

Writing a UTF-8 file

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2007-01/msg00235.html>

- *From:* "pearly" <michael_perle@xxxxxxxx>
 - *Date:* 5 Jan 2007 07:37:31 -0800
-

Hello everybody,

Does anyone know how I can write UTF-8 files without a BOM in Perl?

Whether I open files in utf8 mode (2nd parameter of open or via binmode) I always end up with

- A BOM "FF FE" (UTF-16LE afaik) at the start of the output file;
- Encoding with minimum 2 bytes per character.

I am reading strings from an external resource, so the following is not 100% representative but has the same effect:

```
my $string_with_special_chars = "Château Müller\nGarçon";
# String contains entities acirc, uuml and ccedil.
open F, ">:utf8", "test.txt";
print F $string_with_special_chars;
```

Tried it both on Linux (Perl 5.8.6) and Windows (Perl 5.8.7).

Difference between utf8 and default mode:

The file created without explicit utf8 mode is readable in Firefox (UTF-8 encoding). My hex editor shows that for all characters the 2nd byte is 0x00.

The file opened with ">:utf8" shows hex C3 00 A2 00 for the u umlaut resp. in total 6 bytes more due to the 3 special chars.

Where does the BOM 0xFF 0xFE come from?

Why does Perl add it?

Doesn't Perl write UTF-8 by default?

Why adding the BOM and why 2 or more bytes per character?

Puzzeling since ages (ok, days) on this.

Thank you for any hints.

MP

.