PHP: Toolkit Examples

Mike Pavlak
Solutions Consultant
mike.p@zend.com
(815) 722 3454

The Omni User
An Association of Chicagoland Computer Professionals
Audience

- New to PHP
- Looking to leverage IBM i resources in web applications
- Seeking business value of PHP
Agenda

• Review Zend Server: Home of the Toolkit

• Overview of toolkit functions
  ‣ Studio Integration

• Examples
  ‣ Data Area
  ‣ Data Queue
  ‣ Program Call

• Explore 5250 bridge
  ‣ Bridge functions
  ‣ Updated emulator

• ITK - A new way to do IBM i functions
Questions?

• Let’s keep it interactive!

• Follow us!

  ‣ [Facebook](http://bit.ly/cjueZg) (Zend Technologies or search for Zend)

  ‣ [Twitter](http://twitter.com/zend)
PHP: Toolkit Examples

www.zend.com

Review Zend Server
Review of Zend Server

• Variables & Data Types
• Functions
• Copybooks
Zend Server 5.0 GA on 04/12/2010!

- The Linux and Windows distributions have been GA since Feb
- Code base consistency is making this gap smaller
- Training class available...see zend.com/store for more info
What Is Zend Server?

• Production quality PHP stack
  ▸ PHP, ZF, DB connectivity, debugging extension, and more

• Two Editions - Free “Community Edition” provided by IBM & Full commercial edition
  ▸ Both are production ready

• Application monitoring and diagnostics (integrated with Zend Studio)

• Multi-level performance enhancement capabilities

• Software updates and security hot fixes

• Easy and quick installation

• Zend Framework Integration
Zend Server for IBM i

• Next generation of PHP stack for IBM i
• Best of both Zend Core and Zend Platform
• Single Licensed Program Installation
• Two products
  ‣ Zend Server for IBM i Community Edition
    • Available at no charge per IBM partnership
  ‣ Zend Server for IBM i
    • Subscription available from Zend
    • High value extra features
    • Higher Support SLAs
Why a new generation?

• Based on feedback of key areas for Improvement

• Improved basic PHP Performance
  ▸ 30% to 600% depending on the application

• New capabilities
  ▸ Code Tracing - Faster problem resolution
  ▸ Job Queue - Performance and scalability
  ▸ Page Caching - Performance and Scalability
  ▸ Support for PHP 5.3

• Easier installation, administration & maintenance
  ▸ Single apache server
Zend Server Under the Covers

ILE Apache:10088
• Default configuration
FastCGI

IBM i
i/OS
PASE

HTTP:10088 Server (FastCGI)

PHP CGI

Zend Server

DB2 UDB

PHP file

*PGM
CMD

URL Request

HTML
Overview of Toolkit Functions
i5 Toolkit APIs

- Are shipped with Zend products
  - Zend Server for IBM i
  - Documented in the Zend Server User Guide
- Geared for accessing DB2 data & IBM I resources from PHP
  - Simplifies modern application integration with legacy data and applications
- Note: The i5_COMD job must be running in the ZENDSVR subsystem
  - Use WRKACTJOB SBS(ZENDSVR) to see i5_COMD job
  - Use the Zend menu to start the i5_COMD job if not started:
    - GO ZENDSVR/ZSMENU
    - Option 5 (Service Management menu)
    - Option 8 (Start I5_COMD service)
Zend Server Menu

Select one of the following:

1. Change password for Web Administration Console
2. Update using Zend Server PTFs menu
3. Run Support Tool
4. Service Management menu
5. MySQL Management menu
6. 5250 Bridge Management Menu
7. Reset Zend Server environment
8. Signoff

Selection or command

F3=Exit  F4=Prompt  F9=Retrieve  F12=Cancel  F23=WRKUSRJOB

1902 - Session successfully started
Zend Server Subsystem

![Session A - [24 x 80]](image)

CPU %: 0.0 Elapsed time: 00:00:00 Active jobs: 278

Type options, press Enter.
2=Change 3=Hold 4=End 5=Work with 6=Release 7=Display message 8=Work with spooled files 13=Disconnect ...

<table>
<thead>
<tr>
<th>Opt</th>
<th>Subsystem/Job</th>
<th>User</th>
<th>Type</th>
<th>CPU %</th>
<th>Function</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ZENDSVR</td>
<td>QSYS</td>
<td>SBS</td>
<td>0.0</td>
<td>DEQW</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>COMD</td>
<td>QTCP</td>
<td>BCH</td>
<td>0.0</td>
<td>PGM-EASYCOMD</td>
<td>TIMW</td>
</tr>
<tr>
<td></td>
<td>ZSJOBQMQNG</td>
<td>QTMMHTTP</td>
<td>BCI</td>
<td>0.0</td>
<td>PGM-watchdog</td>
<td>THDW</td>
</tr>
<tr>
<td></td>
<td>ZSJOBQMQNG</td>
<td>QTMMHTTP</td>
<td>BCI</td>
<td>0.0</td>
<td>PGM-jqd</td>
<td>SELW</td>
</tr>
<tr>
<td></td>
<td>ZSMONMNG</td>
<td>QTMMHTTP</td>
<td>BCI</td>
<td>0.0</td>
<td>PGM-watchdog</td>
<td>THDW</td>
</tr>
<tr>
<td></td>
<td>ZSMONMNG</td>
<td>QTMMHTTP</td>
<td>BCI</td>
<td>0.0</td>
<td>PGM-MonitorNode</td>
<td>SELW</td>
</tr>
</tbody>
</table>

Parameters or command

F3=Exit  F5=Refresh  F7=Find  F10=Restart statistics
F11=Display elapsed data  F12=Cancel  F23=More options  F24=More keys

I902 - Session successfully started
i5 Toolkit APIs

• Connection management
  - i5_connect
  - i5_close
  - i5_adopt_authority
  - i5_error
  - i5_errno
  - i5_errnomsg

• Data retrieval
  - i5_fetch_array
  - i5_fetch_assoc
  - i5_fetch_object
  - i5_fetch_row
  - i5_info
  - i5_field_len
  - i5_field_name
  - i5_field_scale
  - i5_field_type
  - i5_list_fields
  - i5_num_fields
  - i5_result

• Command calls
  - i5_command

• Program calls
  - i5_program_prepare
  - i5_program_prepare_PCML
  - i5_program_call
  - i5_program_close

• Native file access
  - i5_open
  - i5_addnew
  - i5_edit
  - i5_delete
  - i5_cancel_edit
  - i5_setvalue
  - i5_update
  - i5_range_from
  - i5_range_to
  - i5_range_clear
  - i5_data_seek
  - i5_seek
  - i5_bookmark
  - i5_free_file
  - i5_new_record
  - i5_update_record
i5 Toolkit APIs

• System values
  • i5_get_system_value

• Data areas
  • i5_data_area_prepare
  • i5_data_area_receive
  • i5_data_area_send
  • i5_data_area_close

• Job logs
  • i5_jobLog_list
  • i5_jobLog_list_read
  • i5_jobLog_list_close

• Active jobs
  • i5_job_list
  • i5_job_list_read
  • i5_job_list_close

• Objects list
  • i5_object_list
  • i5_object_list_read
  • i5_object_list_close

• Print/Get spooled file
  • i5_spool_list
  • i5_spool_list_read
  • i5_spool_list_close
  • i5_spool_get_data
  • i5_spool_from_file

• User space
  • i5_userspace_create
  • i5_userspace_prepare
  • i5_userspace_get
  • i5_userspace_put
Connection

- Must connect to IBM i system to use any of the toolkit APIs
  - Connecting (i5 for i5 functions, db2 for DB2 functions)
    - i5_connect
      - Need system, user profile name, and password as parameters
      - Can set library list here
    - i5_close
      - Always close a connection you’ve opened
    - i5_adopt_authority
      - Can adopt other authorities while running
  - Error handling
    - i5_error
      - Get data about an error
    - i5_errno
      - Error number
    - i5_errormsg
      - Error message
PHP: Toolkit Examples

www.zend.com

Examples
Two toolkit examples

- Data Area
- System Values
- Program call
- Spooled File Access
Data Area Contents

Display Data Area

Data area: COMPANY
Library: ZENDDATA
Type: *CHAR
Length: 50
Text: Company Name

Value
Offset  Value
0      'Zend Technologies, Inc.'

Press Enter to continue.

F3=Exit  F12=Cancel

Disconnect

© All rights reserved. Zend Technologies, Inc.
<?php

include("i5db2connectonly.php");

$heading='test';

$heading = i5_data_area_read("ZENDDATA/COMPANY");
if (!$heading)
    die("<br>data area read failed.");

echo "<h1>" . $heading . "</h1>";

echo "Heading = $heading";
i5_close($conn);

?>
Data Area Output

Zend Technologies, Inc.

Heading = Zend Technologies, Inc.
System Values Code Part 1

```php
<html> <head><title>Customer Insert</title></head><body>

<h1>Main Computer Information</h1>

<?php

include("i5db2connectonly.php");

//if ($conn)
//die("<br>Connection failed. Error number = " . i5_errno() . " msg = " . i5_errormsg());

//Define table and start filling rows...

echo '<TABLE BORDER="10" BORDERCOLOR="red"><TBODY BGCOLOR="00FF00">';

echo '<TR><TD><B>Description</TD><TD><B>System Value</TD></TR>';

print "<TR><TD<Model></TD><TD>" . i5_get_system_value("QMODEL") . "</TD></TR>";

print "<TR><TD>Serial</TD><TD>" . i5_get_system_value("QSRLNBR") . "</TD></TR>";
```

System Value Code Part 2

```php
34    print "<TR><TD>Minimum Password Length</TD><TD>" . i5_get_system_value("QPWDMINLEN") . "</TD></TR>";
35    print "<TR><TD>Maximum Password Length</TD><TD>" . i5_get_system_value("QPWDMAXLEN") . "</TD></TR>";
36    print "<TR><TD>Auto Config</TD><TD>";
37    
38    if(i5_get_system_value("QAUTOCFG") == 1) {
39        echo "On";
40    } else {
41        echo "Off";
42    }
43    print "</TD></TR>";
44
45    echo '</table>';  
46    i5_close($conn);
47    ?>
```
### Main Computer Information

<table>
<thead>
<tr>
<th>Description</th>
<th>System Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>520</td>
</tr>
<tr>
<td>Serial</td>
<td>10B95CE</td>
</tr>
<tr>
<td>Processor Feature Code</td>
<td>7736</td>
</tr>
<tr>
<td>Date</td>
<td>042609</td>
</tr>
<tr>
<td>Time</td>
<td>100258242</td>
</tr>
<tr>
<td>Code Character SetID</td>
<td>37</td>
</tr>
<tr>
<td>Controlling Subsystem</td>
<td>QBASE QSYS</td>
</tr>
<tr>
<td>Start Up Program</td>
<td>QSTRUP QSYS</td>
</tr>
<tr>
<td>Minimum Password Length</td>
<td>6</td>
</tr>
<tr>
<td>Maximum Password Length</td>
<td>8</td>
</tr>
<tr>
<td>Auto Config</td>
<td>On</td>
</tr>
</tbody>
</table>
Program Call...

- Most popular use of the API Toolkit
- Can call ANY IBM i program
- Our example will show CL
- Three parameters
- Concatenate a string

Steps:
- Parameter setup
- Prepare the program
- Load the i/o array of parameters & map return values
- Call the program
- Retrieve values
Program Call, script...Parameters

<h1>Village Water Billing System</h1>
<h2>Program Call</h2>

```php
<?php

include("i5db2connectlib.php");

// Setup parameters in associative array...
$desc = array ( 
  array ("name"=>"string1", "io"=>I5_INOUT, "type" => I5_TYPE_CHAR, "length"=> "10"),
  array ("name"=>"string2", "io"=>I5_INOUT, "type" => I5_TYPE_CHAR, "length"=> "10"),
  array ("name"=>"string3", "io"=>I5_INOUT, "type" => I5_TYPE_CHAR, "length"=> "20"),
);
```
// Prepare the program, similar to prototype in ILE...
$ prog = i5_program_prepare("clp1", $desc);
if ($prog === FALSE)
{
    $errorTab = i5_error();
    echo "Program prepare failed <br>";
    var_dump($errorTab);
    die();
}
Program Call, script...Load parms

// Load parameters...
$params = array("string1"=>"The quick ","string2"=>"brown fox ","string3"=>" ");

// Map parameters to variables...
$retvals = array("string1"=>"string1","string2"=>"string2","string3"=>"string3");
Program Call, script...Execute!

```php
$ret = i5_program_call($prog, $params, $retvals);
if ($ret === FALSE)
{
    $errorTab = i5_error();
    echo "FAIL : i5_program_call failure code <br>";
    var_dump($errorTab);
    die();
}
```
Program Call, example...

Village Water Billing System

Program Call

The return values are:
String 1: The quick
String 2: brown fox
String 3: The quick brown fox
Spooled file listing...

- Everybody has them
- Used to print miles of standard reports
- Many utilities to make them “pretty”
- List them
- Maybe do something more
Spooled file listing...

Spooled File

i5_spool_list

resource i5_spool_list([array description][, resource connection])

Create an spool file lists, of certain output queue or for all queues.

Returns: resource if OK, false if failed

Arguments:

<table>
<thead>
<tr>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>description</strong></td>
<td>The data by which the spool files will be filtered, array with following keys:</td>
</tr>
<tr>
<td></td>
<td>• username - username that created the job</td>
</tr>
<tr>
<td></td>
<td>• outq - qualified name for the output queue containing the spool file</td>
</tr>
<tr>
<td></td>
<td>• userdata - the user-supplied key data for the spool file</td>
</tr>
<tr>
<td></td>
<td>All keys are optional and can be provided together</td>
</tr>
<tr>
<td><strong>connection</strong></td>
<td>Connection - result of i5_connect.</td>
</tr>
</tbody>
</table>
Spooled file listing...

Village Water Billing System

Spooled Files for MPAVLAK

<table>
<thead>
<tr>
<th>Job Name</th>
<th>User Name</th>
<th>File Name</th>
<th>Date</th>
<th>OUTQ</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSP04</td>
<td>MPAVLAK</td>
<td>QPEZBCKUP</td>
<td>1070121</td>
<td>MPAVLAK</td>
<td>1</td>
</tr>
<tr>
<td>DSP04</td>
<td>MPAVLAK</td>
<td>QPEZBCKUP</td>
<td>1070121</td>
<td>MPAVLAK</td>
<td>2</td>
</tr>
<tr>
<td>DSP04</td>
<td>MPAVLAK</td>
<td>QSYSRTR</td>
<td>1070121</td>
<td>MPAVLAK</td>
<td>3</td>
</tr>
<tr>
<td>DSP04</td>
<td>MPAVLAK</td>
<td>QPDSPNET</td>
<td>1070121</td>
<td>MPAVLAK</td>
<td>4</td>
</tr>
<tr>
<td>DSP04</td>
<td>MPAVLAK</td>
<td>QSYSRTR</td>
<td>1070121</td>
<td>MPAVLAK</td>
<td>10</td>
</tr>
<tr>
<td>DSP04</td>
<td>MPAVLAK</td>
<td>QSYSRTR</td>
<td>1070121</td>
<td>MPAVLAK</td>
<td>67</td>
</tr>
<tr>
<td>DSP04</td>
<td>MPAVLAK</td>
<td>QPUSRPRF</td>
<td>1070121</td>
<td>MPAVLAK</td>
<td>96</td>
</tr>
<tr>
<td>MPAVLAKA</td>
<td>MPAVLAK</td>
<td>QPDSPLIB</td>
<td>1070127</td>
<td>MPAVLAK</td>
<td>1</td>
</tr>
<tr>
<td>MPAVLAKA</td>
<td>MPAVLAK</td>
<td>QSYSRTR</td>
<td>1070221</td>
<td>MPAVLAK</td>
<td>1</td>
</tr>
<tr>
<td>MPAVLAKA</td>
<td>MPAVLAK</td>
<td>QPSRDMP</td>
<td>1070620</td>
<td>QEZDEBUG</td>
<td>1</td>
</tr>
</tbody>
</table>
Spooled file listing...

- Need some particulars
  - User
  - Output Queue
  - User data
- Break code and conquer!
  - Create the handle
  - Read through the table
Spooled file listing...

```php
<?php
$userName="MPAVLAK";
echo "<h2>Spooled Files for ",$userName,"</h2>";
include("i5db2connectonly.php");

$HdlSpl = i5_spool_list(array(I5_USERNAME=>$userName));
if (!$HdlSpl){
    $ret = i5_errno();
    print_r($ret);
}

echo '<TABLE BORDER="10" BORDERCOLOR="red"><TBODY BGCOLOR="00FF00">';
echo '<TR><TD><B>Job Name</TD><TD><B>User Name</TD><TD><B>File Name</TD><TD><B>Date</B></TR>
```
```
Spooled file listing...

```php
$continue=true;
while ($continue){
    $ret = i5_spool_list_read($HdlSpl);
    if (!$ret){
        $erreur = i5_error();
        if ($erreur['num'] != 14){
            // error code here...
        }
        $continue = false;
        break;
    }
}

<tr>
<td width="20%"><?php echo $ret['JOBNAME']; ?> </td>
<td width="20%"><?php echo $ret['USERNAME']; ?> </td>
<td width="20%"><?php echo $ret['SPLFNAME']; ?> </td>
<td width="20%"><?php echo $ret['OUTQNAME']; ?> </td>
<td width="20%"><?php echo $ret['SPLFNBR']; ?> </td>
</tr>
```
PHP: Toolkit Examples

www.zend.com

5250 Bridge & Emulator
What is the "Bridge"

- Three sets of APIs that allow a PHP script to drive a 5250 session
  - Green Screen Simulator
  - Object Oriented APIs
  - Procedural APIs
    - During the session these are the ones we will be focusing on
- Multiple 5250 sessions can be connected to a single script
  - So you could combine the data from several green screen applications in a single browser window!
- Bridge is available as part of Zend Platform App Server
Product Definition

- The 5250 Bridge is a PHP based solution for running interactive applications in the i5/OS environment.
1. PHP sends a ‘start session’ request to the 5250 Program using the 5250 Bridge.
2. The 5250 Bridge returns a confirmation to PHP that the session has started.
3. The 5250 Bridge sends current screen information in XML format to PHP (e.g. format name, input and output fields and current cursor position).
4. The input values can be automatically entered into the program or sent on in an html format for user input.
ARRAY DUMP OF RETURN SET

1. array(5)
2. {
   [0]= array(8) { ["id"]= int(0) ["row"]= int(6)
3. ["column"]= int(53) ["length"]= int(10) ["value"]=
4. string(10) " " ["type"]= string(11) "Alpha shift" ["font"]=
5. string(11) "Not defined" ["format"]= string(11) "Not Defined" }
6. [1]= array(8){ ["id"]= int(1) ["row"]= int(7)
7. ["column"]= int(53) ["length"]= int(10) ["value"]=
8. string(10) " " ["type"]= string(11) "Alpha shift" ["font"]=
9. string(11) "Not defined" ["format"]= string(11) "Not Defined" }
10.[2]= array(8) { ["id"]= int(2) ["row"]= int(8)
11.["column"]= int(53) ["length"]= int(10) ["value"]= string(10) " " ["type"]= string(11) "Alpha shift"
"font"]=
12.string(11) "Not defined" ["format"]= string(11) "Not Defined" }
13.[3]= array(8) { ["id"]= int(3) ["row"]= int(9)
14.["column"]= int(53) ["length"]= int(10) ["value"]=
15.string(10) " " ["type"]= string(11) "Alpha shift" ["font"]=
16.string(11) "Not defined" ["format"]= string(11) "Not Defined" }
17.[4]= array(8) { ["id"]= int(4) ["row"]= int(10)
18.["column"]= int(53) ["length"]= int(10) ["value"]= string(10) " " ["type"]= string(11) "Alpha shift"
"font"]=
19.string(11) "Not defined" ["format"]= string(11) "Not Defined" }

PHP: Toolkit Examples
A bit about me

• Kevin Schroeder
  ▸ Technology Evangelist - Zend Technologies
  ▸ Blog: http://www.eschrade.com
  ▸ Email: kevin@zend.com
ITK

• What is it?
  ▶ An easy to use interface into the i5 PHP Toolkit
  ▶ Fully unit tested

• Why use it?
  ▶ It drastically simplifies accessing RPG functionality in PHP

• What’s it cost?
  ▶ Nothing

• Where do I get it?
  ▶ http://www.github.com/...
How does one currently access RPG functionality in PHP?
How can one now access RPG functionality in PHP?
Concepts

- **Adapters**
  - Represents the interface to the functionality be accessed

- **Programs**
  - Represents the program definition

- **Unit Tests**
  - Mechanisms for testing whether or not program functions are properly handling RPG-based data
Adapters

• Two adapters are included
  ▹ Live
    • Represents an individual connection to the PHP Toolkit implementation
    • Used in production scenarios
  ▹ Mock
    • Used for Unit Testing
    • Will return specific results for specific calls
    • Allows you to test that your class is properly handling data

• A Program class can be called regardless of the adapter being used
Programs

• Based on the Itk_PgmAbstract class

• Definitions required

  ▸ Program Name
    • The name of the RPG program being called
      protected $_programName = 'ZENDSVR/COMMONPGM';

  ▸ Description
    • Describes the input/output parameters
      protected $_description = array(
        'CODE' => array(
          self::DESC_IO => I5_INOUT,
          self::DESC_TYPE => I5_TYPE_CHAR,
          self::DESC_LENGTH => "10"
        )
      );
Unit Testing

• The entire library is designed to be easily testable
  ▸ Repeatability in testing is KEY to not losing your hair
  ▸ Well written tests ensure that changes to not break compatibility

• Unit Tests can be run using
  ▸ Live adapter on i Series machines
  ▸ Mock adapter on either
    • i Series machines (if you don’t want to change data)
    • Linux/Windows workstations

• Unit Tests are run using PHPUnit
Recommendations

• Use a bootstrap to define the adapter at the front of the request

• Keep as much code out of your HTML as possible

• If you are not familiar with OOP this is a good place to start
  ▶ Small example
  ▶ Follows good practices
  ▶ Demonstrates the benefits on a small scale
New book, new printing, same great stuff!

Kevin Schroeder from Zend’s Global Services Group  
with  
Jeff Olen, co-author of...

Get yours at MCPressonline  
or at fine bookstores everywhere
Resources

• Recorded Webinars

• Zend Server for IBM i main page, link to downloads

• Zend Server manual:
Thank you & Q&A

• Follow us!

  ▶ http://bit.ly/cjueZg (Zend Technologies or search for Zend)

  ▶ http://twitter.com/zend
Q&A

www.zend.com

mike.p@zend.com

Please fill out your Session Evaluation!