Excel Export mit PEAR::Spreadsheet_Excel_Writer

(25.02.2003)

In vielen Fällen wünscht sich der Kunde einen **Excel Export** der mit der PHP Applikation dargestellten Daten. In vielen Fällen erfolgt die Lösung über separierte CSV-Dateien, die zum Download angeboten und von Excel importiert werden können.

Das Problem bei CSV-Dateien ist aber zum Beispiel, dass man keine Daten damit transportieren kann, die Zeilenumbrüche enthalten. Außerdem geht hier jede Möglichkeit der Integration von Grafiken, Formeln, Schriftwahl, Zellenvorder- und -hintergründe, mehrere Worksheets und ähnliches verloren.

Das in PEAR enthaltene Package **PEAR::Spreadsheet_Excel_Writer** möchte hiermit aufräumen. Es schreibt die Daten im so genannten BIFF5 Format (eine Binärdarstellung von Daten in sequenzieller Form), das von Excel gelesen werden kann. Das Package bietet nicht nur das Schreiben der Daten, sondern ebenfalls das Erzeugen mehrerer Worksheets, Einpacken von Formeln und Integration von Grafiken, Festlegung von Druckrändern und mehr.

Die Installation gestaltet sich hierbei recht einfach. Mit einem

```
pear install Spreadsheet_Excel_Writer
```

wird das Package heruntergeladen sowie installiert und steht fortan mit einem

```
<?php
include_once 'Spreadsheet/Excel/Writer.php';
?>
```

zur Verfügung.

Wie sich der Dokumentation auf der Seite des Autors entnehmen läßt, so ist das Package eine Portierung des aus Perls CPAN bekannten Spreadsheet::Writer Moduls, angepaßt an die PEAR Coding Standards (Studly Caps Funktionen etc.).

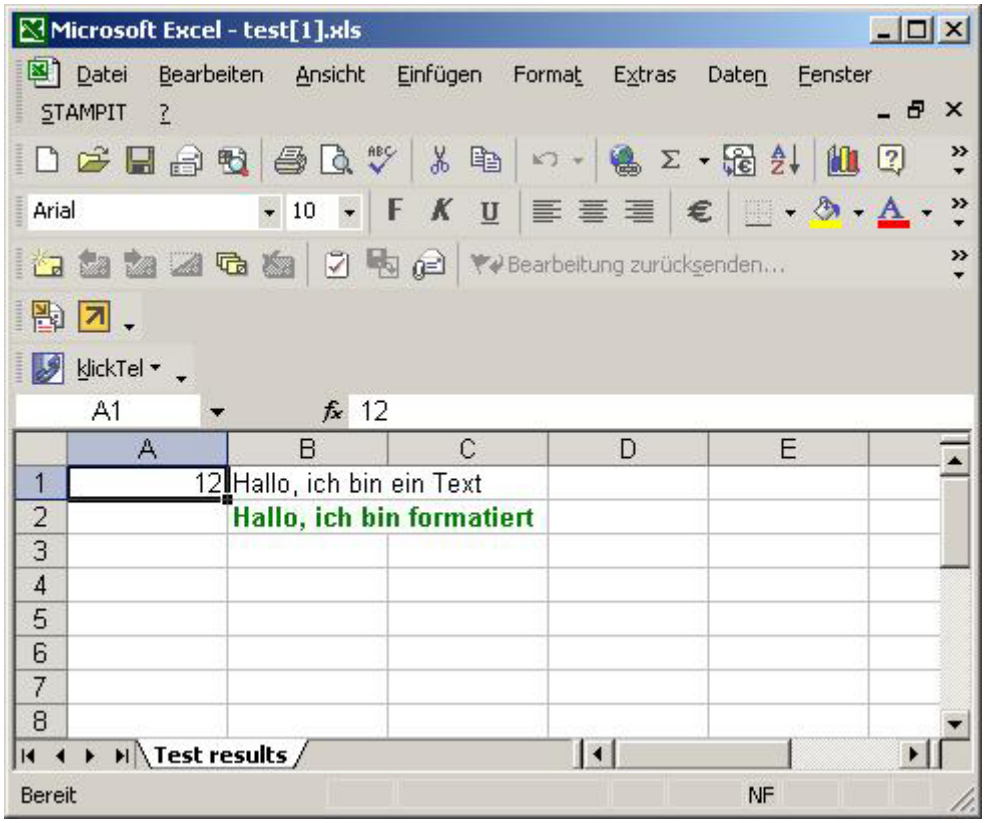
Auch die Verwendung des Pakets gestaltet sich einfach:

```
<?php
include_once "Spreadsheet/Excel/Writer.php";

$xls =& new Spreadsheet_Excel_Writer();
$xls->send("test.xls");
$sheet =& $xls->addWorksheet('Test results');
$sheet->write(0,0,12);
$sheet->write(0,1,"Hallo, ich bin ein Text");
$format =& $xls->addFormat();
$format->setBold();
$format->setColor("green");
$sheet->write(1,1,"Hallo, ich bin formatiert",$format);
$xls->close();
?>
```

Zunächst wird die Klasse instanziiert, mit der Methode send() werden die entsprechenden Header zum Browser zum Download der Datei gesendet und danach ein neues Worksheet hinzugefügt, das auch einen Namen tragen kann. Beachten Sie hierbei die Referenzangabe, die Factory addWorksheet() liefert ein neues Objekt zurück, auf dem man arbeiten kann. Die Methode write() sollte selbsterklärend sein: hier wird auf einer Matrix gearbeitet, erster Parameter ist die Zeile (0 = 1. Zeile), 2. Parameter die Spalte (0 = Spalte A) und der 3. Parameter ist entweder eine Zahl oder z.B. ein String.

Besondere Beachtung verdient hierbei die Factory addFormat(). Sie liefert ein Formatierungsobjekt zurück, dem man verschiedene Eigenschaften verpassen kann. Im Beispiel wird hier eine BOLD Formatierung (setBold()) sowie die Farbe auf Grün gesetzt. Dieses Formatierungsobjekt kann man als vierten, optionalen Parameter der Methode write() angeben, und somit wird diese Zelle bzw. deren Text entsprechend fett und in grün dargestellt.

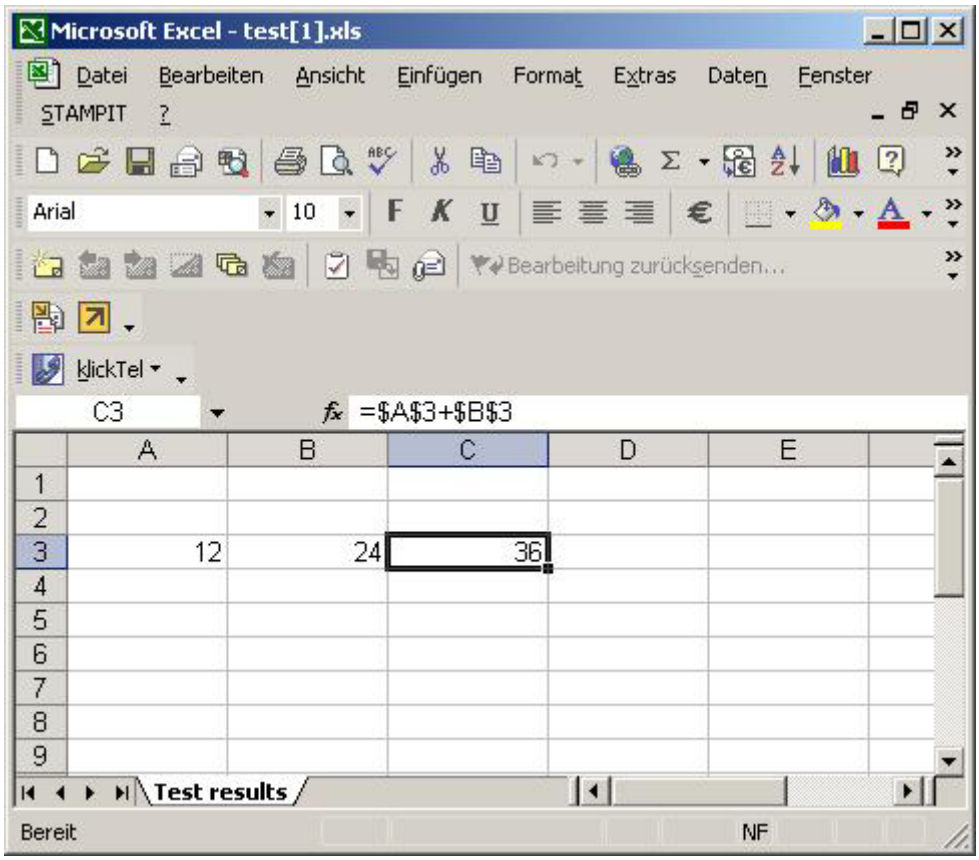


Natürlich möchte man unter Umständen auch Formeln speichern. Dazu kennt die Klasse die Methode writeFormula, die die Position der Zelle sowie die Formel übergeben bekommt. Ein kurzes und schmerzloses Beispiel soll die Verwendung verdeutlichen:

```
<?php
include_once "Spreadsheet/Excel/Writer.php";

$xls =& new Spreadsheet_Excel_Writer();
$xls->send("test.xls");
$sheet =& $xls->addWorksheet('Test results');
$sheet->write(2,0,12);
$sheet->write(2,1,24);
$sheet->writeFormula(2,2,"=A3+B3");
$xls->close();
?>
```

Das Ergebnis ist ebenso einfach wie verblüffend:



Zum Schluß noch die beiden Files zum Download: **excelexport.zip**

Ich wünsche Ihnen viel Spaß beim Ausprobieren. :-)
(bs)