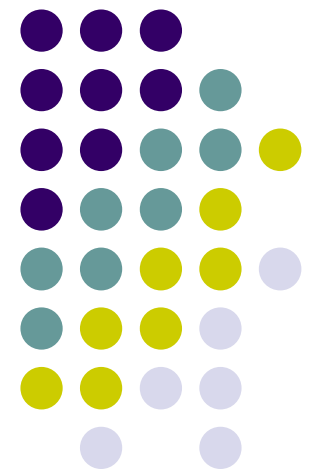


Zend Framework for IBM i

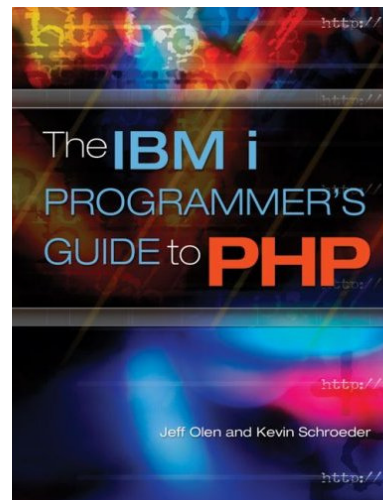
Getting Started



Introduction



- Author of bestselling “IBM i Programmers Guide to PHP”
- Zend Certified Engineer – PHP 5
- IBM i developer for 20+ years
- Co-founder of Olen Business Consulting, Inc.
- Lead Systems Software Engineer at Vertex Business Services NA





What we will learn

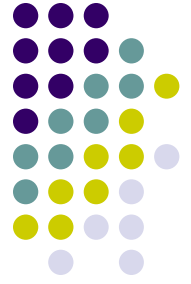
- How the “use at will” architecture simplifies the transition from straight PHP to ZF applications.
- What improvements using MVC offers
- Some of the advantages of using ZF to build your PHP applications.
- A real world example using PEAR, Zend Framework and the IBM i Toolkit to create PDFs from SCS spool files and email them as attachments. All called from an RPG program.

MVC – Model/View/Controller



What is MVC?

- MVC is a design pattern. You might think of a design pattern as a skeleton or “framework” on which your application will be built.
- The MVC design pattern is based on the idea that there are three components that work together to form any complex application. These components are the Model, the View and the Controller.



MVC Advantages

Entangled code

- Changes ripple the effects

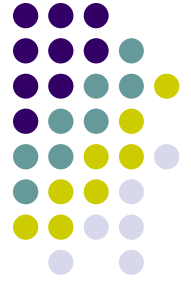
Unreliable applications

Difficult to extend

- What if the table or database changes?
- What if you need to use application on a mobile phone?

Difficult to maintain

The Three Components of MVC: Model, View and Controller



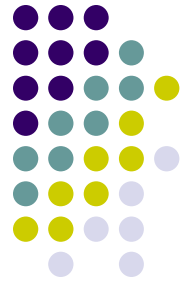
- What is a Model?
 - Models are the interfaces with the DB.
 - This is where your actual database i/o and validation takes place.
 - There is also where you will place the “business logic”.
- What is a View?
 - The View contains any code that relates to presentation (i.e. User interface) and presentation logic such as caching or templates.

The Three Components of MVC: Model, View and Controller – con't



- What is a controller?
 - A controller is just that. It controls and launches each of the various activities in an application. For instance in a CRUD application you might have methods in your controller for addAction, updateAction, viewAction and deleteAction.
 - Although there may be some business logic here in the Controller, strictly speaking this is not best practice.

MVC and Zend Framework

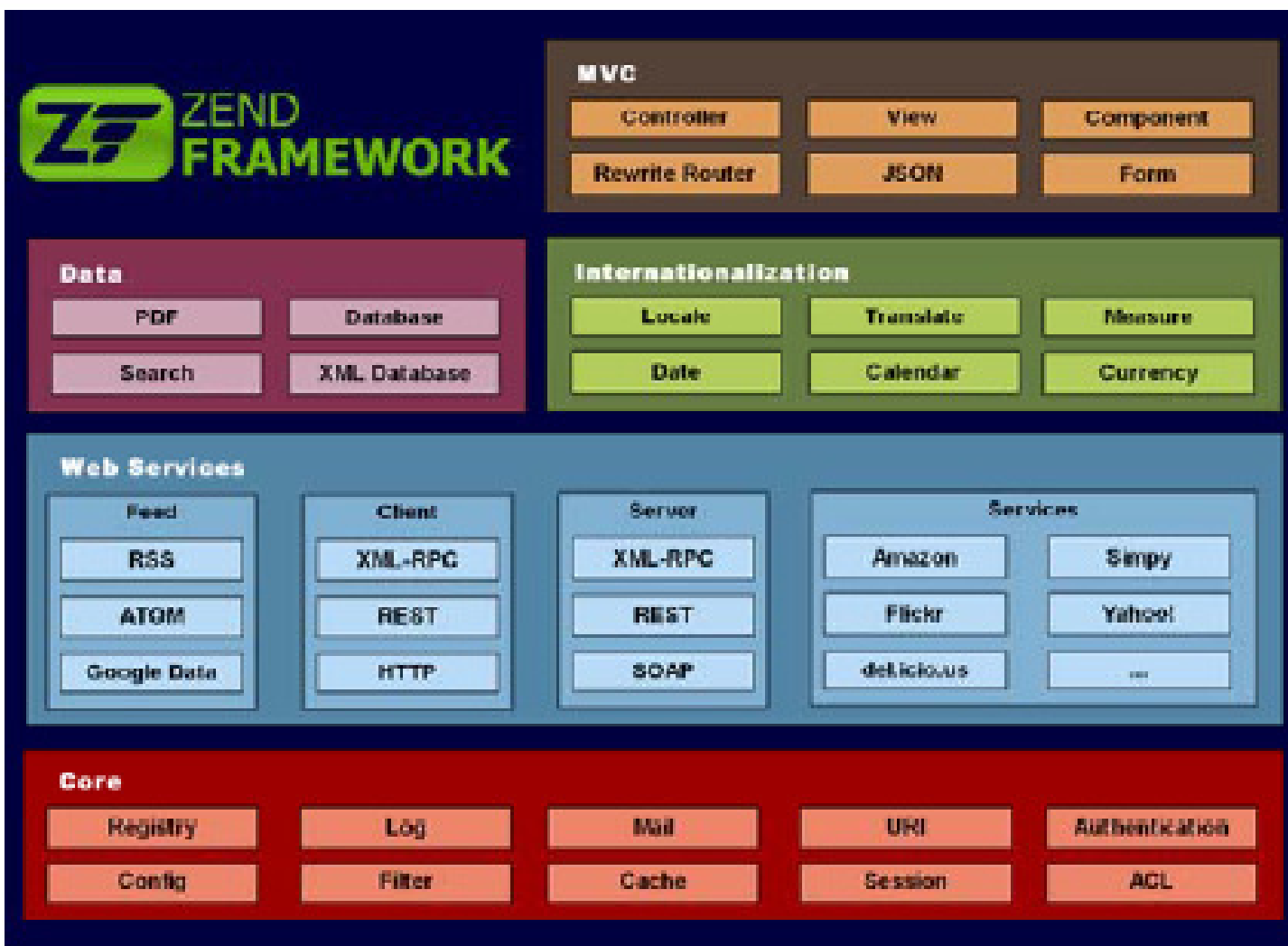


Why use a Framework at all?

- A single entry point for all applications.
- Removing the headaches involved with maintaining numerous scripts, each with their own relative paths, database connections, authentication, etc.
- Allowing the [consolidation](#) and reuse of code.
- Often, changes to business rules have no impact on Views or Controllers. Even when Views, Controllers, or both are affected, the required changes tend to be minimal, localized, and easy to find.
- Your team can debug an MVC application more easily because the flow of invocations and events is consistent.

What's in Zend Framework?

Simple,
convenient,
object-oriented
solutions to
common web
application
development
tasks,
including:



ZF use at will architecture



Our first ZF application...

```
<?php
```

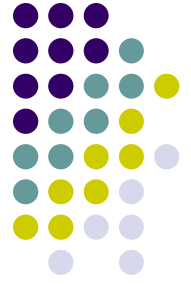
```
require_once("Zend/Version.php");
```

```
echo "Zend Framework Version: ".Zend_Version::VERSION;
```

```
?>
```

Save as: getZFVersion.php

ZF use at will architecture



Our first ZF application...

```
<?php  
require_once("Zend/Version.php");  
echo "Zend Framework Version: ".Zend_Version::VERSION;  
?>
```

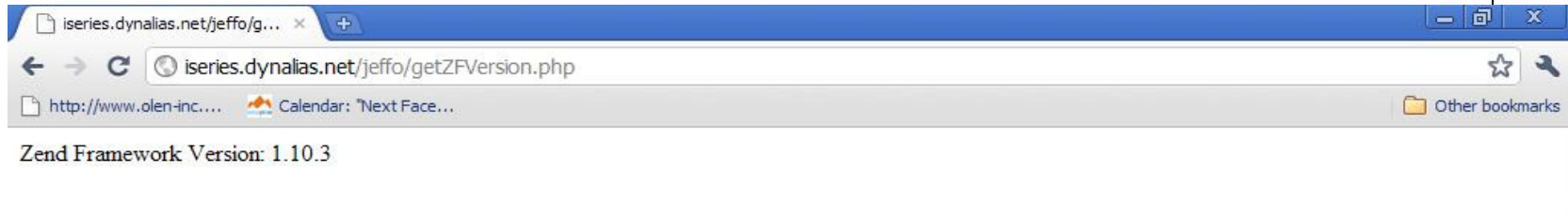
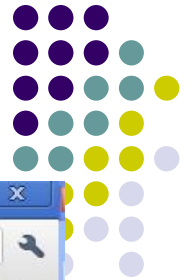
Save as: getZFVersion.php

Scope resolution Operator

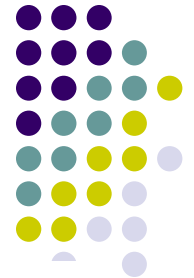
Also called Paamayim Nekudotayim



ZF use at will architecture



ZF use at will architecture

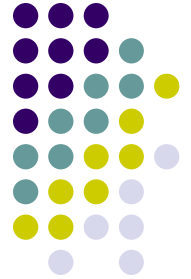


```
<?php

//-----
// Set to display errors - TESTING ONLY!!
//-----
ini_set('display_errors', '1');

//-----
// Use Autoloader to include the Zend Framework Classes we are going to u
//-----
require_once 'Zend/Loader/Autoloader.php';
Zend_Loader_Autoloader::getInstance()->setFallbackAutoloader(true);
```

ZF use at will architecture



We'll talk more about the Autoloader later

```
$autoloader = Zend_Loader_Autoloader::getInstance();  
$autoloader->registerNamespace('Foo_');
```

ZF use at will architecture



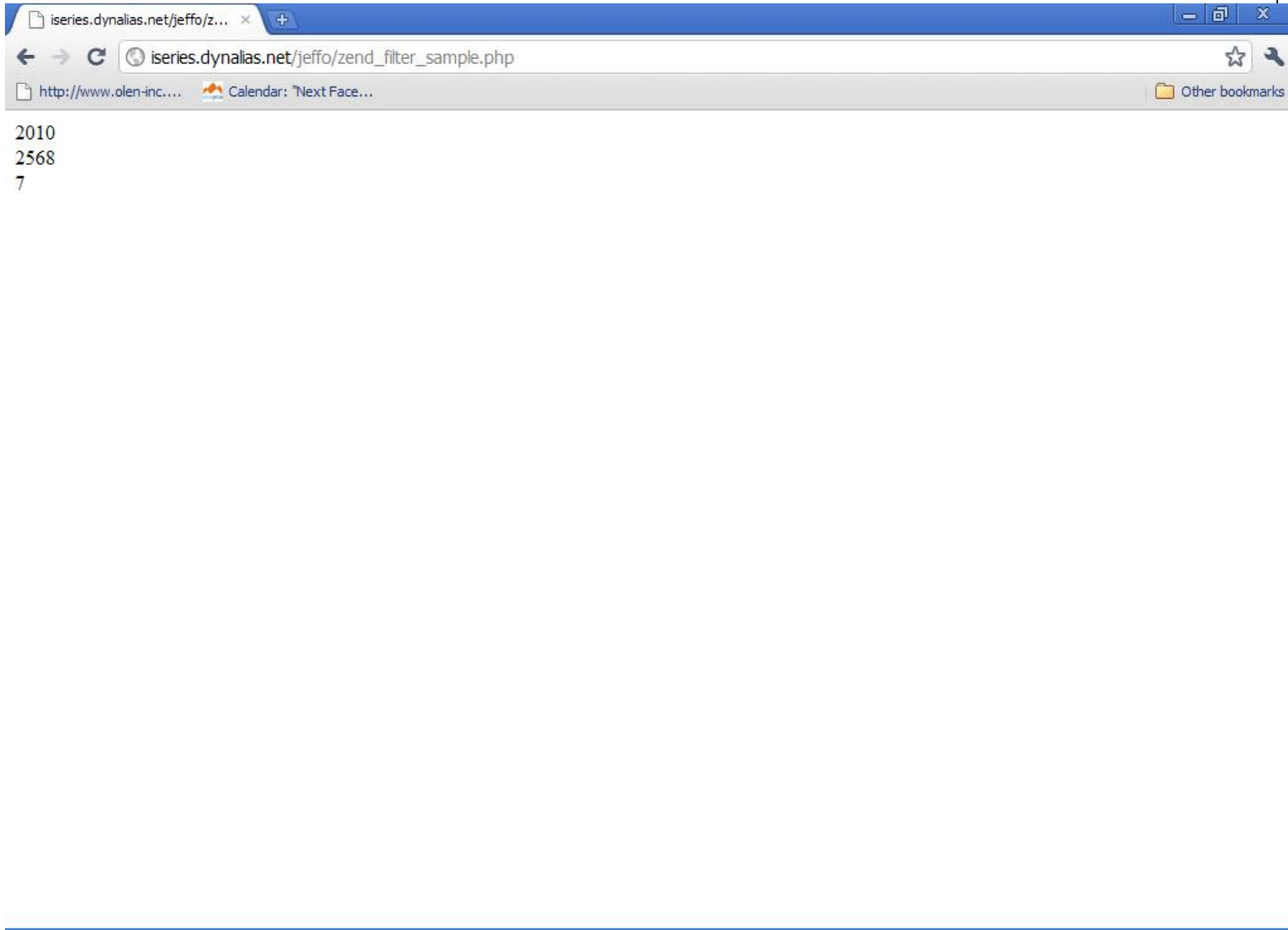
```
<?php

//-----
// Set to display errors - TESTING ONLY!!
//-----
ini_set('display_errors', '1');

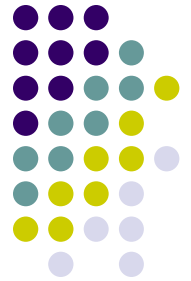
//-----
// Use Autoloader to include the Zend Framework Classes we are going to use.
//-----
require_once 'Zend/Loader/Autoloader.php';
Zend_Loader_Autoloader::getInstance()->setFallbackAutoloader(true);

$filter1 = new Zend_Filter_Digits();
print $filter1->filter("2010 was a good year")."<br>";
// 2010
print $filter1->filter("25.68 is all the cash I have")."<br>";
// 25
print $filter1->filter("It is -7 degrees outside.")."<br>";
// 7
```

ZF use at will architecture



ZF use at will architecture

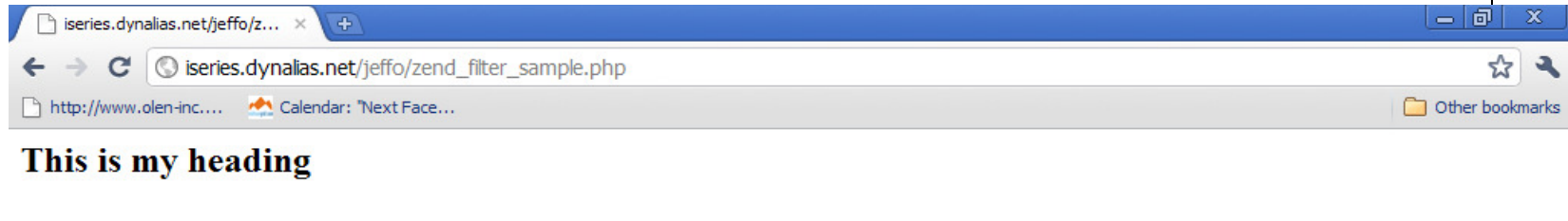
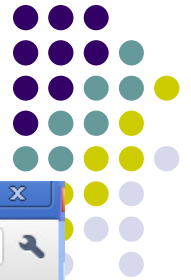


Kick it up a notch

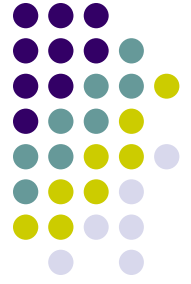
```
$filter2 = new Zend_Filter_StripTags();
$workString = "<h2><b>This is my heading</b></h2>";
print $workString."<br>";
print $filter2->filter($workString)."<br>";
```

```
$filterChain = new Zend_Filter();
$filterChain->appendFilter(new Zend_Filter_Alnum())
    ->appendFilter(new Zend_Filter_StringToUpper());
$workString = "At the tone the time will be 4:30pm!";
print $workString."<br>";
print $filterChain->filter($workString)."<br>";
// ATTHETIMETHETIMEWILLBE430PM
```

ZF use at will architecture



ZF use at will architecture

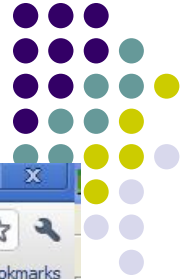


Leave the whitespace

```
$filter2 = new Zend_Filter_StripTags();  
$workString = "<h2><b>This is my heading</b></h2>";  
print $workString."<br>";  
print $filter2->filter($workString)."<br>";
```

```
$filterChain = new Zend_Filter();  
$filterChain  
    ->appendFilter(new Zend_Filter_Alnum(array('allowwhitespace' => true)))  
    ->appendFilter(new Zend_Filter_StringToUpper());  
$workString = "At the tone the time will be 4:30pm!";  
print $workString."<br>";  
print $filterChain->filter($workString)."<br>";  
// ATTHETONETHETIMEWILLBE430PM
```

ZF use at will architecture

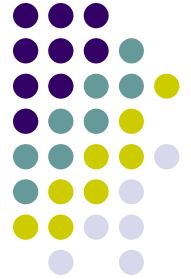


The screenshot shows a web browser window with the following content:

- Address bar: `iseries.dynalias.net/jeffo/z...`
- Page title: `iseries.dynalias.net/jeffo/zend_filter_sample.php`
- Page content:
 - This is my heading**
 - This is my heading
 - At the tone the time will be 4:30pm!
 - AT THE TONE THE TIME WILL BE 430PM

The text "AT THE TONE THE TIME WILL BE 430PM" is circled in red.

ZF in the real world



PDF converter / Mailer Mashup

1. Uses Zend Framework PDF functions
2. Uses Zend Server for IBM i toolkit functions
3. Uses PEAR Mail and MIME extensions
4. Uses the PHP-CLI command line interface
5. Is written to be called from an RPG program.



Where to find PEAR extensions



The screenshot shows the PEAR website homepage. The browser address bar displays 'pear.php.net'. The page features a green header with the PEAR logo and navigation links: Main, Support, Documentation, Packages, Package Proposals, Developers, Bugs, Home, News, Quality Assurance, The PEAR Group, and Mirrors. The main content area is titled 'PEAR - PHP Extension and Application Repository' and includes sections for 'What is it?', 'Hot off the Press', and 'Recent Releases'. The 'Recent Releases' section lists several packages with their versions and release dates, such as Text_CAPTCHA 0.4.2 and PHP_CodeSniffer 1.3.0RC2. The 'Popular Packages*' section lists packages like HTML_Template_Flexy 1.3.11 and Net_Socket 1.0.10. The 'Recently Proposed' section lists packages like Console::PHP cPanel Script and Services::Services OAuthU ploader.

PEAR - PHP Extension and Application Repository

>> What is it?

PEAR is a framework and distribution system for reusable PHP components.

Sounds good? Perhaps you might want to know about [installing PEAR on your system](#) or [installing pear packages](#).

You can find help using PEAR packages in the [online manual](#) and the [FAQ](#).

If you have been told by other PEAR developers to sign up for a PEAR website account, you can use [this interface](#).

>> Hot off the Press

PEAR in March 2010

After a quiet holiday season, the PEAR community has started rumbling again.

Digg gave PEAR a plug, new versions of Mail, Services_Facebook, System_Daemon, and HTML_Template_IT were released, the number of bugs reported dropped to **less than one per package** for a brief time, and two promising new proposals in PEPs arrived.

What's even more exciting, we've got a continuous integration environment currently set up; discussion (and bug fixing) is turning more and more towards [the future of pear](#), and the quiet on the mailing lists has vanished in a recent flood of posts.

Even better, we've seen the conceptual basis of PEAR (channel server and installer) take off with Pirum and PEARFarm being launched; and several large projects like PHPUnit, Smarty, and even Zend Framework (1) being installable via pear channels.

If this level of activity is anything to judge by, the future of PEAR looks bright for 2010!

19th Mar 2010 08:31pm. Read [more](#) or see [comments](#)

Recent Releases:

- Text_CAPTCHA 0.4.2 (Sun, 16th Jan 11)
- PHP_CodeSniffer 1.3.0RC2 (Fri, 14th Jan 11)
- Services_JSON 1.0.3 (Fri, 14th Jan 11)
- Net_Growl 2.2.2 (Thu, 13th Jan 11)
- Services_ProjectHoneyPot 0.5.3 (Thu, 13th Jan 11)

Popular Packages*:

- HTML_Template_Flexy 1.3.11 (231.29)
- Net_Socket 1.0.10 (140.11)
- XML_Parser 1.3.4 (73.79)
- MIME_Type 1.2.1 (41.09)
- Mail 1.2.0 (26.24)

* downloads per day

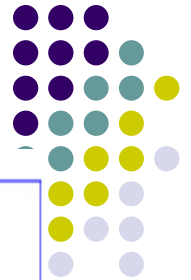
Recently Proposed:

- Console::PHP cPanel Script by stonelgg
- Web Services::Services OAuthU ploader by withgod
- Nagios::Nagios Plugin by feraudet
- Database::SDBA by atkrad
- XMPP::JAXL by abhinavsingh



EmailTool.php

emailtool.php



```
<?php

// include path for PEAR utilities and i5Toolkit classes
ini_set("include_path", ini_get("include_path").":/usr/local/Zend/Core/share/pear:/usr/loca
require_once "Mail.php";
require_once "mime.php";

include_once('Toolkit_classes.php');

// set Zend Framework autoloader
require_once("Zend/Loader/Autoloader.php");
Zend_Loader_Autoloader::getInstance()->setFallbackAutoloader(true);

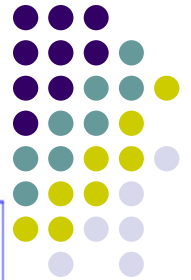
// retrieve passed parms from RPG pgm.
$spoolFileName = $_SERVER['argv'][1]; // spool file name
$spoolJob      = $_SERVER['argv'][2]; // spool file job - job#/user/job
$spoolFileNbr = $_SERVER['argv'][3]; // spool file number
$filePath     = $_SERVER['argv'][4]; // IFS path of output file
$toEmail      = $_SERVER['argv'][5]; // to email addresses (comma separated)
$reportDesc   = $_SERVER['argv'][6]; // report description for email

// setup email "from" and "subject" lines
$from        = "tbs_noreply@olen-inc.com";
$subject     = "Report - ".$reportDesc;

// setup email connection parms (not using IBM i email server)
$host        = "ssl://smtp.gmail.com";
$port        = "465";
$username    = "info@olen-inc.com";
$password    = "pass123";
```

EmailTool.php

(con't)



emailtool.php

```
// connect to IBM i and retrieve spool file data
try {
    $conn = new i5_Connection('localhost', "jolen", "pass123");
    $conn->connect();
} catch (Exception $e) {
    echo "Failure to connect. ";
    echo $e->getMessage();
    die();
}

// parse the job name elements into an array
$nameElements = explode("/", $spoolJob);

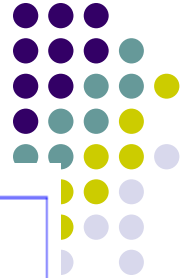
$spoolData = i5_spool_get_data( $spoolFileName,
                               $nameElements[2],
                               $nameElements[1],
                               $nameElements[0],
                               $spoolFileNbr);

// explode the spooldata into an array of pages
$spoolPages = explode("\f", $spoolData);

// explode each page into an array of print lines
foreach ($spoolPages as $key=>$pageData) {
    $spoolPages[$key] = explode("\r", $pageData);
}
```

EmailTool.php

(con't)



emailtool.php

```
// explode each page into an array of print lines
foreach ($spoolPages as $key=>$pageData) {
    $spoolPages[$key] = explode("\r", $pageData);
}

// create your PDF document
$pdf = new Zend_Pdf();

// parse the entire spool file into the PDF document
foreach ($spoolPages as $key=>$pageData) {
    // create a new PDF page
    $page = $pdf->newPage(Zend_Pdf_Page::SIZE_LETTER);
    // Set font
    $page->setFont(Zend_Pdf_Font::fontWithName(Zend_Pdf_Font::FONT_COURIER), 7);
    // Get page height
    $pageHeight = $page->getHeight();
    $x = 10;
    $y = $pageHeight - 20;

    // draw each line of text into the PDF
    foreach ($pageData as $line) {
        //echo $line."<br>";
        $page->drawText($line, $x, $y);
        $y -= 7;
    }
    // add the completed page to the PDF
    $pdf->pages[] = $page;
}

$tempName = $filePath.'/'.$spoolFileName.'.pdf';
$pdf->save($tempName);
```

EmailTool.php

(con't)



emailtool.php

```
$message = new Mail_mime(); // create new mail_mime object
$recipients = explode(",", $toEmail); // load array of email recipients
$message->setTXTBody($reportDesc); // load the report description as the message body
$message->addAttachment($tempName); // add the PDF we just created as the attachment
$body = $message->get(); // create the actual email body text (e.g. BASE64 c

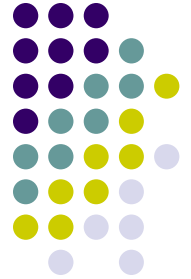
// create extra header info
$extraheaders = array("From"=>"Olen Business Consulting<info@olen-inc.com>", "Subject"=>$su
$headers = $message->headers($extraheaders);

// create a PEAR mail object to send the email
$mail = Mail::factory('smtp',
    array ('host' => $host,
        'port' => $port,
        'auth' => true,
        'username' => $username,
        'password' => $password));

// execute the send method to send the email
$rs = $mail->send($recipients, $headers, $body);

// check for error
if (PEAR::isError($rs)) {
    echo("<p>" . $rs->getMessage() . "</p>");
} else {
    echo("<p>Message successfully sent!</p>");
}
?>
```

EmailTool.php - Test



QSH Command Entry

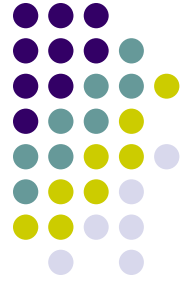
```
===> /usr/local/zendsvr/bin/php-cli /www/zendsvr/htdocs/jeffo/emailtool.php  
TBS003P 162629/JOLEN/PHPJOB 1 /home/jolen jeff olen@hotmail.com  
'Vidafon Lead List'
```

Learning more...



- **System i Network offers 5 week PHP training classes including:**
 - [PHP101 for the RPG Programmer](#) - with Mike Pavlak
Check the web for upcoming classes
 - [PHP102 for the RPG Programmer](#) - with Mike Pavlak
Next class starts February 28th
 - [Introduction to Zend Framework](#) – with Jeff Olen
Next class starts April 6th
- **Zend offers a whole range of classes including ZF Certification Prep.**

Thank you!



Questions? Last chance!