Just a peak at PHPExcel

Presented to

KC PUG: Kansas City's PHP User Group

2011-04-16

Daniel Holmes

kcpug
Kansas City's
AMP Community

Today's Discussion

- What is PHPExcel?
- How to use it
- How to add PHPExcel to your project
- Tips, Tricks, Gotchas, Architecture

kcpug
Kansas City's
AMP Community

PHPExcel

- It's like having an excel engine in PHP
- Cross Platform
- Reads and Writes spreadsheets
- Multiple formats
  - xls, xlsx, ods, csv, pdf and others
- VERY OO – Great for extending
  - Don't be afraid to refer to the source

kcpug
Kansas City's
AMP Community

PHPExcel: What can I use it for?

Captain Obvious says:
"Dumping Data to Excel"

kcpug
Kansas City's
AMP Community

PHPExcel: What can I use it for?

Writing Spreadsheets

- Supports multiple worksheets, formats, column widths, group/outline
- Add Excel 2007 Validation rules
- Set page Setup options for printing (scaling, landscape, etc)
- Use functions in your export

kcpug
Kansas City's
AMP Community

PHPExcel: What can I use it for?

Reading Spreadsheets

- Read specific cells from a user upload
- Use tabs, headings and values as a db import
- Combine several CSV's into separate sheets.
- Works with calculated columns to!
- Use one as a template for writing!

kcpug
Kansas City's
AMP Community

Writing your first spreadsheet

Supports “Fluent” interfaces (chaining) but not required. I'll show in a mix of both styles. Use what works best for you.

```
require_once("PHPExcel.php");

$excel = new PHPExcel();
$excel->setTitle("My Spreadsheet");
$sheet = $excel->setActiveSheetIndex(0)->getActiveSheet();
$sheet->setTitle("Books");

$sheet->setCellValue("A1", "Your Book Orders");
$sheet->getStyle("A1")->getFont()->setSize(25);

$sheet->getRowDimension('1')->setRowHeight(40);
$sheet->getRowDimension('2')->setRowHeight(30);
$sheet->getDefaultColumnDimension()->setWidth(30);

$writer = PHPExcel_IOFactory::createWriter($excel, 'Excel2007');
$writer->save('myfile.xls');
```

Reading Spreadsheets

```
$row = "3";
$startRow = $row;
$col = "0";

$startColumn = PHPExcel_Cell::stringFromColumnIndex($col);
$sheet->setCellValueByColumnAndRow($col, $row, "ISBN"); $col++;
$sheet->setCellValueByColumnAndRow($col, $row, "Title"); $col++;
$sheet->setCellValueByColumnAndRow($col, $row, "Author"); $col++;

$sheet->setCellValueByColumnAndRow($col, $row, "Num Pages"); $col++;
$sheet->setCellValueByColumnAndRow($col, $row, "Quantity"); $col++;
$sheet->setCellValueByColumnAndRow($col, $row, "Price"); $col++;

$endColumn = PHPExcel_Cell::stringFromColumnIndex(--$col);
```

TIP: There are better ways to lay this out. This pattern does provide some flexibility for adding more logic though.

Add Some Data

```
foreach ($order as $itemId => $orderItem) {
    $col = 0;
    $row++;
    $book = $orderItem->getBook();
    $sheet->setCellValueByColumnAndRow($col, $row, $book->getISBN()); $col ++;
    $sheet->setCellValueByColumnAndRow($col, $row, $book->getTitle()); $col ++;
    $sheet->setCellValueByColumnAndRow($col, $row, $book->getAuthor()); $col ++;
    $sheet->setCellValueByColumnAndRow($col, $row, $book->getNumPages()); $col ++;
    $sheet->setCellValueByColumnAndRow($col, $row, $orderItem->getQuantity()); $col ++;
    $sheet->setCellValueByColumnAndRow($col, $row, $orderItem->getPrice()); $col ++;
}
```

kcpug
Kansas City's
AMP Community

Format your Column Headers

```
$styleArray = array(
    'borders' => array(
        'outline' => array(
            'style' => PHPExcel_Style_Border::BORDER_THICK,
            'color' => array('argb' => 'FF000000'),
        ),
    ),
    'font' => array(
        'bold' => true,
    ),
    'alignment' => array(
        'vertical' => PHPExcel_Style_Alignment::VERTICAL_BOTTOM,
    ),
    'fill' => array(
        'type' => PHPExcel_Style_Fill::FILL_SOLID,
        'startcolor' => array(
        'rgb' => 'EEFFFF'))
);

$sheet->getStyle("{$startCol}{$row}:{$endCol}{$row}")
    ->applyFromArray($styleArray);
```

Reading Spreadsheet Files

```
$inputFileName = './sampleData/example1.xls';

/**  Identify the type of $inputFileName  **/
$inputFileType = PHPExcel_IOFactory::identify($inputFileName);

/**  Create a new Reader of the type that has been identified  **/
$objReader = PHPExcel_IOFactory::createReader($inputFileType);

/**  Load $inputFileName to a PHPExcel Object  **/
$objPHPExcel = $objReader->load($inputFileName);
```

Several of these next examples were provided by Mark Baker's excellent presentation Writing, Reading and Arithmetic! He saw a tweet of mine and offered to share his slides! Thanks Mark!!

Reading Tab Delimited Files

```
$inputFileType = 'CSV';
$inputFileName = './sampleData/example1.tsv';

/**  Create a new Reader of the type defined in $inputFileType  **/
$objReader = PHPExcel_IOFactory::createReader($inputFileType);
/**  Set the delimiter to a TAB character  **/
$objReader->setDelimiter("\t");

/**  Load the file to a PHPExcel Object  **/
$objPHPExcel = $objReader->load($inputFileName);
```

kcpug
Kansas City's
AMP Community

Another example from Mark Baker

Extracting Values

```
/* Get the calculated value from a cell */
$worksheet = 'Worksheet2';
$cell = 'A1';

/* Get a normal value from a cell */
$cellValue = $objPHPExcel->getSheetByName($worksheet)
                         ->getCell($cell)->getValue();

/* Get the calculated value from a cell */
$cellValue = $objPHPExcel->getSheetByName($worksheet)
                         ->getCell($cell)->getCalculatedValue();

/* Grab an array of Values for a range*/
$cellRange = 'A1:C3';
$cellValues = $objPHPExcel->getActiveSheet()
                          ->rangeToArray($cellRange);
```

kcpug
Kansas City's
AMP Community

More from Mark Baker

Extracting Values with the Iterator

```
$worksheet = $objPHPExcel->getActiveSheet();

echo $worksheet->getTitle(),'<br />';

foreach ($worksheet->getRowIterator() as $row) {
    echo 'Row number: ',$row->getRowIndex(),'<br />';

    $cellIterator = $row->getCellIterator();
    /**  Loop all cells in the row  **/
    $cellIterator->setIterateOnlyExistingCells(false);
    foreach ($cellIterator as $cell) {
        if (!is_null($cell)) {
            echo 'Cell: ',$cell->getCoordinate(),
                 ' - ',$cell->getCalculatedValue(),'<br />';
        }
    }
}
```

kcpug
Kansas City's
AMP Community

More from Mark Baker

Excel Formulas in PHPExcel

Cube Functions None Implemented (0/7)	e.g. CUBESETCOUNT(), CUBEVALUE()
Database Functions All Implemented (12/12)	e.g. DSUM(), DMIN(), DGET()
Date & Time Functions All Implemented (21/21)	e.g. DATEDIF(), EOMONTH(), DAY()
Engineering Functions All Implemented (39/39)	e.g. BESSELI(), ERF(), IMPOWER()
Financial Functions 45 Implemented (45/54)	e.g. DDB(), NPER(), SLN(), RATE()
Information Functions 15 Implemented (15/18)	e.g. ISEVEN(), ISTEXT()
Logical Functions All Implemented (7/7)	e.g. AND(), TRUE(), IF()
Lookup & Reference Functions 14 Implemented (14/18)	e.g. CHOOSE(), ROW(), TRANSPOSE()
Mathematical & Trigonometric Functions 59 Implemented (59/60)	e.g. FACT(), GCD(), PERMUT(), SUM()
Statistical Functions 76 Implemented (76/86)	e.g. CHIDIST(), COUNTIF(), LINEST()
Text & Data Functions 28 Implemented (28/34)	e.g. CHAR(), MID(), REPT(), PROPER()

More from Mark Baker

Tips and Tricks

Beware of big sheet titles!

```
$sheetTitle = $this->formatDate('now') . " - " . $customer->getName() ;
$sheet->setTitle('Active '.substr($sheetTitle,0,23));
```

Remove HTML, but keep blocky data

```
$cellStyle =$sheet->getStyleByColumnAndRow($col, $row);
$cellStyle->getAlignment()->setWrapText(true);
html_entity_decode(strip_tags(
   $userComment->getDisplayValue()),ENT_QUOTES));
```

kcpug
Kansas City's
AMP Community

Tips and Tricks

Beware the Leading = of death!

A leading equal in a cell starts function call.

So, be sure to trim($val,'=')
or you can disable formula parsing be default.

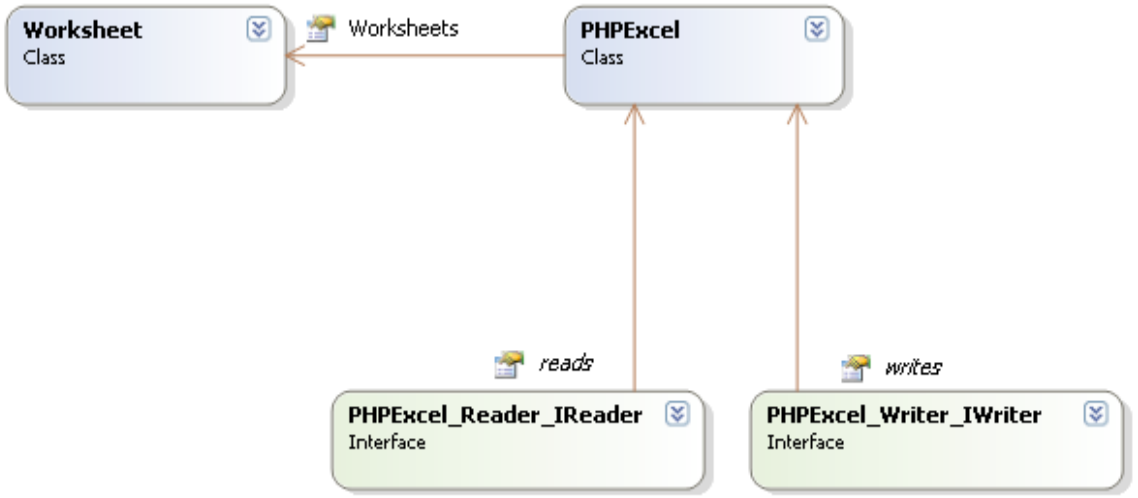
kcpug
Kansas City's
AMP Community

PHPExcel Limitations

- Slower than some of the older alternatives (such as PEAR PHP_Excel_Writer or PHP_Excel_Reader).
- Memory Hungry. Everything is built up internally and then written in one shot.
- Methods such as cell caching, reading in “chunks”, and setting styles across a range of cells (rather than each cell individually) or by row or column reduce the memory overheads.
- Unrecognised/Unsupported workbook elements are discarded when reading.
- Not all Readers/Writers support all features of the core PHPExcel class.
- Calculation Engine can’t handle references to external workbooks.
- Can’t execute Macros or VBA functions.

kcpug
Kansas City's
AMP Community

Architecture



- PHPExcel 422
  - CachedObjectStorage 394
  - Calculation 395
  - Cell 395
  - locale 395
  - Reader 395
  - RichText 394
  - Shared 422
  - Style 395
  - Worksheet 395
  - Writer 395
    - Excel2007 394
    - Excel5 395
      - CSV.php 394
      - Excel2007.php 394
      - Excel5.php 394
      - HTML.php 394
      - IWriter.php 394
      - PDF.php 394
      - Serialized.php 394

kcpug
Kansas City's
AMP Community

Questions?

PHPExcel is hosted on Codeplex
http://phpexcel.codeplex.com/

Daniel Holmes

@dan_holmes

kcpug
Kansas City's
AMP Community