

## Re: [Q] Text vs Binary Files

**Source:** <http://coding.derkeiler.com/Archive/General/comp.programming/2004-05/1797.html>

---

**From:** Arthur J. O'Dwyer (*ajo\_at\_nospam.andrew.cmu.edu*)

**Date:** 05/27/04

Date: Thu, 27 May 2004 12:11:01 -0400 (EDT)

On Thu, 27 May 2004, Eric wrote:

>

> Arthur J. O'Dwyer <*ajo@nospam.andrew.cmu.edu*> wrote:

> >

> > Ah, but what do you do when the XML standard changes? :)

>

> Please correct me if I am wrong, but the design of XML already takes

> this into account. In otherwords, the idea that it can and will change

> is a part of the design – this is one reason why XML is such a nifty

> technology.

Probably true. I don't know much about XML's namespacing rules (by which I mean the rules that say that <foo> is an okay tag for a user to create, but <bar> could be given special meaning by future standards). [If anyone wants to give me a lecture, that's fine; otherwise, I'll just look it up when I need to know. ;) ]

> > Misunderstood. By "the same systems," I meant the systems I just

> > mentioned: DOS/Win32, Unix, and MacOS. Their binary data formats are

> > identical.

>

> What do you mean by 'their binary data formats are identical'?...this

> would seem to imply that big/little endian issues are a thing of the

> past...?

Yup. The vast majority of computers these days use eight-bit byte-oriented transmission and storage protocols. Whatever bit-ordering problems there are have moved "downstream" to those people involved in the construction of hardware that has to choose whether to transmit bit 0 or bit 7 first (and I'm sure they have their own relevant standards in those fields, too).

Again, I refer you to standards like RFCs 1950, 1951, and 1952 (Google "RFC 1950"). Note the utter lack of concern with the vagaries of the machine. We have indeed moved past big/little-endian wars; now, whoever's[1] writing the relevant standard simply says, "All eggs distributed according to the Fred protocol must be broken at the big end," and that's the end of \*that!\*

comp.programming: Re: [Q] Text vs Binary Files

> > > *Plus, the potential for spending time attempting to figure out why the*  
> > > *@#%@\$ isn't being read properly isn't accounted for here.*  
> >  
> > *Of course not. I/O is trivial.*  
>  
> *Once you track down the problem...however, it would not be uncommon to*  
> *think the problem lies elsewhere first and spend hours before finding*  
> *the trivial fix.*

You misunderstand me. I/O is trivial; thus, after the first five minutes spent making sure the trivial code is correct (which is trivial to prove), you never need to touch it or look at it again. If you never touch it, you can't possibly introduce bugs into it. And if it starts out bugfree (trivially proven), and never has any bugs introduced into it (because it's never modified), then it will remain bugfree forever. (And thus you never need to fix it, trivially or not.)

I'm completely serious and not using hyperbole at all when I say I/O is trivial. It really is.

–Arthur

[1] – In speech I'd say "who'sever writing...," but that looks awful no matter how I spell it. Whosever? Whos'ever? Who's–ever? Yuck. :(