PHP Batch Jobs on IBM i

Alan Seiden
PHP on IBM i consultant/developer

email: aseiden@sbsusa.com
blog: http://alanseiden.com
Strategic Business Systems, Inc.

• Developing Web apps on IBM i (and iSeries, i5...) since 1996
• Based in Ramsey, New Jersey

• IBM Business Partner
  ▪ Power Systems hardware, software, development, consulting

• Zend Business Partner
  ▪ Working with Zend since they brought PHP to IBM i in 2006
  ▪ We offer Zend’s training and Zend Server software to complement our consulting/development services
Alan Seiden

**PHP on IBM i Developer / Consultant / Mentor**

- Contributor to IBM’s Redbook *PHP: Zend Core for i5/OS*
- First IBM i developer certified in Zend Framework
- Developer of IBM/COMMON’s “Best Web Solution” of 2009

Contact: aseiden@sbsusa.com or 201-327-9400
Blog/articles: http://alanseiden.com
What we’ll discuss today

- Quick overview, Zend Server for IBM i
- PHP for Batch (non-web) Tasks on IBM i
  - “batch” = command line or scheduled PHP
- Tips and techniques
- Questions
PHP/web
PHP was built for server-side web apps

- Started as a web development language in 1995
- Over time, the open source community and Zend made PHP more and more powerful
- Currently one of the most popular web languages
  - It’s everywhere, eBay, Wikipedia, Facebook…
  - But it’s not limited to the web
  - It would be a shame to restrain PHP’s power to only the web

- On IBM i, all of PHP’s power is available in Zend Server (formerly Zend Core).
Zend Server
Zend Server for IBM i

• **Next generation PHP stack for IBM i**
  ▪ Best of Zend Core and Platform in one licensed program

• **Two license levels**
  ▪ **Zend Server for IBM i Community Edition (CE)**
    • Available at no charge per IBM partnership
    • Includes “Optimizer+” that speeds up code
    • One year silver (email) support
  
  ▪ **Zend Server for IBM i**
    • Subscription-based license
    • High value extra features, higher Support SLAs

• **Details of differences:**
Installation

- **http://www.zend.com/products/server/downloads**
  - Click on the “IBM i” tab

- **Prerequisites**
  - v5r4 and:
    - IBM’s FastCGI PTF (free)

- **Try Zend Studio’s IDE (no charge, courtesy of IBM)**
  - “Zend Studio for Eclipse, IBM i Edition”
  - Look for “IBM i Edition”
• **Zend Server combines “Core” and “Platform” controls in one interface**

• **Simpler Apache setup**
  - Single web server
    - PASE server eliminated
  - IBM HTTP Server Powered by Apache
    - PHP implemented using IBM’s Fast CGI
  - Clean, more complete graphical user interface
    - Favorite: the Restart button
Admin menu (GO ZENDSVR/ZSMENU)

Zend Server for IBM i Setup 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
Zend Server’s Interface

Recent Events

<table>
<thead>
<tr>
<th>Id</th>
<th>Generated by Rule</th>
<th>Count</th>
<th>Last Occ.</th>
<th>Severity</th>
<th>Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>000001</td>
<td>Severe Slow Query Execution</td>
<td>1</td>
<td>05-Jan 06:22</td>
<td>Critical</td>
<td>db2_exec</td>
</tr>
<tr>
<td>000002</td>
<td>Severe Slow Request Execution (Absolute)</td>
<td>1</td>
<td>05-Jan 06:22</td>
<td>Critical</td>
<td><a href="http://192.168.15.203:10098/mpavelk/customers/partnerroads/showzend/1">http://192.168.15.203:10098/mpavelk/customers/partnerroads/showzend/1</a></td>
</tr>
<tr>
<td>000003</td>
<td>PHP Error</td>
<td>4</td>
<td>05-Jan 06:28</td>
<td>Warning</td>
<td><a href="http://192.168.15.203:10098/mpavelk/customers/partnerroads/showzend/1">http://192.168.15.203:10098/mpavelk/customers/partnerroads/showzend/1</a></td>
</tr>
<tr>
<td>000004</td>
<td>PHP Error</td>
<td>7</td>
<td>05-Jan 06:28</td>
<td>Warning</td>
<td><a href="http://192.168.15.203:10098/mpavelk/customers/partnerroads/showzend/1">http://192.168.15.203:10098/mpavelk/customers/partnerroads/showzend/1</a></td>
</tr>
<tr>
<td>000005</td>
<td>PHP Error</td>
<td>7</td>
<td>05-Jan 06:28</td>
<td>Warning</td>
<td><a href="http://192.168.15.203:10098/mpavelk/customers/partnerroads/showzend/1">http://192.168.15.203:10098/mpavelk/customers/partnerroads/showzend/1</a></td>
</tr>
</tbody>
</table>

System Overview

- PHP Version: 5.2.11
- Zend Framework Version: 1.9.6

Zend Server

- Zend Code Tracing: ON
- Zend Data Cache: ON
- Zend Debugger: ON
- Zend Guard Loader: OFF
- Zend Java Bridge: OFF
- Zend Job Queue: OFF
- Zend Monitor: OFF
- Zend Optimizer+: OFF
- Zend Page Cache: OFF

Tasks

- Learn how to start with Zend Server and PHP
- Configure Zend Server Extensions

Your license will expire in 29 days. Click here to update your license.
## Cheat sheet for upgrade


<table>
<thead>
<tr>
<th></th>
<th>Zend Core</th>
<th>Zend Server</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Installation folder</strong></td>
<td>/usr/local/zend/core</td>
<td>/usr/local/zendsvr</td>
</tr>
<tr>
<td><strong>PHP.INI</strong></td>
<td>/usr/local/zend/core/etc</td>
<td>/usr/local/zendsvr/etc</td>
</tr>
<tr>
<td><strong>Web server root(s)</strong></td>
<td>/www/zendcore, /usr/local/zend/apache2</td>
<td>/www/zendsvr</td>
</tr>
<tr>
<td><strong>Document root</strong></td>
<td>/www/zendcore/htdocs</td>
<td>/www/zendsvr/htdocs</td>
</tr>
<tr>
<td><strong>Zend Framework</strong></td>
<td>/usr/local/Zend/ZendFramework</td>
<td>/usr/local/zendsvr/share/ZendFramework</td>
</tr>
<tr>
<td><strong>PHP binaries folder</strong></td>
<td>/usr/local/zend/core/bin</td>
<td>/usr/local/zendsvr/bin</td>
</tr>
<tr>
<td><strong>PHP Log files</strong></td>
<td>/usr/local/zend/core/logs</td>
<td>/usr/local/zendsvr/var/log</td>
</tr>
<tr>
<td><strong>Web user profile (assign authority to it)</strong></td>
<td>NOBODY</td>
<td>QTMHHTTP</td>
</tr>
<tr>
<td><strong>Default HTTP Port</strong></td>
<td>:89</td>
<td>:10088</td>
</tr>
<tr>
<td><strong>Admin Interface URL</strong></td>
<td><a href="http://yourIBMi:89/ZendCore/">http://yourIBMi:89/ZendCore/</a></td>
<td><a href="http://yourIBMi:10088/ZendServer/">http://yourIBMi:10088/ZendServer/</a></td>
</tr>
</tbody>
</table>
Zend Server and PHP resources

• **Support:** Zend Server CE includes one year of email support. Upgrade to Zend Server for more years and fast phone support
  - Register at [http://www.zend.com/products/server/downloads](http://www.zend.com/products/server/downloads) and also get Zend’s IBM i newsletter

• **Web:**
  - Zend’s recorded webinars: [http://www.zend.com/resources/webinars/i5-os](http://www.zend.com/resources/webinars/i5-os)
  - Zend Forums: forums.zend.com (look for IBM i Zend Server forums)
  - [http://alanseiden.com](http://alanseiden.com) (of course!)

• **Books (online and print)**
  - PHP on IBM i book by Jeff Olen & Kevin Schroeder (MC Press)

• **Training from Zend/others**
PHP beyond the web
I first used “batch PHP” on an EDI project

• Generate an XML-based price list and FTP it to customer each week at the same time
• Poll for and receive XML orders arriving in a folder
  ▪ Process the XML, generate a new XML acknowledgement, and then FTP a response
• Send e-mail reports to administrators
• Special e-mails if any error occurred
Why use PHP for non-web tasks?

• **PHP easily handles:**
  - IFS stream files
  - manipulation of any string data and common formats such as XML and JSON
  - communication with other processes through TCP/IP, whether on your network or on the internet

• **Leverage your PHP skills instead of learning yet another language**
  - And have another opportunity to **improve** your PHP skills!

• **Share functions between web and non-web**
Batch PHP is useful beyond the IBM i

- I once needed to change dates of thousands of folders on a Windows XP system

- I could have done it with a complex DOS batch file, or a “WSH” Windows shell script

- But I already knew PHP and could find many of the code pieces online

- You can download a Windows version of PHP from [http://windows.php.net/](http://windows.php.net/), XAMPP, or WAMP
Examples of what you can do with PHP on IBM i

- Download, upload, process files from web and FTP servers
- Send e-mails with great flexibility
- Create graphics
- Read and write from your regular db2 files
- Schedule these tasks to occur when you want
Two methods to do this

• **Method #1: “PHP-CLI” (command line)**
  - Test on command line, embed in CL/RPG
  - Automate the process with scheduled CL programs
  - It’s not difficult for IBM i professionals
    • Need some knowledge of PASE or QSHELL environment
  - This is how I’ve always done it

• **Method #2 (NEW): Zend Server’s Job Queue**
  - Better for people not familiar with command line OR
    needing more flexibility, dynamic scheduling
  - accepts complex parms, multi-dimensional arrays, objects
  - PHP professionals may prefer this method
  - Included in Zend Server Professional Edition
Method #1: PHP-CLI
CLI = Command Line

- **PHP-CLI is a simplified PHP executable**
  - Location in Zend Server: /usr/local/zendsvr/bin/php-cli

- **It’s optimized for the command line**
  - It doesn’t look for, or have access to, web server variables or HTTP headers
  - It assumes that you will use command line arguments ($argc and $argv) rather than web server $_GET and $_POST arrays
PHP-CLI doesn’t need a web server to run

- Apache web server does not have to be running

- Scripts needn’t be in a web-accessible location
  - In fact, for security, you should put them anywhere BUT the web server root, unless you also want the scripts to be run in a browser.

  - That means don’t put them in /www/zendsvr/htdocs

  - I like a location such as /php/appname/myscript.php
    - Note: no “www” in the path. Not accessible to the web
    - Choose a convention so you’ll be able to find the scripts
PHP-CLI provides sanity checks

- Its independence from the web server makes it a good tool to check your PHP installation’s health

- `php-cli -v` outputs PHP version information
- `php-cli -i` outputs the equivalent of `phpinfo()`
  - Shows what extensions and options you have loaded. An excerpt:
    ```
    ibm_db2
    IBM DB2, Cloudscape and Apache Derby support => enabled
    Module release => 1.8.1
    ```

- **Use `php-cli -h` to see all options**
PHP-CLI Runs via PASE or QSHELL

• PASE (Portable Application System Environment) is an AIX-like environment on the IBM i

• QSHELL is a scripting environment

• I prefer PASE for calling php-cli
  ▪ PASE lets you pass parameters separately
  ▪ Qshell requires a single concatenated command string
  ▪ But you’ll see Qshell in articles, or may have a need for it

• I compared PASE and QSHELL in this article:
PASE Primer
Calling applications in PASE

- If an app runs in AIX it will probably run in PASE

- **How to use it**
  - "CALL QP2TERM"
    - Launches interactive command-line session
    - Think "TERM" = "terminal"
  - "CALL QP2SHELL PARM('/php/sendinvoice.php' &EMAIL &NAME 'Y')"
    - Launches an application of your choice directly
    - Don’t confuse with QSHELL.
  - "CALL QP2SHELL2"
    - The same as QP2SHELL except it runs in the caller’s ILE activation group.
Parameters for QP2SHELL or QP2SHELL2

From IBM Infocenter entry for QP2SHELL

- **pathName**
  - Character string that identifies the stream file in the Integrated File System that contains the OS/400 PASE program to run. The pathName string may include an absolute or relative path qualifier in addition to the stream file name. Relative path names are resolved using the current working directory.

- **argument strings**
  - Optional character strings that are passed to the OS/400 PASE program as arguments. The system copies argument strings into OS/400 PASE memory and converts them from the job default CCSID to the CCSID specified by ILE environment variable QIBM_PASE_CCSID.

Note: When calling QP2SHELL or QP2SHELL2 from CL, be sure to quote any argument string that could be interpreted as a numeric value. CL converts unquoted numeric arguments to decimal or floating-point format, which does not match the assumption made by these APIs and OS/400 PASE programs that all arguments are null-terminated character strings.
CALL PGM(QP2SHELL2) PARM('C42PDF' '–o' 'invoice.pdf' '–l' + 'tifflist.txt')

This is an example of calling an AIX binary in PASE. The C42PDF binar combines “TYPE IV TIFF” images into a multi-page PDF file.

I found the C42PDF program on the internet. It converts TIFFs to PDF, and it had a compiled AIX binary. I tried it and it worked on IBM i.
Qshell Primer
Qshell is another UNIX-like environment

- Run commands on its command line or launch Qshell environment from CL, etc.

- Allows complex shell script utilities if you are comfortable with shell scripting

- For our purposes, use it similarly to PASE, except you’ll have to concatenate commands into a single string.
Using Qshell

• **STRQSH** or **QSH** command can launch command line or immediately run a program

• **How to use it**
  - “QSH” by itself
    - Launches interactive session
  - “QSH CMD(‘MYCOMMAND’)”
    - Launches a command or application of your choice
Try PHP-CLI
Start PASE session with QP2TERM

QP2TERM launches an interactive ("terminal") PASE session where you can use UNIX-style commands
Do a simple version check: php-cli -v

$ cd /usr/local/zendsvr/bin
$ php-cli -v
PHP 5.2.10 (cli) (built: Sep 9 2009 11:50:07)
Copyright (c) 1997-2009 The PHP Group
Zend Engine v2.2.0, Copyright (c) 1998-2009 Zend Technologies
   with Zend Extension Manager v5.1, Copyright (c) 2003-2009, by Zend Technologies
   - with Zend Data Cache v4.0, Copyright (c) 2004-2009, by Zend Technologies
   - with Zend Java Bridge v3.1, Copyright (c) 2004-2009, by Zend Technologies
   - with Zend Job Queue v4.0, Copyright (c) 2004-2009, by Zend Technologies

===>
Now run our “Hello World” script

```
/QOpenSys/usr/bin/-sh
$
> cd /usr/local/zendsvr/bin
$
> php-cli /www/zendsvr/htdocs/helloworld.php
  Hello, world!
```

```bash
$ cd /usr/local/zendsvr/bin
$ php-cli /www/zendsvr/htdocs/helloworld.php
Hello, world!
```
Output the results to an IFS file

Type this into QP2TERM (note ‘pipe’ symbol):

```
> php-cli /www/zendsvr/htdocs/helloworld.php > /alan/hello.html
$ 
```

Produces a file that I can view in Notepad:
<?php

// emailme.php

// establish "from" user (optional)
ini_set("sendmail_from","sender@example.org");

// Parameters: recipient, subject, body
mail("alan@alanseiden.com",
    "Sent from CL",
    "CL can call PHP scripts.");

?>
Call the email script from a CL program

```
Columns . . .: 1 71 Edit
SEU==>  ALAN/QCLSRC
       CALLPHP

*************** Beginning of data ********************************************
0001.00  CALL PGM(QP2SHELL) +
0002.00  PARM('/usr/local/zendsvr/bin/php-cli' +
          '/www/zendsvr/htdocs/emailme.php')
0003.00  

*************** End of data ***********************************************
```

Now call your CL

Type command, press Enter.

```bash
===> call alan/callphp
```
I received the message

CL can call PHP scripts.
How to automate it?

• Add a job schedule entry (ADDJOBSCDE) for your CL program

• The CL will act like a normal CL program, which it is!

• (In the UNIX world they call this a “CRON JOB.”)

• ADDJOBSCDE JOB(SNDPRICES) SCDDATE(*NONE) CMD(CALL PGM(MYLIB/PRICEPGM)) SCDDAY(*FRI) SCDTIME(‘23:00’) FRQ(*WEEKLY) RCYACN(*NOSBM) JOBD(MYLIB/PRICEJOBD)
Receive params in PHP
Command-line arguments in PHP

• When we use PHP in a website, we receive parameters using the $_GET and $_POST arrays

• How do we get parameters/arguments when PHP is called from the command line or a CL?
Two special variables

- **$argv: array of arguments**
  - $argv[0] is always the PHP script name itself, e.g. helloworld.php
  - $argv[1], $argv[2], and so on are the actual arguments

- **$argc: count (how many) of arguments**
  - Use it to check that you’ve received what you expected
<?php

// args.php
// remember, [0] is the script itself.

echo "\n"; // blank line

// check argument count
if ($argc > 0) {
    echo $argc . " argument(s) received, including the script name itself.\n\n";

    foreach ($argv as $num=>$val) {
        echo "arg $num: $val\n";
    }
} else {
    echo "We should have had at least one argument: the script name itself.";
}

// usage: /usr/local/zendsvr/bin/php-cli /batchphp/args.php firstarg secondarg
?>
Pass parameters into CL, into PHP

```
PGM        PARM(&FIRST &SECOND)
DCL        VAR(&FIRST) TYPE(*CHAR) LEN(25)
DCL        VAR(&SECOND) TYPE(*CHAR) LEN(25)

/* Call PHP with command line arguments */
CALL       PGM(QP2SHELL) +
            PARM('/USR/LOCAL/ZENDSVR/BIN/PHP-CLI' +
              '/BATCHPHP/ARGS.PHP' +
              &FIRST &SECOND)
ENDPGM

Then...
CALL PHPPARAMS PARM('param1' 'param2')
```
What else can you do?

- Generate and email Excel spreadsheets or HTML-formatted emails
- Access web services (see my talk tomorrow!)
- Create PDF files with dynamic text and graphics
- Manipulate IFS stream files
Excel spreadsheets: simple or formatted

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
<tr>
<td>1</td>
<td><strong>Houlihan's price list</strong></td>
<td><strong>Order</strong></td>
<td>***<strong>October</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><strong>Description</strong></td>
<td><strong>Code</strong></td>
<td><strong>Cases</strong></td>
<td><strong>Btls</strong></td>
<td><strong>Size</strong></td>
</tr>
<tr>
<td>4</td>
<td><strong>WHISKEY</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>BEAM 8 STAR</td>
<td>0102030</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>BELLOWS CLUB BOURBON</td>
<td>0204030</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>BUSHMILLS IRISH</td>
<td>0393030</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CANADIAN CLUB 6YR</td>
<td>0360030</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>CROWN ROYAL</td>
<td>0387030</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>JACK DAN #7</td>
<td>0226030</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>JAMESON REG 80</td>
<td>0448930</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>JIM BEAM BOURBON</td>
<td>0260030</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>JIM BEAM RED STAG</td>
<td>0200030</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>KNOB CREEK BBN 6PK</td>
<td>0242030</td>
<td>LITER</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>MAKERS MARK</td>
<td>0261030</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>SEAG WH VO</td>
<td>0388030</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>SEAG WH 7 CROWN</td>
<td>0183030</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>W.TURKEY 101</td>
<td>0331630</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>W.TURKEY 80</td>
<td>0331530</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>COGNAC/BRANDY/GR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>APPA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>COURVOISIER VSOP</td>
<td>1234230</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>HENNESSY V.S.</td>
<td>1225030</td>
<td>LITER</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>
How to generate spreadsheets

• **Simple CSV spreadsheet: Output (echo) comma-delimited data**
  - Note: change curly quotes to straight quotes in your code
  ```php
  <?php
  header('Expires: 0');
  header('Cache-control: private');
  header('Content-Description: File Transfer');
  header('Content-Type: application/vnd.ms-excel');
  header('Content-disposition: attachment; filename="file_name.csv"');
  // commas to separate. \n for new line
  echo "firstname,lastname\nMike,Pavlak\nAndi,Gutmans";
  ?>
  ```

• **True Excel file with formatting**
  - http://pear.php.net/package/Spreadsheet_Excel_Writer
  - http://phpexcel.net (for newer format, 2007+)
  - These free packages allow complex styling, nonscrolling headers, lots more
Generate and send HTML email

From: Alan Seiden/strategic
Recipients: Houlihan’s Hasbrouck Hts <andyladrum@aol.com>, Kevin Donahue <kevin.donahue@alliedbeve <kjls1@optonline.net>, <lisa.kligge@alliedbeverage.com>, jparrott@optonline.net, japarrott@gmail.com>, <paulpruckc <shawn.kelly@alliedbeverage.com>
Subject: Order from Houlihan’s
Date: 02:25:13 PM Yesterday

An order from Houlihan’s
Allied customer 34838, Houlihan’s Hasbrouck Hts, HASBROUCK HTS NJ
Thank you for ordering from Allied Beverage!
Please print for your records.

Order reference number
Message for your rep
Order date
Web confirmation no.

WHISKEY
Sending authenticated HTML mail

```php
require_once '/usr/local/Zend/ZendFramework/library/Zend.Mail.php';
require_once '/usr/local/Zend/ZendFramework/library/Zend/Mail/Transport/Smtp.php';
$mail = new Zend_Mail_Transport_Smtp('mail.myhost.com',
    array('auth' => 'login',
          'username' => 'myuser',
          'password' => 'mypass'));
Zend_Mail::setDefaultTransport($mail);

$mail = new Zend_Mail();
$mail->setFrom($mailFrom, $mailFromName);

$mailBody = '<HTML><table><tr><td>Alert 1: hot tea at buffet<td><tr><tr><td>Alert 2: decaf available<td></table></HTML>'';

// add addresses
foreach ($custEmails as $custEmail) {
    $mail->addTo(trim($custEmail['email']), trim($custEmail['name']));
} // (foreach ($custEmails as $custEmail) )

$mail->AddTo('alan@alanseiden.com', 'Alan Seiden');
$mail->AddCc('president@common.org', 'Mr. President');

$mail->setSubject("Alert of vital importance");
$mail->setBodyHTML($mailBody);
$mail->send();
```
Generate a PDF from text provided

```php
<?php
require_once("Zend/Pdf.php");

$textToShow = $argv[1]; // assume argument is present

// Create new PDF
$pdf = new Zend_Pdf();

// Add new page to the document
$page = $pdf->newPage(Zend_Pdf_Page::SIZE_A4);
$pdf->pages[] = $page;

// Set font
$page->setFont(Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_HELVETICA), 40);

// Draw text
$page->drawText($textToShow, 100, 510);
$this->_helper->layout()->disableLayout();
$this->_helper->viewRenderer->setNoRender();
// plus I create an ellipse shape (code not shown here...see zend documentation, above, for how to do that)
$pdf->save('/alan/pdffromtext.pdf'); // now save to a file on IFS
?>
```
Run PDF generator

```
/usr/local/zendsvr/bin/php-cli
 /batchphp/pdffromtext.php "Look, ma! I'm in a PDF!"
```

**Tip**: edit the last line in php-cli so that spaces can be included in params:

```
/usr/local/zendsvr/bin/php-cli
Change $* to "@" in the last line
```

It is documented here:

Look ma, I'm in a PDF!
Method #2: PHP Job Queue
Job Queue is part of Zend Server (non-CE)

- A nice bonus when you buy the license
- More flexible than the CL technique

From the sales literature: “Long-running report generation, order processing, database cleanup, and pulling of RSS feeds are some examples of jobs that can be executed asynchronously. Zend Server for IBM i incorporates a job queue to provide full support for creating, executing and managing jobs to optimize application performance and reduce server load, minimizing application bottlenecks and improving the end-user experience.”
Job Queue allows you to...

- **Move tasks into a separate execution queue**
  - Off-load to new process (or defer to a later time)
  - Off-load to a different server

- **Execute certain tasks at a specified time**
  - Distribute processing load to off-hours

- **Use complex parameters**

- **Do it from a PHP API**
Creating Jobs

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

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

• Pass parameters
  - Simple: via query string `$_GET`
  - Complex: as an associative array of key => value pairs

• Set job options
  - Name, priority, schedule, etc.
  - Create deferred or recurring jobs

• Launch right from PHP or use the Zend Server UI
More about Zend Job Queue

Tutorial: http://files.zend.com/help/Zend-Server/working_with_jobs.htm

API signature for createHttpJob:
```php
int ZendJobQueue::createHttpJob (  
    string $url,  
    array $vars,  
    mixed $options)
```

Example with nested array variables:
```php
$queue = new ZendJobQueue();
$queue->createHttpJob(  
    'http://yourIBM.com/send_alert.php',  
    array('from'=> 'alan@alanseiden.com',  
           'to'=> array('someone@example.com',  
                        'another@example.com')),  
    array('persistent'=>false) );
```
Job Queue Management

View job status and manage execution
Other Job Capabilities

- Managing priorities
- Job dependencies
- Querying for jobs
- Checking job status and queue statistics
- Suspending and resuming recurring jobs & queues
- Passing custom HTTP headers
- Handling failures and controlling retries
Other ideas
Other ways to use batch PHP

- Web services to synchronize data
  - Call from interactive RPG
  - OR schedule for nighttime

- Use Zend Framework’s “livedocx” service to merge data with .doc templates/overlays to create MS Word .docx files

- Authenticate users with an Active Directory server
Now it’s your turn
Share what you’ve been working on

• Or brainstorm some ideas
Thanks! Stay in touch

Alan Seiden (blog and articles: alanseiden.com)
aseiden@sbsusa.com or  201-327-9400

To receive free PHP/i tips by email, write to
aseiden@sbsusa.com with subject: “PHP tips.”
Or write “PHP tips” on a business card today