10 Cool PHP Things You Can Run on Your System i

Check out open-source components that are easy to install, offer business value, and can liven your screen

by Alan Seiden

PHP is one of the most popular web development tools. With Zend Core for i5/OS (*zend.com/en/products/core*), the System i community has access to an incredibly diverse body of programming tools, code packages, and even full-fledged applications.

In this article, I focus on 10 "cool" open-source PHP components that are easy to install, offer business value and, in many cases, are highly graphical.

A note about file formats: Downloads are compressed as zip files or tar.gz. To extract tar.gz files from within Windows, I recommend using ZipGenius (*zipgenius.it*), a free zip utility.

1. PHP/SWF Charts

PHP/SWF Charts (*maani.us/charts*) creates attractive, colorful, animated charts and graphs on the web. Your scripts provide the data, which can come from your DB2 database or any other source. The charts are built in Flash, the popular cross-browser technology. (Flash Player 6 or later must be installed in your visitors' browsers. You can download Flash for free at *macromedia .com/go/getflashplayer.*)

The basic version of PHP/SWF Charts is free. For the registered version, which includes technical support and the ability to link charts to other web content and embed charts in other Flash documents, the author charges a modest license fee. Registered users can click charts to drill down for details. To install the program, download and extract charts.zip. Copy the contents (charts.swf, charts.php, and charts_library) into a web server directory such as /www/zendcore/charts. To see samples of the many available charts — line, pie, bar, mixed, and more — go to the "gallery" section of the PHP/SWF Charts website at *maani .us/charts/index.php?menu=Gallery*. Code samples are available at *maani.us/charts/index.php?menu=Reference*.

Using the code in Figure 1, I created an example chart (Figure 2). Although not apparent from the screen shot, script 3dpie.php generates a 3-D animated, spinning pie chart from data I provided. Showchart.php uses the InsertChart function to output the chart. On my server, I ran the script with a URL similar to http://example.com:89/charts/showchart.php.

FIGURE 1
HTML to create a pie chart using PHP/SWF charts

```
<?php
// 3dpie.php
include 'charts.php';

$chart[ 'chart_data' ] = array ( array ( "", "Vinny", "Jeff",
"Rose", "Jim"), array ( "", 15, 27,45,60 ) ); // specify data
$chart[ 'chart_grid_h' ] = array ( 'thickness'=>0 );
$chart[ 'chart_pref' ] = array ( 'rotation_x'=>60 );
$chart[ 'chart_rect' ] = array ( 'x'=>50, 'y'=>50, 'width'=>300,
'height'=>200, 'positive_alpha'=>0 );
$chart[ 'chart_transition' ] = array ( 'type'=>"spin",
'delay'=>.5, 'duration'=>.75, 'order'=>"category" );
$chart[ 'chart_type' ] = "3d pie";
$chart[ 'chart_value' ] = array ( 'color'=>"000000", 'alpha'=>65,
'font'=>"arial", 'bold'=>true, 'size'=>10, 'position'=>"inside",
'prefix'=>"", 'suffix'=>"", 'decimals'=>0, 'separator'=>"",
'as_percentage'=>true );

$chart[ 'draw' ] = array ( array ( 'type'=>"text",
'color'=>"CCC000", 'alpha'=>74, 'size'=>25, 'x'=>0, 'y'=>0,
'width'=>500, 'height'=>50, 'text'=>"Pizza left uneaten", 'h_
align'=>"center", 'v_align'=>"middle" )) ;

$chart[ 'legend_label' ] = array ( 'layout'=>"horizontal",
'bullet'=>"circle", 'font'=>"arial", 'bold'=>true, 'size'=>12,
'color'=>"ffffff", 'alpha'=>85 );
$chart[ 'legend_rect' ] = array ( 'x'=>0, 'y'=>45, 'width'=>50,
'height'=>210, 'margin'=>10, 'fill_color'=>"ffffff", 'fill_
alpha'=>10, 'line_color'=>"000000", 'line_alpha'=>0, 'line_
thickness'=>0 );
$chart[ 'legend_transition' ] = array ( 'type'=>"dissolve",
'delay'=>0, 'duration'=>1 );

$chart[ 'series_color' ] = array ( "00ff88", "ffaa00","44aaff",
"aa00ff" );
$chart[ 'series_explode' ] = array ( 25, 75, 0, 0 );

SendChartData ( $chart );
?>

<HTML>
<!-- showchart.php -->
<BODY bgcolor="#FFFFFF">

<?php
include "charts.php";

//insert the charts.swf flash file into the web page
//tell charts.swf to get the chart's data from 3dpie.php created
in the first step
echo InsertChart ( "charts.swf", "charts_library", "3dpie.php" );

?>

</BODY>
</HTML>
```

2. PHP Interactive

For developers, PHP Interactive (*hping.org/phpinteractive*) is a simple PHP program that provides a convenient, web-based way to test PHP ideas. You don't need to create standalone PHP script files every time you want to try something new. Paste your PHP code into PHP Interactive and try it there.

When you click "update," the results of your script appear below (Figure 3). You can choose between raw or HTML (browser-style) views. You can also add multiple tabs (each holding a script), name the tabs, and delete tabs. PHP Interactive remembers your scripts from session to session.

To aid development, the program provides a search box to look up PHP function definitions, leading to the appropriate manual page. To install the program, download and extract contents of phpinteractive-0.2.tar.gz. Inside the phpinteractive folder, create a subdirectory called /scripts. (The application needs this scripts folder for storing the scripts you create.) Now copy the phpinteractive folder under /www/zendcore, making /www/zendcore/phpinteractive/. Run php interactive on your System i at a URL such as http://example.com:89/phpinteractive.

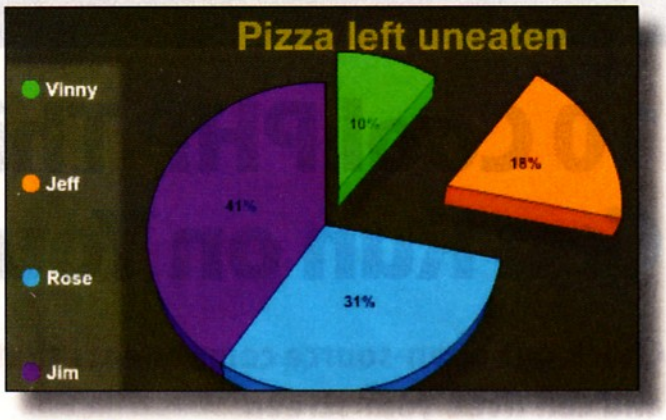

FIGURE 2
Whimsical chart based on code in Figure 1

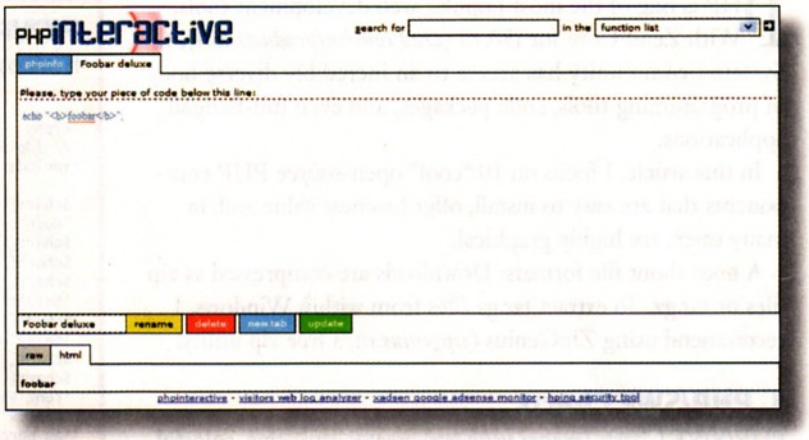

FIGURE 3
PHP Interactive

3. Dial Gauge

Dial Gauge (*phpclasses.org/browse/package/4080.html* — free membership required) provides a visual progress indicator that resembles a car's speedometer. The developer needs to supply only a low and high range and the number to display. Such a display could be useful in executive dashboard systems.

To install the package, download and extract the zip file. Copy the files to a folder such as /www/zendcore/dialgauge. Within Zend Core, you will need to enable the "gd" graphics library, a popular toolkit for the dynamic creation of images. You can do this by tapping Zend Core's administration pages (Configuration/Extensions) and then restarting Zend Core. You can run the example script on your server at a URL such as http://example.com:89/dialgauge/example.php. Figure 4 shows an example of a gauge.

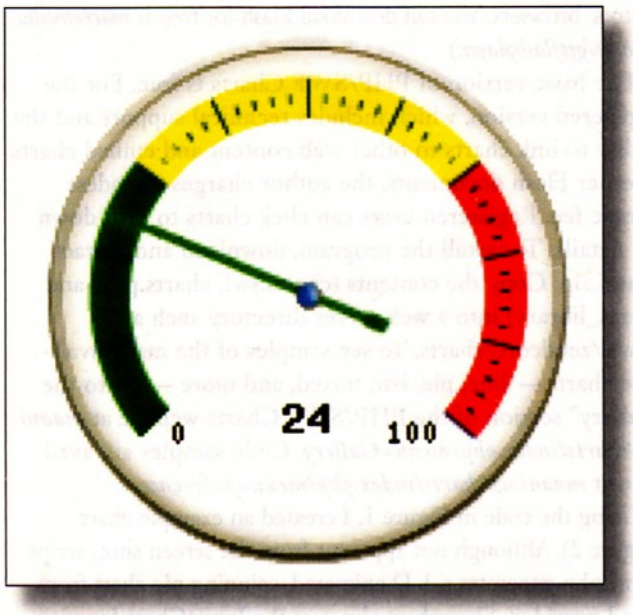

FIGURE 4
A Dial Gauge shows where value falls within a range

4. TailFile Class

When Unix professionals want to see the latest entries in a large log file, they use the "tail" command. With the TailFile class (*phpclasses.org/browse/package/499.html* — free registration required), you can control "tail" functionality from PHP. I show how to use it to detect any new errors that appear in the PHP error log.

To install the program, download and extract the contents of the zip file. Copy the contents, tail-example.php and tailfile.phpc, to a directory such as /www/zendcore/tailfile. I've modified tail-example.php (Figure 5) so that it works with Zend Core for i5/OS. I called my version check_php_log.php. It checks the PHP error log for new errors, something I often want to do.

FIGURE 5
TailFile class checks for new errors in PHP error log

```
#!/usr/local/Zend/Core/bin/php -q
<?
// check_php_log.php
// Display new errors that appear in Zend Core's error log
// Adapted from Matthew Frederico's tail-example.php
//
// First line gives default location of PHP executable in Zend
Core for i5/OS
// $tailfile is set to default name of Zend Core's error log

/*****************************/
/* Configuration parameters */
/*****************************/
# File Locations

$tailfile               = "/usr/local/Zend/Core/logs/php_error_
log";

# How many seconds delay between checks

$check_delay    = 5;

/*********************************************/
/*         M A I N   P R O G R A M           */
/*********************************************/
include("tailfile.phpc");       // PHP Tail class.

// This is where we watch our PHP error log for updates
//-------------------------------------------
$t = new TailFile($tailfile);
print "Watching PHP log... \n";

while ($t->isOpen())
{
        $t->checkUpdates();
        $myres = $t->getResults();

        if ($myres)
                print $myres;

        $t->wait($check_delay);
}
//-------------------------------------------
```

I launched QShell and my script with the following statement:

```
qsh
/www/zendcore/htdocs/tailfile/check_php_log.
php
```

Press F3 to exit. Figure 6 shows the recent messages that my script found.

5. Sparkline PHP Graphing Library

According to Edward Tufte, an expert on analytical design, sparklines are "intense, simple, word-sized graphics." Thanks to the package at *sparkline.org*, developers can embed sparklines in otherwise dry reports or web pages.

To install the library, download and unzip the contents of the zip file into a folder such as /www/zendcore/htdocs/sparkline. Within Zend Core, you need to enable the "gd" graphics library extension. You can do this by accessing Zend Core's administration pages (Configuration/Extensions) and then restarting Zend Core. Several sample scripts are included. I ran one that charts the performance of IBM stock using dynamic data from Yahoo Finance. The URL with parameters was something such as http://example.com:89 /sparkline/samples/stock_chart2.php?s=ibm&y=5. Figure 7 shows the resulting sparkline.

6. Mantis/400

Mantis (*mantis400.com*) is a popular web-based help-desk and bug-tracking application. You'll never lose track of support issues again! Best of all, Mantis has run on good old DB2 since 2007, when I worked with consultants from IBM, Zend, Curbstone Corporation, and Mantis's project leader, Victor Boctor, to create Mantis/400, a DB2-enabled version.

When you run Mantis/400, its data is saved in ordinary DB2 files. Your traditional programming tools, such as RPG, can access these files.

Mantis includes the following features:

- IT staff members and end users can report and comment on bugs and issues. (Figure 8 shows Mantis's own issue list.)
- Users can upload supporting files (e.g., logs) to report their issues.
- The system can automatically send e-mail to appropriate users when reports are updated.
- Categories allow for easy searching of issues.
- Change logs help build a methodical change-management process.
- IT staff members can post product road maps and news.

To install the application, go to *mantis400.com* and request

```
                    QSH Command Entry

 $
> /www/zendcore/htdocs/tailfile/check_php_log.php
  Watching PHP log...

  [13-Oct-2007 19:19:43] PHP Parse error:  syntax error, unexpected T_STRING in /www/zendcore/htdocs/tailfile/errorfile.php on line
   4

  [13-Oct-2007 19:21:00] PHP Parse error:  syntax error, unexpected T_STRING in /www/zendcore/htdocs/tailfile/errorfile.php on line
   4
```

FIGURE 6 Check_php_log.php runs in QShell and uses TailFile to show new errors

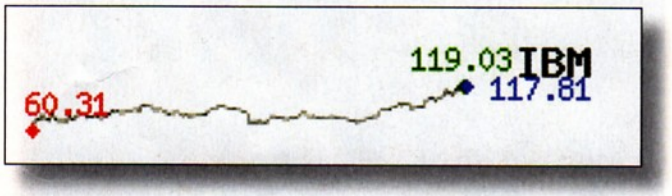

FIGURE 7
Sparkline chart of IBM's stock performance in 2005

My script is meant to run in the QShell terminal environment rather than in a browser. Why? A browser will not let a web page run perpetually. It will time out unless JavaScript redirects the address periodically. QShell has no such restrictions. It lets a PHP script run a continuous loop. Here you see PHP's power for general-purpose shell scripting.