

---

# PHP Batch Jobs on IBM i

---



Alan Seiden Consulting  
[alanseiden.com](http://alanseiden.com)

# Alan's PHP on IBM i focus

- Consultant to innovative IBM i and PHP users
- PHP project leader, Zend/IBM Toolkit
- Contributor, Zend Framework DB2 enhancements
- Award-winning developer
- Authority, web performance on IBM i

# Founder, Club Seiden



[club.alanseiden.com](http://club.alanseiden.com)

# Contact information

Alan Seiden

alan@alanseiden.com

201-447-2437

alanseiden.com

twitter: @alanseiden

# Where to download these slides

From my site

**<http://alanseiden.com/presentations>**

On SlideShare

**<http://slideshare.net/aseiden>**

The latest version will be available on both sites

# 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
  - PHP as a utility language
- **Running web tasks in the background for better perceived speed**
- **Tips and techniques**

# PHP/web

# PHP was built for server-side web apps

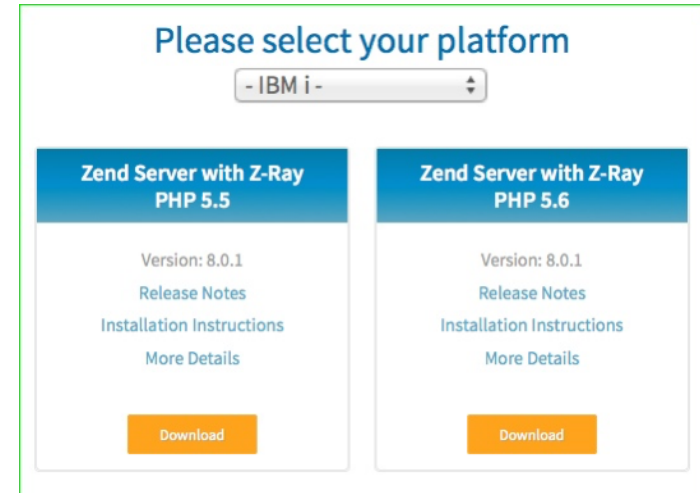
- **Started as a web development language in 1995**
- **Over time, the open source community and Zend made PHP more and more powerful**
- **Currently one of the most popular web languages**
  - It's everywhere, eBay, Wikipedia, Facebook...
  - But it's not limited to the web
  - It would be a shame to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



# Zend Server

# Zend Server for IBM i

- **Download Zend Server 8.x**
  - <http://www.zend.com/en/products/server/downloads-ibmi>
    - Easy upgrade from 6.x



- **Editions**
  - Basic (free), Professional, Enterprise
  - <http://www.zend.com/en/products/server/editions>
  - Same download, different license

# Zend Server and PHP resources

- **Support:** Zend Server “Basic” includes one year of email support. Upgrade for more years and fast phone support
  - **Register at <http://www.zend.com/en/products/server/downloads-ibmi> and also get Zend’s IBM i newsletter**
- **Web:**
  - Zend’s recorded webinars: <http://www.zend.com/resources/webinars/i5-os>
  - Zend Forums: [forums.zend.com](http://forums.zend.com) (look for IBM i Zend Server forums)
  - <http://alanseiden.com> (of course!)
    - Blog, articles
- **Books (online and print)**
  - PHP Manual: <http://php.net>
  - PHP on IBM i book by Jeff Olen & Kevin Schroeder (MC Press)
  - New Advanced PHP on IBM i book by Kevin Schroeder
- **Training from Zend**

# 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 on IFS**
  - Process the XML, generate a new XML acknowledgement, and then FTP a response
- **Send e-mail reports to administrators**

# 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**

# More possibilities with PHP on 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**

# Three methods to do this

- **Method #1: “PHP-CLI” (command line)**
  - Included in Zend Server Basic (no charge)
  - Test on command line, embed in CL/RPG
  - Need knowledge of PASE or QSHELL environment
  - Accepts simple string parameters only
  - Delay of a few seconds for PHP-CLI to load
  
- **Method #2: Zend Server’s Job Queue**
  - Included in Zend Server Professional/Enterprise Editions
  - Web based, more flexibility, dynamic scheduling
  - Accepts varied parms, multi-dimensional arrays, objects
  - Faster loading



# Three methods to do this

- **Method #3: SQL HTTP Web services**
  - RPG calls PHP using DB2 HTTP and XML capabilities
  - PHP acts as a web service under the Apache web server, like a normal PHP web script
  - Can return data to RPG via an XML string
  - Covered in my talk, “PHP Tricks for RPG Programmers”

# Method #1: PHP-CLI

# CLI = Command Line

- **PHP-CLI runs PHP when not in a web server**
  - <http://docs.php.net/manual/en/features.commandline.php>
  - Location in Zend Server: /usr/local/zendsvr6/bin/php-cli
- **Optimized for the command line**
  - No web server variables or HTTP headers
  - Get parameters from \$argv array
    - Not web server \$\_GET and \$\_POST arrays

# Tips for PHP-CLI

- **To ensure success**
  - Navigate to PHP's "bin" directory where php-cli is located
    - `cd /usr/local/zendsvr/bin` [5.x]
    - `cd /usr/local/zendsvr6/bin` [6.x-8.x]
  - The two paths above are used interchangeably in this presentation. Use the correct one for your version
  - Specify **php-cli** explicitly; "php" will not work in all version

# Techie note

- **PHP-CLI difference on IBM i vs. other systems**
  - On IBM i, php-cli is a shell script that sets up paths for PHP
  - On Linux, Windows, et al., php-cli is an actual executable file

# PHP-CLI doesn't need a web server

- **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`
  - Prefer 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
    - Assign \*RX permissions for chosen user
      - Example: `CHGAUT OBJ (' /php/appname/ ' ) USER (MYUSER) DTAAUT (*RX) SUBTREE (*ALL)`

# PHP-CLI provides sanity checks

- **Independence from the web server makes it a good tool to check health of PHP installation**
- **`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, Cloudscape and Apache Derby support => enabled  
Module release => 1.8.1
```
  - Tip: filter results by piping to “grep”
    - `php-cli -i | grep ssl` outputs only ssl info
    - `php-cli -i | grep db2` outputs only db2 info
- **Use `php-cli -h` to see all options**

# Do a simple version check: php-cli -v

```
Session A - [24 x 80]
File Edit View Communication Actions Window Help
/QOpenSys/usr/bin/-sh

$
> 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
    s [loaded] [licensed] [disabled]
    - with Zend Java Bridge v3.1, Copyright (c) 2004-2009, by Zend Technologies
    es [loaded] [licensed] [enabled]
    - with Zend Job Queue v4.0, Copyright (c) 2004-2009, by Zend Technologies

===> |
_____
_____
_____
_____

F3=Exit      F6=Print      F9=Retrieve    F11=Truncate/Wrap
F13=Clear    F17=Top       F18=Bottom     F21=CL command entry
```



# PHP-CLI Runs via PASE or QSHELL

- **Two environments**
  - **PASE** (Portable Application System Environment): AIX-like environment on the IBM i
  - **QSHELL**: 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:**
  - <http://www.mcpressonline.com/tips-techniques/programming/techtip-qshell-vs-pase.html>

# PASE Primer

# Calling applications in PASE

- If an app runs in AIX it will probably run in PASE
- Invoke with **QP2TERM** and **QP2SHELL**
  - Interactively with QP2TERM (“TERM” = “terminal”)
    - Two steps: start QP2TERM, then run application
    - **CALL QP2TERM**
    - **/usr/local/zendsvr6/bin/php-cli '/php/sendinvoice.php' fred@example.com Fred 'Y'**
  - From CL or to launch application in one step
    - **CALL QP2SHELL PARM('/usr/local/zendsvr6/bin/php-cli' '/php/sendinvoice.php' &EMAIL &NAME 'Y')**
    - Don't confuse these with QSHELL environment

# Non-PHP example of QP2SHELL

```
CALL PGM(QP2SHELL) PARM('/mypath/C42PDF' '-o'  
  'invoice.pdf' '-l' + 'tifflist.txt')
```

- **Example of calling an AIX binary in PASE**
- **C42PDF binary combines “TYPE IV TIFF” images into a multi-page PDF file**
- **Compiled binary at <http://c42pdf.ffii.org>**
- **Works well 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**
- **For our purposes, use it similarly to PASE, except you'll have to concatenate commands into a single string**

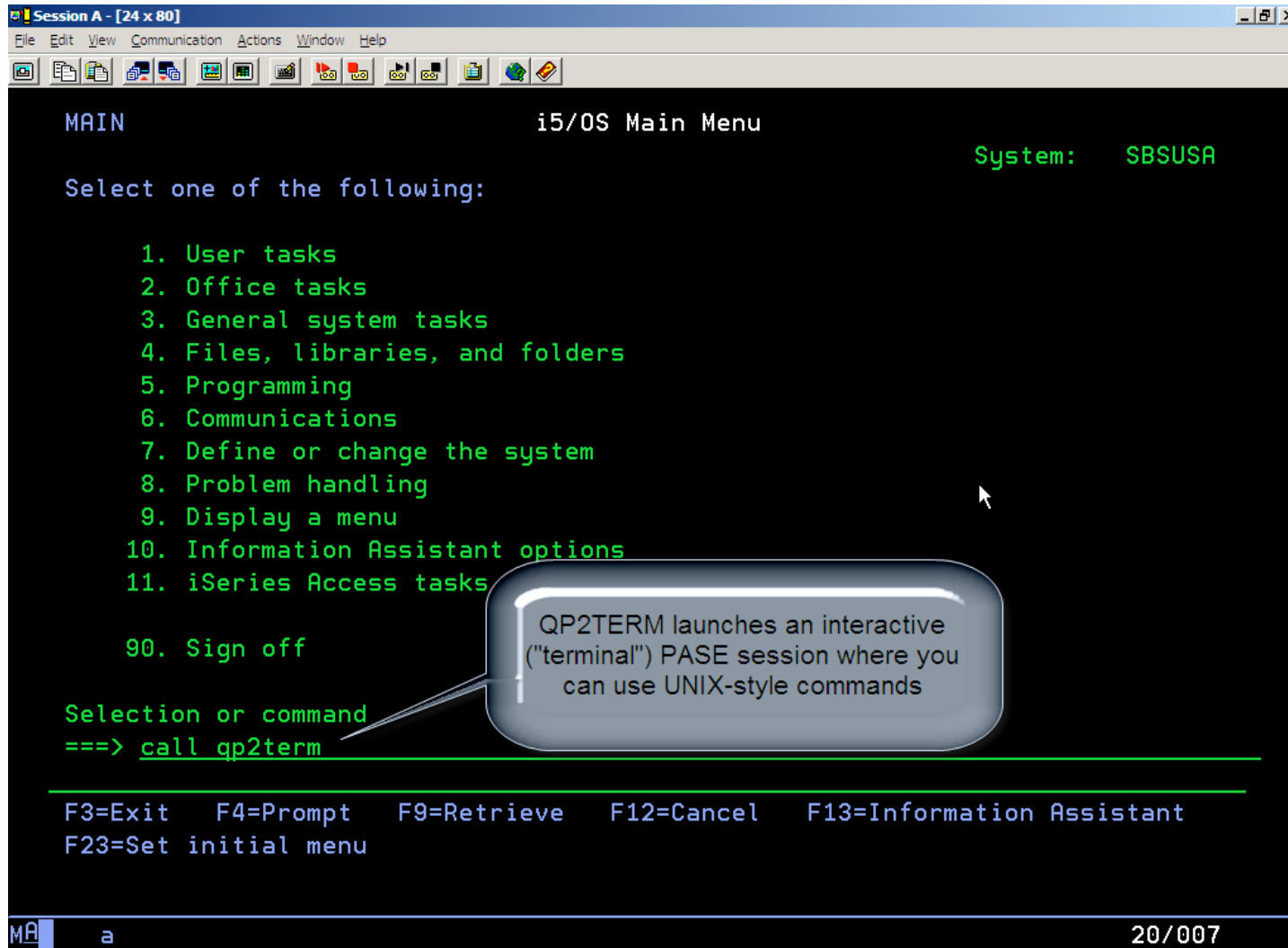
# Using Qshell

- **Invoke with QSH and QSH CMD()**
  - Interactively
    - `QSH` by itself
  - From CL or to launch application in one step
    - `QSH CMD ( 'MYCOMMAND ' )`
    - Launches a command or application of your choice
- **Note: QSH is equivalent to STRQSH**

**Try PHP-CLI**



# Start PASE session with QP2TERM



```
Session A - [24 x 80]
File Edit View Communication Actions Window Help
MAIN                               i5/OS Main Menu                               System:  SBSUSA

Select one of the following:

1. User tasks
2. Office tasks
3. General system tasks
4. Files, libraries, and folders
5. Programming
6. Communications
7. Define or change the system
8. Problem handling
9. Display a menu
10. Information Assistant options
11. iSeries Access tasks

90. Sign off

Selection or command
==> call qp2term

-----
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Information Assistant
F23=Set initial menu

MA  a                               20/007
```

QP2TERM launches an interactive ("terminal") PASE session where you can use UNIX-style commands

# Do a simple version check: php-cli -v

```
Session A - [24 x 80]
File Edit View Communication Actions Window Help
/QOpenSys/usr/bin/-sh

$
> 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 [loaded] [licensed] [disabled]
    - with Zend Java Bridge v3.1, Copyright (c) 2004-2009, by Zend Technologies [loaded] [licensed] [enabled]
    - with Zend Job Queue v4.0, Copyright (c) 2004-2009, by Zend Technologies

===> |
_____
_____
_____
_____
F3=Exit      F6=Print     F9=Retrieve  F11=Truncate/Wrap
F13=Clear    F17=Top      F18=Bottom   F21=CL command entry
```

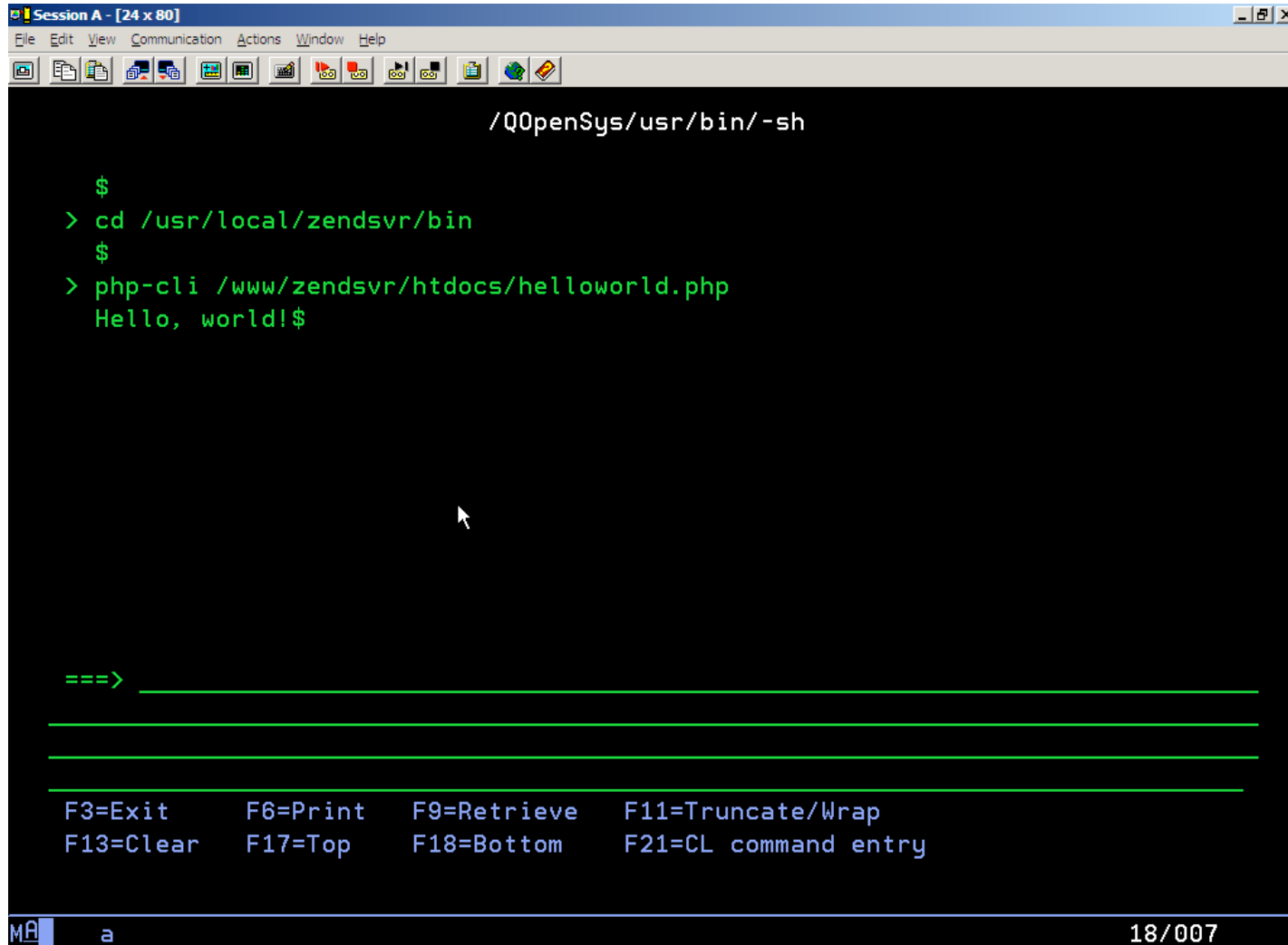
# Try a 'hello world' test

```
<?php
```

```
// if run from browser, output to browser  
// if run from command line, output there  
echo 'Hello, world!'
```

```
?>
```

# Run “Hello World” script



The image shows a terminal window titled "Session A - [24 x 80]". The window has a menu bar with "File", "Edit", "View", "Communication", "Actions", "Window", and "Help". Below the menu bar is a toolbar with various icons. The terminal content is as follows:

```
/QOpenSys/usr/bin/-sh

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

===> _____
_____
_____
_____

F3=Exit    F6=Print    F9=Retrieve  F11=Truncate/Wrap
F13=Clear   F17=Top     F18=Bottom   F21=CL command entry
```

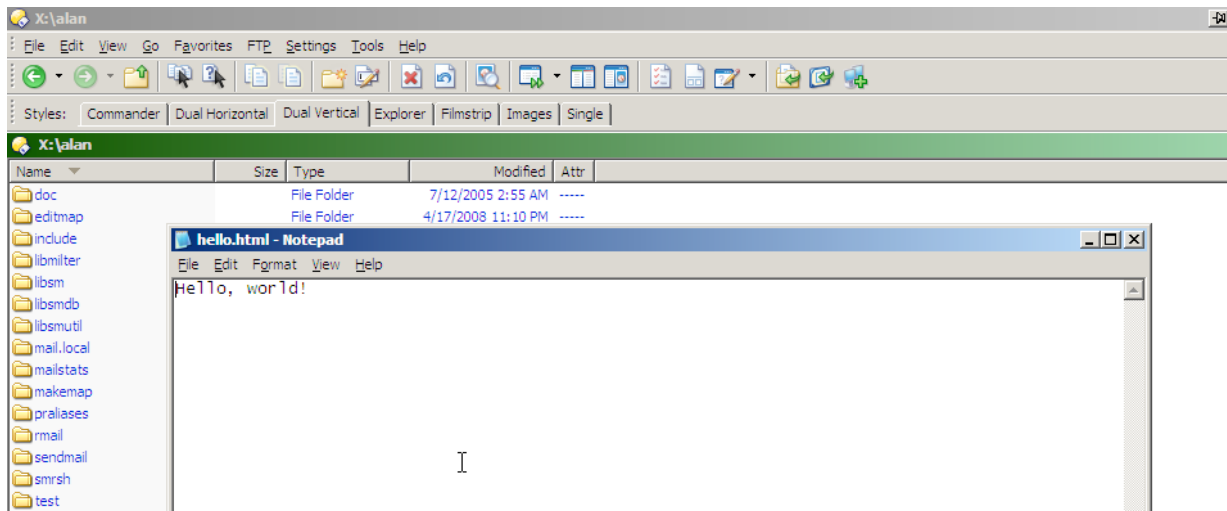
At the bottom left of the terminal window, there is a small icon and the letter "a". At the bottom right, the text "18/007" is displayed.

# Output the results to an IFS file

Type this into QP2TERM (note 'redirection' symbol):

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

Produces a file that I can view in Notepad:



# Email with PHP

```
<?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.");

?>
```

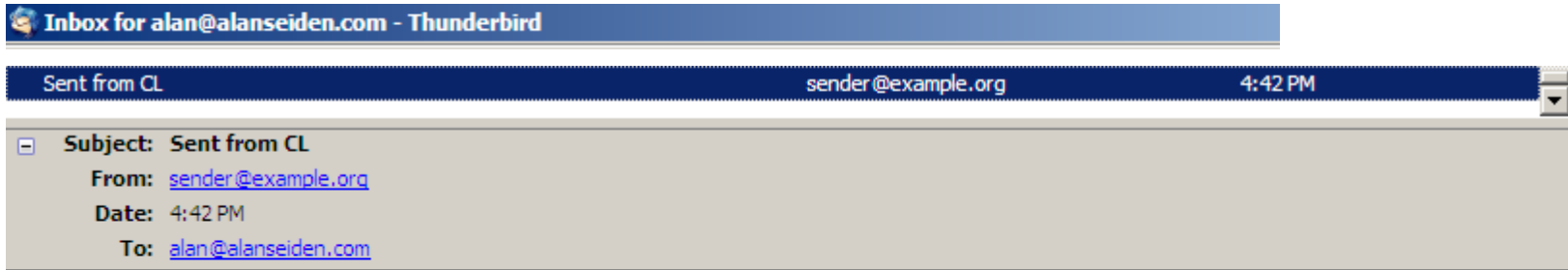
# Call the email script from a CL program

```
Columns . . . : 1 71 Edit ALAN/QCLSRC
SEU==> CALLPHP
***** Beginning of data *****
0001.00 CALL PGM(QP2SHELL) +
0002.00 PARM('/usr/local/zendsvr/bin/php-cli' +
0003.00 '/www/zendsvr/htdocs/emailme.php')
***** End of data *****
```

## Now call your CL

```
Type command, press Enter.
==> 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
- (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)
JOB(D MYLIB/PRICEJOB)
```

# Read parameters in PHP

# Command-line arguments

- **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

# Args.php gets params with \$argc and \$argv

```
<?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/zendsvr6/bin/php-cli /batchphp/args.php firstarg
secondarg
```

# Run args.php in QP2TERM (PASE)

- **cd /usr/local/zendsvr6/bin**
- **php-cli /batchphp/args.php**  
1 argument(s) received, including the script name itself.

```
arg 0: /batchphp/args.php
$
```

- **php-cli /batchphp/args.php hello "I like PHP"**  
3 argument(s) received, including the script name itself.

```
arg 0: /batchphp/args.php
arg 1: hello
arg 2: I like PHP
$
```

# CL that passes parameters to PHP

```
/* PHPPARAMS CL */
PGM      PARM(&FIRST &SECOND)
DCL      VAR(&FIRST)  TYPE(*CHAR)  LEN(25)
DCL      VAR(&SECOND) TYPE(*CHAR)  LEN(25)
DCL      VAR(&NULL)   TYPE(*CHAR)  LEN(1)  VALUE(X'00')

/* Terminate variables with null for safety */
CHGVAR VAR(&FIRST)  VALUE(&FIRST *TCAT &NULL)
CHGVAR VAR(&SECOND) VALUE(&SECOND *TCAT &NULL)

/* Call PHP with command line arguments */
CALL      PGM(QP2SHELL) +
          PARM('/USR/LOCAL/ZENDSVR6/BIN/PHP-CLI' +
              '/BATCHPHP/ARGS.PHP' +
              &FIRST &SECOND)

ENDPGM
```

# Call that CL with parameters

**Just like a normal CL program**

```
CALL PHPPARAMS PARM('param1' 'param2')
```



# Examples

# Examples of batch tasks

- **Generate and email Excel spreadsheets or HTML-formatted emails**
- **Access web services**
- **Create PDF files with dynamic text and graphics**
- **Manipulate IFS stream files**

# Excel spreadsheets: plain or formatted

|   | A         | B        |
|---|-----------|----------|
| 1 | firstname | lastname |
| 2 | Joe       | Smith    |
| 3 | Andi      | Gutmans  |
| 4 |           |          |

|    | A                            | B           | C                                                                                   | D            | E           | F              |          |
|----|------------------------------|-------------|-------------------------------------------------------------------------------------|--------------|-------------|----------------|----------|
| 1  | <b>Houlihan's price list</b> |             |  |              | ***         | <b>October</b> |          |
| 2  |                              |             |                                                                                     | <b>Order</b> |             |                |          |
| 3  | <b>Description</b>           | <b>Code</b> | <b>Cases</b>                                                                        | <b>Btls</b>  | <b>Size</b> | <b>Pk</b>      | <b>B</b> |
| 4  |                              |             |                                                                                     |              |             |                | <b>F</b> |
| 5  | <b>WHISKEY</b>               |             |                                                                                     |              |             |                |          |
| 6  | BEAM 8 STAR                  | 0102030     |                                                                                     |              | LITER       | 12             |          |
| 7  | BELLOWS CLUB BOURBON         | 0204030     |                                                                                     |              | LITER       | 12             |          |
| 8  | BUSHMILLS IRISH              | 0393030     |                                                                                     |              | LITER       | 12             | \$       |
| 9  | CANADIAN CLUB 6YR            | 0360030     |                                                                                     |              | LITER       | 12             | \$       |
| 10 | CROWN ROYAL                  | 0387030     |                                                                                     |              | LITER       | 12             | \$       |
| 11 | JACK DAN #7                  | 0226030     |                                                                                     |              | LITER       | 12             | \$       |
| 12 | JAMESON REG 80               | 0448930     |                                                                                     |              | LITER       | 12             | \$       |
| 13 | JIM BEAM BOURBON             | 0260030     |                                                                                     |              | LITER       | 12             | \$       |
| 14 | JIM BEAM RED STAG            | 0200030     |                                                                                     |              | LITER       | 12             | \$       |
| 15 | KNOB CREEK BBN 6PK           | 0242030     |                                                                                     |              | LITER       | 6              | \$       |
| 16 | MAKERS MARK                  | 0261030     |                                                                                     |              | LITER       | 12             | \$       |
| 17 | SEAG WH VO                   | 0388030     |                                                                                     |              | LITER       | 12             | \$       |
| 18 | SEAG WH 7 CROWN              | 0183030     |                                                                                     |              | LITER       | 12             | \$       |
| 19 | W.TURKEY 101                 | 0331630     |                                                                                     |              | LITER       | 12             | \$       |
| 20 | W.TURKEY 80                  | 0331530     |                                                                                     |              | LITER       | 12             | \$       |
| 21 |                              |             |                                                                                     |              |             |                |          |
|    | <b>COGNAC/BRANDY/GR</b>      |             |                                                                                     |              |             |                |          |
| 22 | <b>APPA</b>                  |             |                                                                                     |              |             |                |          |
| 23 | COURVOISIER VSOP             | 1234230     |                                                                                     |              | LITER       | 12             | \$       |
| 24 | HENNESSY V.S.                | 1225030     |                                                                                     |              | LITER       | 12             | \$       |

# How to generate spreadsheets

- **Simple CSV spreadsheet**


- Note: change curly quotes to straight quotes in your code

```
function getCsv() {  
    // commas to separate. \n for new line  
    $csvStr = "firstname,lastname\nJoe,Smith\nAndi,Gutmans";  
    return $csvStr;  
}  
  
$csvStr = getCsv();  
// save it in an IFS file  
$ret = file_put_contents('/dir/names.csv', $csvStr);
```

|   | A         | B        |
|---|-----------|----------|
| 1 | firstname | lastname |
| 2 | Joe       | Smith    |
| 3 | Andi      | Gutmans  |
| 4 |           |          |

- **True Excel file with formatting**

- <http://phpexcel.net> (.xlsx, .pdf...)
- These free packages allow complex styling, nonscrolling headers, lots more

|   | A                     | B       | C                                                                                     | D     |
|---|-----------------------|---------|---------------------------------------------------------------------------------------|-------|
| 1 | Houlihan's price list |         |  |       |
| 2 |                       |         |                                                                                       | Order |
| 3 | Description           | Code    | Cases                                                                                 | Btls  |
| 4 |                       |         |                                                                                       |       |
| 5 | WHISKEY               |         |                                                                                       |       |
| 6 | BEAM 8 STAR           | 0102030 |                                                                                       |       |
| 7 | BELLOWS CLUB BOURBON  | 0204030 |                                                                                       |       |

# Generate and send HTML email

**From:** Alan Seiden/strategic  
**Recipients:** Houlihan's Hasbrouck Hts <andyladrum@aol.com>, Kevin Donahue <kevin.donahue@alliedbeve  
<kjlr1@optonline.net>,  
<lisa.kligge@alliedbeverage.com>, <jparrott@optonline.net>, <jparrott@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

```
require_once 'Zend/Mail.php';
require_once '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><b>Alert</b>: hot tea at buffet</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->setSubject("Alert of vital importance");
$mail->setBodyHTML($mailBody);
// attach CSV file
$csvStr = getCsv(); // string
$attachment = $mail->createAttachment($csvStr, 'application/csv');
$attachment->filename = 'names.csv';
$mail->send();
```

# Generate a PDF from text provided

```
<?php
// Zend_Pdf docs: http://files.zend.com/help/Zend-Framework/zend.pdf.drawing.html
require_once 'Zend/Pdf.php';

// **** Get user-supplied text
$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/zendsvr6/bin/php-cli /batchphp/  
pdffromtext.php "Look, ma! I'm in a PDF!"
```



# Result

Look ma, I'm in a PDF!



# Method #2: PHP Job Queue

# Job Queue is part of Zend Server

- **Requires at least Professional Edition**
- **Web-based**
- **More flexible than the CL technique**
- **Easily launched from within PHP**
- **Runs long-running tasks asynchronously**

# Job Queue allows you to...

- **Move tasks into a separate execution queue**
  - Off-load to new process, later time, or different server
- **Execute certain tasks at a specified time**
  - Schedule, optionally repeat at intervals
- **Use complex parameters**
- **Control from PHP or via Zend Server GUI**
- **Launch faster than PHP-CLI**

# Creating Jobs

- Create jobs with `createHttpJob()` method

```
$queue = new ZendJobQueue ();  
$queue->createHttpJob ('http://myIbmi.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 from PHP scripts or use the Zend Server UI

# More about Zend Job Queue

Specs: <http://files.zend.com/help/Zend-Server/jobqueue-global-api.html>

Tutorial: [http://files.zend.com/help/Zend-Server/working\\_with\\_jobs.htm](http://files.zend.com/help/Zend-Server/working_with_jobs.htm)

## API signature for createHttpJob:

```
int ZendJobQueue::createHttpJob (
    string $url,
    array $vars,
    mixed $options)
```

## Example with nested array variables:

```
$queue = new ZendJobQueue();
$queue->createHttpJob(
    'http://yourIBMi.com/send_alert.php',
    array('from'=> 'alan@alaseiden.com',
        'to'=>array('someone@example.com',
            'another@example.com')),
    array('persistent'=>false) );
```

# Job Queue Management

View job status and manage execution

The screenshot displays the Zend Server web interface for job queue management. The browser address bar shows the URL `http://pc-karnaf:10081/ZendServer/Index/Index#1252590602687`. The interface includes a navigation menu with options like Monitor, Rule Management, Server Setup, and Administration. The 'Jobs' tab is active, showing a list of jobs with columns for ID, URL, Application, Status, Priority, and Run Time. The status of jobs is color-coded: green for completed, red for failed, and grey for scheduled. A 'Delete' button is visible at the bottom left, and a 'Restart PHP' button is at the bottom right. The interface also shows a search bar and navigation controls for the job list.

| ID  | URL                                                    | Application | Status    | Priority | Run Time     |
|-----|--------------------------------------------------------|-------------|-----------|----------|--------------|
| 333 | <code>http://localhost/test/xxx.php?dump_data=1</code> | local       | scheduled | normal   | 10-Sep 17:16 |
| 332 | <code>http://localhost/test/xxx.php?dump_data=1</code> | local       | failed    | normal   | 10-Sep 16:16 |
| 331 | <code>http://localhost/test/xxx.php?dump_data=1</code> | local       | failed    | normal   | 10-Sep 15:16 |
| 330 | <code>http://localhost/test/xxx.php?dump_data=1</code> | local       | failed    | normal   | 10-Sep 14:16 |
| 329 | <code>http://localhost/test/xxx.php?dump_data=1</code> | local       | failed    | normal   | 10-Sep 13:16 |
| 328 | <code>http://localhost/test/xxx.php?dump_data=1</code> | local       | failed    | normal   | 10-Sep 12:16 |
| 327 | <code>http://localhost/test/xxx.php?dump_data=1</code> | local       | failed    | normal   | 10-Sep 11:16 |
| 326 | <code>http://localhost/test/xxx.php?dump_data=1</code> | local       | completed | normal   | 10-Sep 10:16 |
| 325 | <code>http://localhost/test/xxx.php?dump_data=1</code> | local       | completed | normal   | 10-Sep 09:16 |
| 324 | <code>http://localhost/test/xxx.php?dump_data=1</code> | local       | completed | normal   | 10-Sep 08:16 |
| 323 | <code>http://localhost/test/xxx.php?dump_data=1</code> | local       | completed | normal   | 10-Sep 07:16 |
| 322 | <code>http://localhost/test/xxx.php?dump_data=1</code> | local       | completed | normal   | 10-Sep 06:16 |

# Other Job Capabilities

- **Set priority and dependencies**
- **Get job output from PHP**
- **Check job status and queue statistics**
- **Passing custom HTTP headers**
- **Management of jobs**
  - Handling failures and controlling retries
  - Suspend/resume recurring jobs
  - Code tracing, monitor



# Job Queue Timeout Tip

- **If task will run for a long time, increase the Job Queue and FastCGI request timeouts**
  - /usr/local/zendsvr6/etc/jqd.ini
    - Default: zend\_jobqueue.http\_job\_timeout = 120
  - /www/zendsvr6/conf/fastcgi.conf
    - Default: RequestTimeout="60"
- **Otherwise you'll get HTTP 408 errors and unpredictable results in the Job Queue GUI**

# Creative use of Zend Job Queue

- **Imagine a web app where user's navigation path is predictable**
  - User signs in, is redirected to a welcome page, then clicks to see a data-rich dashboard
  - Dashboard contains several slow, user-specific queries
- **Question: How can job queue enable faster loading of the dashboard?**
- **Answer: run queries as soon as user logs in, asynchronously, using job queue**
  - When user gets to dashboard, gather the pre-fetched data

# More examples online

- <https://github.com/clarkphp/Code-Examples/tree/master/ZendServerJobQueue>
  - Examples from Zend's Clark Everetts



The screenshot shows a GitHub repository page for 'Code-Examples / ZendServerJobQueue'. At the top, it indicates the current branch is 'master'. Below the repository name, there is a commit summary: 'ZendServerJobQueue: In-progress Simple Producer Consumer Example (jus...'. The commit is attributed to 'clarkphp' and was authored 8 months ago. A list of files is displayed, each with a file icon, the filename, and a brief description of the change or file's purpose.

| File Name                       | Description                                                             |
|---------------------------------|-------------------------------------------------------------------------|
| README.md                       | ZendServerJobQueue: To README add note on high concurrency directive... |
| job_A_predecessor_sample...     | Add example of two jobs, one a predecessor of the other.                |
| job_B_predecessor_sample...     | Add example of two jobs, one a predecessor of the other.                |
| job_one.php                     | Add scripts to demonstrate a simple recurring job via API.              |
| job_starter_non_existent_tar... | ZendServerJobQueue comment clarifications                               |
| job_starter_predecessor_sa...   | ZendServerJobQueue: Correct typo in source file; add TODOs in README.   |
| job_starter_producer_consu...   | ZendServerJobQueue: In-progress Simple Producer Consumer Example (ju... |
| job_starter_recurring.php       | ZendServerJobQueue comment clarifications                               |
| job_starter_statuses.php        | Initial set of Zend Server Job Queue examples.                          |

# Other ideas

# Other ways to use batch PHP

- **Web services to synchronize data**
  - Call from interactive RPG (better would be my “PHP Tricks for RPG Programmers” technique)
  - OR schedule for nighttime
- **Use Zend Framework’s “livedocx” service to merge data with .doc templates/overlays to create MS Word .docx files**

# Contact and tips

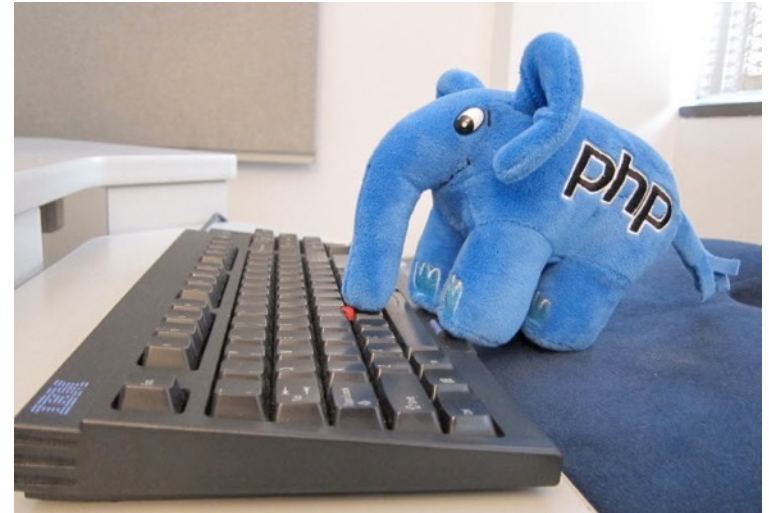
## Alan Seiden

Alan Seiden Consulting

Ho-Ho-Kus, NJ

Free newsletter:

<http://alanseiden.com/tips>



---

**alan@alanseiden.com • 201-447-2437 • twitter: @alanseiden**