Tipps und Tricks

- UTF-8, BOM und das Windows Notepad
- Dateien und Verzeichnisse rekursiv verarbeiten

UTF-8, BOM und das Windows Notepad

Für einen Kunden habe ich ein Tool in Perl programmiert. Es überprüft alle paar Minuten ein Verzeichnis in das von Redakteuren Fotos und eine Textdatei mit Metainformationen hochgeladen werden. In der Textdatei werden z.B. Bildunterschriften und Copyrights übergeben. Das Tool erstellt dann aus den Metainformationen und den Dateinamen der Bilder eine XML-Datei, welche dann zusammen mit den Bildern auf das Redaktionssystem zur Weiterverarbeitung kopiert wird.

Es gab allerdings Probleme beim Verknüpfen der Daten mit einem Bild und zwar immer nur mit den Infos, die in der Textdatei in der ersten Zeile standen. Vom Kunden habe ich dann erfahren, dass die Redakteure diese Datei mit dem Windows 7 Notepad erstellen (ob es mit anderen Windows-Versionen die gleichen Probleme gibt kann ich nicht sagen).

Die Windows-Zeilenumbrüche, die Notepad in Textdateien einbaut, konnten nicht schuld sein, da die Datei im CSV-Format kommt und der Zeilenumbruch nur auf das letzte Feld zutreffen würde, welches die Bildunterschrift enthält. Dieser Umbruch würde also einfach in die Beschreibung übernommen werden.

Als ich mir die Textdatei mit vi angesehen habe, konnte ich nicht entdecken, warum es nicht funktioniert. Erst bei Betrachtung mit „cat -A datei.txt“ habe ich folgendes entdeckt:

```
M-oM-;M-?01.JPG;Hippel;Ein erster Testdurchgang zwecks FTP-Upload^M$
02.JPG;Hippel;Ein erster Testdurchgang zwecks FTP-Upload^M$
03.JPG;Hippel;Ein erster Testdurchgang zwecks FTP-Upload^M$
04.JPG;Hippel;Ein erster Testdurchgang zwecks FTP-Upload
```

Das Problem war also gefunden. Am Anfang der ersten Zeile haben sich ein paar seltsame Zeichen eingeschlichen: „M-oM-;M-?“. Nach ein wenig Recherche hab ich dann herausgefunden, das es sich hier das sogenannte Byte Order Mark (BOM) von UTF-codierten Dateien handelt. Bei 2- und 4-bytigen UTF-Kodierungen ist dieses BOM unbedingt erforderlich, bei UTF-8 codierten (wie sie Windows-Notepad erstellt) jedoch nicht, da diese nur mit einem Byte dargestellt werden.

In Perl habe ich mir damit geholfen, dass ich bei so codierten Dateien das BOM einfach entferne. Dazu müssen die ersten drei Bytes (EF BB BF) der Datei entfernt werden. Das sieht dann so aus:

```
open(META, "<", "meta.txt") || logger("ERROR", "could not open meta.txt: $!\n") && die();
my $x = 0;
while(<META>) {
  # remove the BOM information from UTF-8 encoded textfiles coming from windooze notepad
  if ($x == 0) {
    s/^\xEF\xBB\xBF//;
  }
  # remove windows linebreakes
  s/\r\n/\n/;
  chomp;
  $meta[$x] = $_;
  $x++;
```

```
  logger("DEBUG", "meta-data: " . $_ . "\n");
}
close(META);
```

Dateien und Verzeichnisse rekursiv verarbeiten

Diese Snippet verarbeitet Dateien und Verzeichnisse und steigt dabei auch in Unterverzeichnisse ein. Gibt den Filenamen aus wenn eine Datei gefunden wurde. Wenn ein Verzeichnis gefunden wurde wird der Name aufgerufen und DoDir rekursiv aufgerufen.

```
my $root = "/home/user";
DoDir($root);

sub DoDir {
  my $dir = shift;
  my $file;
  opendir(DIR, $dir) || die "Unable to open $dir: $!";
  my(@files) = grep {!/^\.\.?$/ } readdir(DIR);
  closedir(DIR);
  foreach (@files) {
    if (-d ($file = "$dir\\$_")) {
      print "Found a directory: $file\n";
      DoDir($file);
    } else {
      print "File: $file\n";
    }
  }
}
```

für UNIX muss etwas umgebaut werden:

```
my $root = "/home/user";
DoDir($root);

sub DoDir {
  my $dir = shift;
  my $file;
  opendir(DIR, $dir) || die "Unable to open $dir: $!";
  my(@files) = grep {!/^\.\.?$/ } readdir(DIR);
  closedir(DIR);
  foreach (@files) {
    if (-d ($file = "$dir/$_")) {
```

```
      print "Found a directory: $file\n";
      DoDir($file);
    } else {
      print "File: $file\n";
    }
  }
}
```