
PHP EXCEL

Ilia Alshanetsky - @iliaa

Me, Myself & I

- * PHP Core Developer since 2001
- * Author of Excel extension ;-)

The Need...

**Ability to parse and generate
genuine Excel documents.**

PRIOR TO PHP-EXCEL

PEAR ExcelReader/Writer - Old & Abandoned...



PHPExcel - 30kb per cell & really, really slow...



HENCE PHP-EXCEL

Based around LibXL library

<http://libxl.com/>

Costs \$199

**PHP Interface is Open Source
available at:**

https://github.com/iliaal/php_excel

Core Features

- * Can create Biff-8 and XML based documents
- * Can parse Biff 5 through 8 and XML based documents
- * Really, really fast... 500,000 cells per second fast
- * Does not eat a ton of memory
- * Simple OO Interface

CODE TIME!

Simple Document Creation

```
$x = new ExcelBook();
```

```
$s = $x->addSheet("Sheet 1");
```

```
$s->write(1, 1, 'Test');
```

```
$s->write(2, 2, 123);
```

```
$x->save("file.xls");
```

	A	B	C
1			
2		Test	
3			123
4			
5			

XML Document Handling

```
$x = new ExcelBook(NULL, NULL, TRUE);
```

```
$s = $x->addSheet("Sheet 1");
```

```
$s->write(1, 1, 'Test');
```

```
$s->write(2, 2, 123);
```

```
$x->save("file.xlsx");
```



Fancy Office 2010
XML format

Writing An Entire Row

```
$db = new PDO('database');

$x = new ExcelBook;
$s = $x->addSheet("Weekly TPS Report");

$i = 0;
foreach ($db->query('SELECT * FROM tps', PDO::FETCH_NUM) as $row) {
    $s->writeRow($i++, $row);
}

$x->save("file.xls");
```



Picture Time!

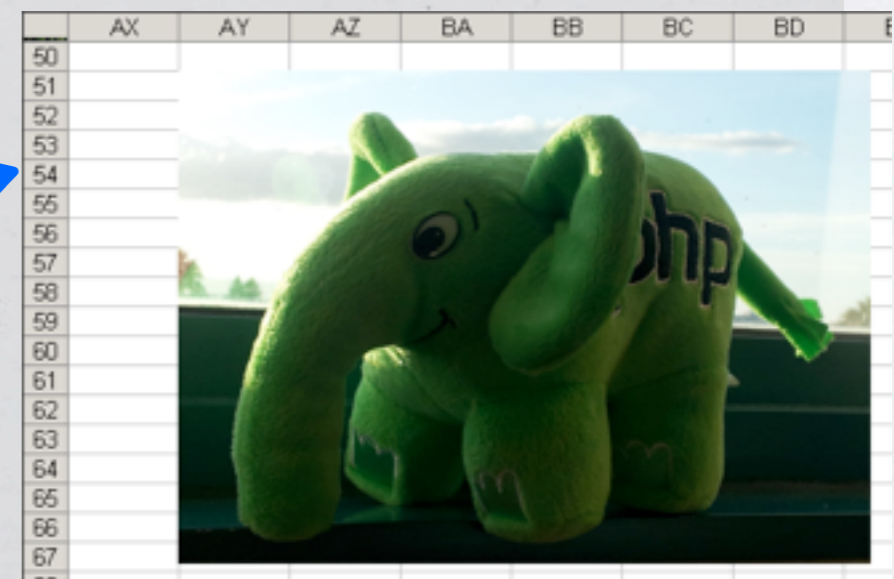
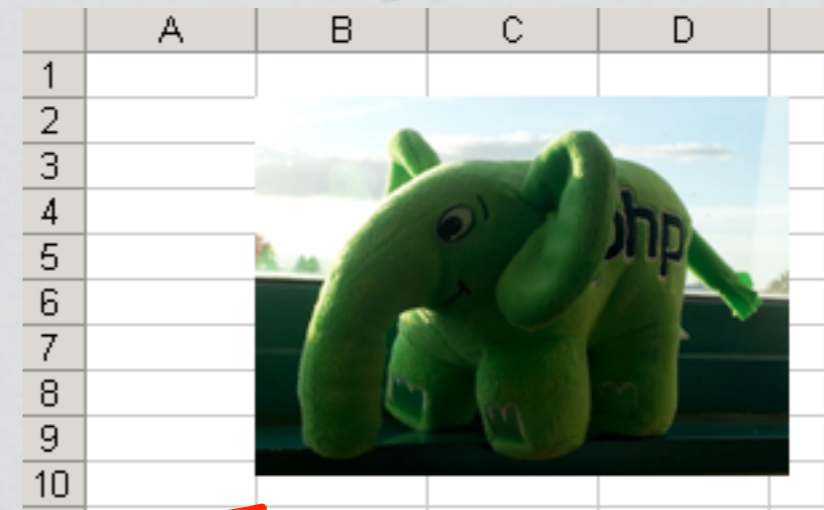
```
$x = new ExcelBook();  
$s = $x->addSheet("Sheet 1");  
  
$pic_file = "./theGreenOne.jpg";
```

```
// Loads picture into memory  
$pic = $x->addPictureFromFile($pic_file);
```

```
// Place picture on 2nd row, 50% scale  
$s->addPictureScaled(1, 1, $pic, 0.5);
```

```
// Place picture with specific dimensions  
$s->addPictureDim(50, 50, $pic, 400, 285);
```

```
$x->save("e.xls");
```



LOTS OF FORMATTING OPTIONS

Borders

```
$x = new ExcelBook();
$s = $x->addSheet("Borders");

$oClass = new ReflectionClass('ExcelFormat');
foreach ($oClass->getConstants() as $c => $val) {
    if (strpos($c, 'BORDERSTYLE_') !== 0) {
        continue;
    }
    $format = $x->addFormat();
    $format->borderStyle($val);

    $name = preg_replace('!.*_!', '', $c);

    $s->write($row++, $col++, $name, $format);
}

$x->save("borders.xls");
```


Output

NONE														
	THIN													
		MEDIUM												
			DASHED											
				DOTTED										
					THICK									
						DOUBLE								
							HAIR							
								MEDIUMDASHED						
									DASHDOT					
										MEDIUMDASHDOT				
											DASHDOTDOT			
												MEDIUMDASHDOTDOT		
													SLANTDASHDOT	

Border Colors

```
$x = new ExcelBook();
$s = $x->addSheet("Border Colors");

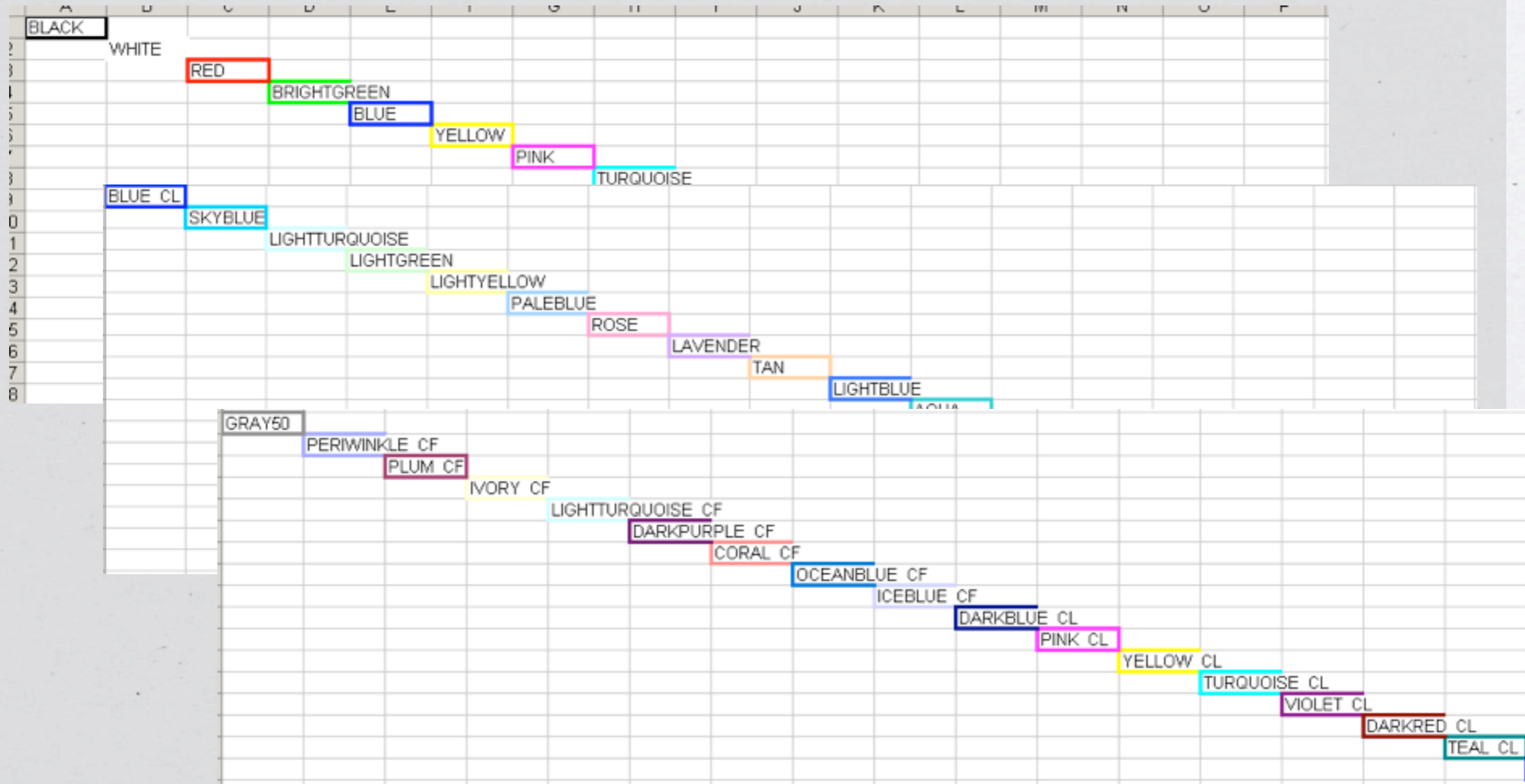
$oClass = new ReflectionClass('ExcelFormat');
foreach ($oClass->getConstants() as $c => $val) {
    if (strpos($c, 'COLOR_') !== 0) {
        continue;
    }
    $format = $x->addFormat();
    $format->borderColor($val);
    $format->borderStyle(ExcelFormat::BORDERSTYLE_THICK);

    $name = preg_replace('!.*_!U', '', $c);

    $s->write($row++, $col++, $name, $format);
}

$x->save("borders.xls");
```

Output



Specific Border Functions

BorderLeftStyle

BorderLeftColor

BorderRightStyle

BorderRightColor

BorderTopStyle

BorderTopColor

BorderBottomStyle

BorderBottomColor

Text Styles

```
$x = new ExcelBook();
$s = $x->addSheet("Text Stuff...");

$font_styles = array('Arial', 'Helvetica', 'Courier',
                    'Times New Roman', 'Tahoma', 'Courier New');

foreach ($font_styles as $style) {
    $font = $x->addFont();
    $font->name($style);

    $format = $x->addFormat();
    $format->setFont($font);

    $s->write($row, $col++, $style, $format);
}
$x->save("font.xls");
```

	A	B	C	D	E	F	G
1	Arial	Helvetica	Courier	Times New Roman	Tahoma	Courier New	

Text Styles

```
$x = new ExcelBook();  
$s = $x->addSheet("Text Stuff...");
```

```
$underline_types = array(  
    ExcelFont::UNDERLINE_NONE, ExcelFont::UNDERLINE_SINGLE,  
    ExcelFont::UNDERLINE_DOUBLE, ExcelFont::UNDERLINE_SINGLEACC,  
    ExcelFont::UNDERLINE_DOUBLEACC  
);
```

```
foreach ($underline_types as $style) {  
    $font = $x->addFont();  
    $font->underline($style);
```

```
    $format = $x->addFormat();  
    $format->setFont($font);
```

```
    $s->write($row, $col++, $style, $format);
```

```
}  
$x->save("font.xls");
```

Underline value	View
UNDERLINE_NONE	UNDERLINE_NONE
UNDERLINE_SINGLE	<u>UNDERLINE_SINGLE</u>
UNDERLINE_DOUBLE	<u><u>UNDERLINE_DOUBLE</u></u>
UNDERLINE_SINGLEACC	<u>UNDERLINE_SINGLEACC</u>
UNDERLINE_DOUBLEACC	<u><u>UNDERLINE_DOUBLEACC</u></u>

More Text Stuff...

```
$x = new ExcelBook();  
$s = $x->addSheet("Text Stuff...");  
  
foreach (array('italics', 'bold', 'strike') as $style) {  
    $font = $x->addFont();  
    $font->$style(true);  
  
    $format = $x->addFormat();  
    $format->setFont($font);  
  
    $s->write($row, $col++, $style, $format);  
}  
$x->save("font.xls");
```

	A	B	C
1	<i>italics</i>	bold	strike

There is more still...

```
$x = new ExcelBook();  
$s = $x->addSheet("Text Stuff...");  
  
foreach (array('NORMAL', 'SUPERSCRIPT', 'SUBSCRIPT')  
as $style) {  
    $font = $x->addFont();  
    $font->mode(constant("ExcelFont::" . $style));  
  
    $format = $x->addFormat();  
    $format->setFont($font);  
  
    $s->write($row, $col++, $style, $format);  
}  
$x->save("font.xls");
```

	A	B	C
1	NORMAL	SUPERSCRIPT	SUBSCRIPT
2			


```

$x = new ExcelBook();
$s = $x->addSheet("Text Stuff...");

$colors = array(
    ExcelFormat::COLOR_BROWN, ExcelFormat::COLOR_RED,
    ExcelFormat::COLOR_BRIGHTGREEN, ExcelFormat::COLOR_ICEBLUE_CF,
);

for ($i = 0; $i < 4; $i++) {
    $font = $x->addFont();
    // valid range is from 0 > 40
    $font->size($i * $i);

    $font->color(array_pop($colors));

    $format = $x->addFormat();
    $format->setFont($font);

    $s->write($row, $col++, "ABC 123", $format);
}
$x->save("font.xls");

```

	A	B	C	D
1	ABC 123		ABC 123	ABC 123

Parsing Documents

```
$x = new ExcelBook();
```

```
$x->loadFile("file.xls");
```

```
$s = $x->getSheet();
```

```
for ($i = 0, $e = $s->lastRow(); $i < $e; $i++) {  
    print_r(array_filter($s->readRow($i)));  
}
```

Precision Parsing

```
$x = new ExcelBook();  
$x->loadFile("file.xls");  
$s = $x->getSheet();  
  
// simple 1 cell read  
$data = $s->read(0, 0);  
  
//read data & obtain format  
$data = $s->read(0, 0, $format);  
  
// $format == ExcelFormat instance
```

Other stuff you can do

- * Freeze Panes
- * Manipulate Excel Sheets
- * Create custom number formats
- * Set printing layouts
- * Copy Data
- * Configure Cell Layout
- * Zoom Controls
- * Split Sheets
- * Text Wrapping
- * Text Alignment
- * Lock & Freeze cells
- * Etc....

Thanks for Listening

Please leave feedback

at: <http://joind.in/7600>

Ilia Alshanetsky

<http://ilia.ws/>

@iliaa