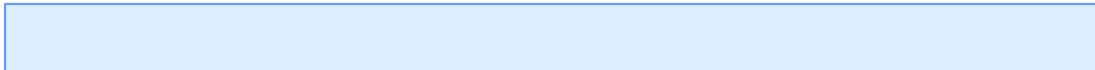


Generer vos fichiers rapidement avec l'extension PHPExcel

par Yannick Komotir (y-komotir.developpez.net)

Date de publication : 12 août 2011

Dernière mise à jour :



I - Introduction.....	3
II - Installation.....	3
III - Structures des classes.....	4
III-A - la classe ExcelBook.....	4
III-A-1 - Structure de la classe.....	4
III-A-2 - Exemples.....	5
III-B - La classe ExcelSheet.....	6
III-B-1 - Structure de la classe.....	6
III-B-2 - Exemples.....	7
III-C - La classe ExcelFont.....	8
III-C-1 - Structure de la classe.....	8
III-C-2 - Exemple.....	9
III-D - La classe ExcelFormat.....	9
III-D-1 - Structure de la classe.....	9
III-D-2 - Exemples.....	10
IV - Conclusion.....	10

I - Introduction

PHP Excel est une extension PHP basé sur l'implémentation de la bibliothèque C libXL, il permet de générer des fichiers excels sur n'importe quelle plateforme supportant PHP sans aucune dépendance avec une rapidité impressionnante sans une consommation mémoire excessive pouvant générer 10000 cellules en une fraction de seconde !

Quelques autres avantages sont :

- Dispose d'une interface orienté objet
- Support native de l'unicode
- Prise en charge de Excel 97-2003 avec le format xls
- Prise en charge de Excel 2007 et 2010 avec le formatxlsx
- Supporte la mise en forme des cellules et la prise en charge des fonctions
- Homogénéité de son api avec d'autres langages comme le C++, le Delphi ou le .NET
- La bibliothèque est vraiment très légère

Le petit inconvénient est au sujet du type de licence fourni avec libXL qui est commerciale, la version gratuite de cette librairie contient une bannière plus ou moins gênante sur la première ligne du fichier excel généré.

Created by LibXL trial version. Please buy the LibXL full version for removing this message.

L'extension PHP Excel diffère des autres librairies que vous avez probablement rencontrée par le fait qu'elle est une extension PHP écrite en C offrant une API au langage.

II - Installation

Pour travailler vous avez besoin d'une installation PHP version VC9 supérieur ou égal à 5.2, PHP Excel est gratuitement disponible sur github, vous pouvez trouver la dernière version windows (exclusivement VC 9) sur le site <http://downloads.php.net/pierre/> une fois l'archive de l'extension téléchargée dézipper le, renommez le en php_excel.dll si ce n'est pas le cas. Ensuite placez le dans le répertoire ext de votre installation windows et modifiez votre fichier de configuration php.ini en y ajoutant la ligne suivante:

```
extension=php_excel.dll
```

Une fois cela fait rendez vous à l'adresse <http://www.libxl.com/download.html> afin de télécharger la dll de la librairie libXL propre à votre plateforme (windows dans mon cas).

Dézippez l'archive et copiez la dll résultante dans un répertoire inscrit dans la variable d'environnement de votre système (PATH sur les systèmes Windows). Dans le cas où vous avez déjà votre PHP présent dans la variable d'environnement, placez simplement la dll dans le même dossier que vos binaires PHP (généralement dans le répertoire que le fichier de configuration PHP php.ini).

Exécuter PHP info afin de vous rassurer de la bonne marche de l'extension

```
<?php  
phpinfo();
```

Cela devrait afficher ceci dans le résultat du phpinfo

excel

excel support	enabled
Excel Version	0.9.1

III - Structures des classes

L'extension PHP Excel ajoute quatre classes à votre environnement PHP. Dans cette section nous présentons la structure de ces classes que nous ne décrirons pas en détail. Les noms des méthodes étant explicites vous comprendrez leur utilité. Pour certaines classes ayant plusieurs constantes, nous afficherons leurs noms en bloc. Cette section de l'article ne se présente pas comme étant ou faisant partie de la documentation officielle, elle se base sur la version de l'extension au moment de l'écriture de l'article.

- la classe ExcelBook
- la classe ExcelSheet
- la classe ExcelFont
- et la classe ExcelFormat

III-A - la classe ExcelBook

III-A-1 - Structure de la classe

```
function void __construct(string $license_name, string $license_key, bool $excel_2007 = false)
function int activeSheet(int $sheet_number = null)
function int addCustomFormat(string $format_string)
function ExcelFont addFont(ExcelFont $font = null)
function ExcelFormat addFormat(ExcelFormat $format)
function int addPictureFromFile(string $filename)
function int addPictureFromString(string $data)
function ExcelSheet addSheet(string $name)
function int colorPack(int $red, int $green, int $blue)
function array colorUnpack(int $color)
function ExcelSheet copySheet(string $name, int $sheet_number)
function bool deleteSheet(int $sheet_number)
function int getActiveSheet()
function ExcelFormat getAllFormats()
function string getCustomFormat(int $id)
function array getDefaultFont()
function string getError()
function ExcelSheet getSheet(int $sheet_number = 0)
function bool load(string $data)
function bool loadFile(string $filename)
function float packDate(int $timestamp)
function bool rgbMode()
function mixed save(string $filename = null)
function bool setActiveSheet(int $sheet_number)
function void setDefaultFont(string $font_name, string $font_size)
function void setLocale(string $locale)
function void setRGBMode(bool $mode)
function int sheetCount()
function int unpackDate(float $date)
```

III-A-2 - Exemples

Exemple 1

```
//donner les bonnes valeurs si vous avez acheter une license
$license_name = NULL;
$license_key = NULL;

//création d'un document Excel de type 97-2003

$book = new ExcelBook( $license_name, $license_key );
//si $license_name = $license_key = NULL la déclaration peut être simplifier en
//$book = new ExcelBook();

/*
    différents traitements
*/

//sauvegarde dans un fichier
$book->save( 'chemin-fichier.xls' );

//ou affichage dans un navigateur web
header( 'Content-type: application/vnd.ms-excel' );
header( 'Content-Disposition: attachment; filename="fichier-excel-97-2003.xls"' );
$book->save( 'php://output' );
```

Exemple 2

```
//donner les bonnes valeurs si vous avez acheter une license
$license_name = NULL;
$license_key = NULL;

//création d'un document Excel de type 2007

$book = new ExcelBook( $license_name, $license_key, TRUE );
//si $license_name = $license_key = NULL la déclaration peut être simplifier en
//$book = new ExcelBook();

/*
    différents traitements
*/

//sauvegarde dans un fichier
$book->save( 'chemin-fichier.xlsx' );

//ou affichage dans un navigateur web
header( 'Content-type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet' );
header( 'Content-Disposition: attachment; filename="nom-fichier-excel-2007.xlsx"' );
$book->save( 'php://output' );
```

Exemple 3

```
$book = new ExcelBook();

//chargement du fichier existant
$book->loadFile( 'bureautique/fichier-demo.xls' );

/*
    différents traitements
*/
```

III-B - La classe ExcelSheet

La classe ExcelSheet est la plus utilisée dans cette extension, elle contient la quasi totalité des méthodes qui vous permettrons de travailler sur votre fichier excel.

III-B-1 - Structure de la classe

Méthodes de la classe

```
function void addPictureDim(int $row, int $column, int $picture_id, int $width, int $height)
function void addPictureScaled(int $row, int $column, int $picture_id, float $scale)
function void cellFormat(int $row, int $column)
function int cellType(int $row, int $column)
function void clear(int $row_start, int $row_end, int $column_start, int $column_end)
function bool clearPrintArea()
function bool clearPrintRepeats()
function float colWidth(int $column)
function void copy(int $row_from, int $column_from, int $row_to, int $column_to)
function bool delNamedRange(string $name)
function bool deleteMerge(int $row, int $column)
function bool displayGridlines()
function int firstCol()
function int firstRow()
function string footer()
function float footerMargin()
function bool getGroupSummaryBelow()
function bool getGroupSummaryRight()
function array getMerge(int $row, int $column)
function bool groupCols(int $column_start, int $column_end, bool $collapse = false)
function bool groupRows(int $row_start, int $row_end, bool $collapse = false)
function bool hcenter()
function string header()
function float headerMargin()
function int horPageBreak(int $row, bool $break)
function bool insertCol(int $column_start, int $column_end)
function bool insertRow(int $row_start, int $row_end)
function bool isDate(int $row, int $column)
function bool isFormula(int $row, int $column)
function bool landscape()
function int lastCol()
function int lastRow()
function float marginBottom()
function float marginLeft()
function float marginRight()
function float marginTop()
function string name()
function int paper()
function bool printGridlines()
function bool printHeaders()
function bool protect()
function mixed read(int $row, int $column, ExcelFormat $format = null)
function array readCol(int $column, int $row_start = 0, int $row_end = null)
function string readComment(int $row, int $column)
function array readRow(int $row, int $column_start = 0, int $column_end = null)
function bool removeCol(int $column_start, int $column_end)
function bool removeRow(int $row_start, int $row_end)
function float rowHeight(int $row)
function void setCellFormat(int $row, int $column, ExcelFormat $format)
function setColWidth(int $column_start, int $column_end, float $width, bool $hidden = false,
    ExcelFormat $format = null)
function void setDisplayGridlines(bool $value)
function bool setFooter(string $footer, float $margin)
function bool setGroupSummaryBelow(bool $direction)
function bool setGroupSummaryRight(bool $direction)
function bool setHeader(string $header, float $margin)
function void setLandscape(bool $value)
```

Méthodes de la classe

```
function void setMarginBottom(float $margin)
function void setMarginLeft(float $margin)
function void setMarginRight(float $margin)
function void setMarginTop(float $margin)
function bool setMerge(int $row_start, int $row_end, int $column_start, int $column_end)
function void setName(string $name)
function bool setNamedRange(string $name, int $row_from, int $column_from, int $row_to, int $column_to)
function void setPaper(int $paper)
function void setPrintGridlines(bool $value)
function void setPrintHeaders(bool $value)
function bool setPrintRepeatCols(int $column_start, int $column_end)
function bool setPrintRepeatRows(int $row_start, int $row_end)
function void setProtect(bool $value)
function setRowHeight(int $row, float $height, ExcelFormat $format = null, bool $hidden = false)
function void setVCenter(bool $value)
function void setZoom(int $value)
function void setZoomPrint(int $value)
function void splitSheet(int $row, int $column)
function bool vcenter()
function bool verPageBreak(int $column, bool $break)
function bool write(int $row, int $column, mixed $data, ExcelFormat $format = null,
    int $data_type = -1)
function bool writeCol(int $column, array $data, int $row_start = 0, ExcelFormat $format = null)
function bool writeComment(int $row, int $column, string $comment, string $author, int $width,
    int $height)
function bool writeRow(int $row, array $data, int $column_start = 0, ExcelFormat $format = null)
function int zoom()
function int zoomPrint()
```

Constantes de la classe

```
//taille papier, doivent être utiliser comme argument pour les méthodes setPaper(), et comme valeur de retour pour
PAPER_*

//type de cellule, doivent être utiliser comme valeur de retour pour la méthode cellType
CELLTYPE_*

//type d'erreurs
ERRORTYPE_*
```

Ne peut être instancier que par les méthodes addSheet, copySheet et getSheet de la classe ExcelBook. les lignes et colonnes sont indicés à partir de 0, la cellule excel (A,1) correspondra donc à (0,0)



Nous n'avons pas énumérées toutes les constantes dans certains cas; considérant qu'elles forment des groupes vu leur utilisation. Vous voir toutes les constantes d'une classe en exécutant le code suivant

```
$reflection = new ReflectionClass( 'ExcelSheet' );

$constates = $reflection->getConstants();

foreach( $constates as $key => $constate ){
    echo $key. '=' . $constate. '<br />';
}
```

III-B-2 - Exemples

créer un classeur simple

```
$book = new ExcelBook();

$sheet = $book->addSheet( "Classeur 1" );
```

créer un classeur simple

```
for( $i=1 ; $i < 100; $i++ ){
    $sheet->write( $i, 0, $i*2 );
}

$book->save( 'chemin-fichier.xls' );
```

Lecture d'un fichier excel dont on ne connaît pas la structure supposant que ce fichier ne contient pas d'objets de type images, sons...

```
$book = new ExcelBook();

//chargement du fichier existant
$loadfile = $book->loadFile( 'C:\\Users\\y\\Documents\\t_clients_results.xls' );

if( $loadfile ){
    $sheet = $book->getSheet();

    echo '<table>';
    for( $i = $sheet->firstRow(); $i < $sheet->lastRow(); $i++ ){
        echo '<tr>';

        for( $j = $sheet->firstCol(); $j < $sheet->lastCol(); $j++ ){
            echo "<td>{$sheet->read( $i, $j )}</td>";
        }
        echo '</tr>';
    }
    echo '</table>';
}
```

Insertion d'une image dans un classeur

```
$book = new ExcelBook();

$sheet = $book->addSheet( "Classeur 1" );

//ajout de l'image dans le classeur
$idlogo = $book->addPictureFromFile( 'logo-entreprise.jpg' );

//insertion de l'image dans une cellule
$sheet->addPictureScaled( 1, 0, $idlogo, 1 );

$book->save( 'chemin-fichier.xls' );
```

III-C - La classe ExcelFont

III-C-1 - Structure de la classe

Méthodes de la classe

```
function __construct(ExcelBook $book)
function bool bold(bool $bold = null)
function int color(int $color = null)
function bool italics(bool $italics = null)
function int mode(int $mode = null)
function string name(string $font_name = null)
function int size(int $size = null)
function bool strike(bool $strike = null)
function int underline(int $underline = null)
```

Constantes de la classe

```
//constante s'appliquant comme argument à la methode mode()
NORMAL
SUPERScript
```


Constantes de la classe

SUBSCRIPT

```
//type de soulignement s'applique comme argument à la méthode underline
UNDERLINE_NONE
UNDERLINE_SINGLE
UNDERLINE_DOUBLE
UNDERLINE_SINGLEACC
UNDERLINE_DOUBLEACC
```

III-C-2 - Exemple

Mise en forme de la police du texte

```
$book = new ExcelBook();

$idlogo = $book->addPictureFromFile( 'android-small.jpg' );
$sheet = $book->addSheet( "Classeur 1" );

//methode d'instantiation ExcelFont #1
$fontpresentation = new ExcelFont( $book );
$fontpresentation->size( 15 );
$fontpresentation->bold( true );
$fontpresentation->underline( ExcelFont::UNDERLINE_DOUBLE );
$fontpresentation->name( 'ravie' );

$formatpresentation = new ExcelFormat( $book );
$formatpresentation->setFont( $fontpresentation );

//methode d'instantiation ExcelFont #2
$fontadresse = $book->addFont();
$fontadresse->size( 11 );
$fontadresse->italics( true );
$fontadresse->name( 'tahoma' );

$formatadresse = new ExcelFormat( $book );
$formatadresse->setFont( $fontadresse );

//Mise en place
$sheet->addPictureScaled( 1, 0, $idlogo, 1 );
$sheet->write( 2, 0, 'Best WEB SOLUTIONS Inc', $formatpresentation );
$sheet->write( 3, 0, "24 Rue des italics - face à l'immeuble Gauthier", $formatadresse );

$book->save( 'chemin-fichier.xls' );
```

III-D - La classe ExcelFormat

la classe ExcelFormat joue un rôle capitale dans la mesure où elle permet la mise en forme des cellules, le formatage des nombres, la manipulation des formules.

III-D-1 - Structure de la classe

Méthodes de la classe

```
function __construct(ExcelBook $book)
function int borderBottomColor(int $color = null)
function int borderBottomStyle(int $style = null)
function int borderColor(int $color = null)
function int borderDiagonalColor(int $color = null)
function int borderDiagonalStyle(int $style = null)
function int borderLeftColor(int $color = null)
function int borderLeftStyle(int $style = null)
function int borderRightColor(int $color = null)
```

Méthodes de la classe

```
function int borderRightStyle(int $style = null)
function int borderStyle(int $style = null)
function int borderTopColor(int $color = null)
function int borderTopStyle(int $style = null)
function int fillPattern(int $pattern = null)
function ExcelFont getFont()
function bool hidden(bool $hidden = null)
function int horizontalAlign(int $halign_mode = null)
function int indent(int $indent = null)
function bool locked(bool $locked)
function int numberFormat(int $number_format)
function int patternBackgroundColor(int $color = null)
function int patternForegroundColor(int $color = null)
function int rotate(int $angle = null)
function bool setFont(ExcelFont $font)
function bool shrinkToFit(bool $shrink = null)
function int verticalAlign(int $valign_mode = null)
function bool wrap($wrap = null)
```

Constantes

```
//constates de couleurs, doit être utiliser partout on l'on fait appel a un code de couleur
COLOR_*

//type format de cellule
AS_DATE
AS_FORMULA
AS_NUMERIC_STRING

//type de format numérique, doivent être utiliser comme argument pour la méthode numberFormat
NUMFORMAT_*

//type d'alignement horizontal, doivent être utiliser comme argument pour la méthode horizontalAlign
ALIGNH_*

//type d'alignement vertical, doivent être utiliser comme argument pour la méthode verticalAlign
ALIGNV_*

//style de bordure, doivent être utiliser comme argument pour la méthode borderTopStyle, borderStyle,
//borderRightColor, borderLeftStyle et borderBottomStyle
BORDERSTYLE_*

//style de bordure diagonal, doivent être utiliser comme argument pour la méthode borderDiagonalStyle
BORDERDIAGONAL_*

//type de remplissage, doivent être utiliser comme argument de la methode fillPattern
FILLPATTERN_*
```

III-D-2 - Exemples

Exemple 1

Exemple 2

Exemple 3

IV - Conclusion