

Re: Is there a standard way to write an 8 bit byte?

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2005-05/msg00128.html

- *From:* "Malcolm" <regnizar@xxxxxxxxxxxxxxxx>
 - *Date:* Mon, 2 May 2005 21:23:55 +0000 (UTC)
-

"Kristian Nybo" <kristian@xxxxxxxxxxxxxxxx> wrote
> I'm writing a simple image file exporter as part of a school project, and
> I would like to write completely platform-independent code if at all
> possible. The problem I've run into is that apparently there's no
> guarantee that a particular implementation will have a data type that is
> exactly 8 bits wide; char is only guaranteed to be at least 8 bits.
> Correct me if I'm wrong, but I assume that if CHAR_BIT happens to be, say,
> 9, writing a char into a file in binary mode would result in 9 bits being
> written.
>
Just use fgetc and fputc to write integers between 0 and 255.

There is no cast iron guarantee in the C standard that this will produce the file you want, if say chars are 9 bits wide. However in practise there will always be an easy way to convert such files so that they can be read by 8-bit machines (maybe by running them through a converter program).. Computers exist in the real world and have to communicate with each other.

-
- *Follow-Ups:*
 - ◆ **[Re: Is there a standard way to write an 8 bit byte?](#)**
◇ *From:* Kristian Nybo
 - *References:*
 - ◆ **[Is there a standard way to write an 8 bit byte?](#)**
◇ *From:* Kristian Nybo
 - Prev by Date: **[Re: Checking if socket is still open?](#)**
 - Next by Date: **[Re: dynamic char - realloc use](#)**
 - Previous by thread: **[Re: Is there a standard way to write an 8 bit byte?](#)**
 - Next by thread: **[Re: Is there a standard way to write an 8 bit byte?](#)**
 - Index(es):
 - ◆ **[Date](#)**
 - ◆ **[Thread](#)**