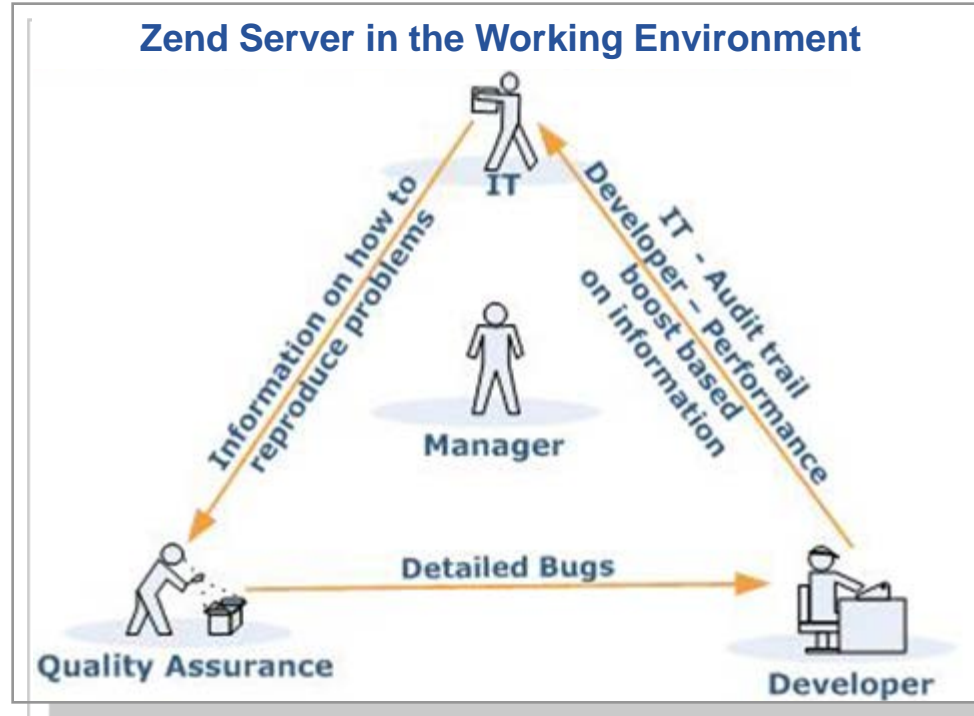
The PHP Company

# Best Practices

# Course Wrap Up - Best Practices

- We will now take a look at some established best practices for this new product
  - Development
  - Deployment
  - Performance
  - Security
  - General Troubleshooting
- More recommendations will surely follow as Zend Server continues to increase its broad base of users

# Problem Resolution Lifecycle



The Lifecycle's main goal is to help ease the strain of challenging environments and tight schedules while preventing problematic issues from falling between the cracks

# Server Best Practices: Development

## Zend Framework (ZF)

- The integration of ZF with Server provides a pre-configured stack, including all system components, for developing PHP applications
- ZF files are housed in:
  - **Windows:** `<install_path>\share\ZendFramework`
  - **RPM, DEB, Tarball, Mac:** `<install_path>/share/ZendFramework`
- ZF classes can be loaded either using the Zend Loader, or by using `require` / `include` calls
- Configure Server for ZF using port-based virtual hosts
  - Advantage of a port-based virtual host is the ease with which isolated applications can be run on the same web server
  - Allows developers working on ZF projects in Zend Studio to immediately test any code changes locally

# Server Best Practices: Development

## Advanced Diagnostics using Zend Server

- We have already covered the concept of Events, generated by rules set in Rules Management, tracked and analyzed using Monitor
- Here are some guidelines around utilizing more advanced diagnostic techniques:
  - Be sure to use Code Tracing
  - Determine if the issue is genuine
    - Is it due to the Monitor Rule parameters (thresholds, functions list), or is it real?
    - Can the Monitor API be used to solve it? (ID the problem so it can be ignored)
    - Is the behavior acceptable for a certain set of circumstances
    - Don't forget to manage issues by changing their status - the **Ignored** status will allow the events to continue to be monitored, but will be ignored
- If the issue is genuine, the Reference Manual provides insight around rules, to help diagnose these problems



# Server Best Practices: Performance

Here are some general recommendations to enhance performance:

1. Only install components that you will be using
  - **Server Setup | Components**
  - Example: Java Bridge
2. Turn off components when not needed
  - Debugger ON when in development phase of Software Lifecycle; OFF in production
  - Keep Monitor running, except during performance testing
  - Guard Loader OFF unless running PHP code encoded by Zend Guard
  - Page Cache OFF if you are not using URL-based caching (similarly for Data Cache)

# Server Best Practices: Performance

## General recommendations (continued)

### 2. Turn off components when not needed (continued)

- Keep Optimizer+ running
  - Can fine tune, if needed
    - Code Change Auto-Detection
    - Code Validation Frequency

### 3. Configure PHP

- It is possible to edit some directives to enhance performance
- **Server Setup | Directives** (*see manual*)

# Server Best Practices: Troubleshooting

## Zend Server Exception Caught

- Installing Zend Server with a bundled Apache assumes that the following port settings are used:
  - The Web server (Apache) is listening on port 80
  - The Zend Server Administration Interface is listening on 10081,10082
- If you receive a "**Zend Server Exception Caught**" error message when trying to access the Administration Interface, make sure that Apache is up and running
- Linux and Mac: When using https, the port is set by default to 10082, and an "invalid certificate" error will display across all browsers
  - Resolve it by making it an Allowed Exception
  - This is not a security issue

# Server Best Practices: Troubleshooting

## Windows: Zend Server is not loading

- If you cannot load Zend Server, or one of the Zend Server related processes causes a crash or unexpected system behavior, use the installer in Repair mode
  1. Run the installer file or go to **Start | Settings| Control Panel | Add or Remove Programs | Zend Server** and select **Modify** to run the installer
  2. Click **Next** to complete the repair process and **Finish** to close the Installer

# Server Best Practices: Troubleshooting

## Windows: Internet Explorer is blocking Zend Server

- Applies only to IE 7 running on the Windows 2008 Server
- **Possible cause:** IE is interpreting Server as a security threat
- **Solution:** Add Zend Server to the "trusted sites" zone

Thank you