

# Re: Transmitting strings via tcp from a windows c++ client to a Java server

---

*Source:* <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2006-02/msg03465.html>

---

- *From:* "Chris Uppal" <[chris.uppal@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:chris.uppal@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Wed, 22 Feb 2006 12:20:07 -0000
- 

qqq111 wrote:

....

But first a request. /Please/ follow Usenet etiquette and say who you are replying to and quote selectively from the post as you reply. Normally I just ignore people who don't follow "The Rules"; I'm making an exception in this case on a whim ;-)

4. Each of our msgs is indeed preceded by a length field (as fixed-size text field). Length is measured in Java characters and dup by 2 to obtain size in bytes

That algorithm will not give you the size in bytes of a UTF-8 encoded string. There is no way to compute the length of the UTF-8 encoding of a Unicode sequence that does not involve scanning every character. The easiest thing, of course, is just to let the platform do the encoding and then transmit the length of the resulting byte array. If you want to calculate the length yourself, then it's a bit messy — the main problem is that in Java or Windows the input data is encoded as UTF-16 so you have to undo that encoding and then re-encode the result as UTF-8. Not especially difficult, but more work than you might expect if you are used to relying on `strlen()` and the like.

It would work for UTF-16. But if you decide to stick with UTF-8 (which sounds better to me) then I suggest you prototype your receiving code (for both platforms) before you set the protocol in stone.

Whatever you do, make very sure that your documentation (formal or informal) of the protocol is /very/ clear about the meaning of the size field. Remember that the word "character" is ambiguous — it could mean Java `char-s`, C++ `wchar-s`, or (most confusingly) Unicode characters. An inexperienced programmer could even assume it meant "byte".

Re: Transmitting strings via tcp from a windows c++ client to a Java server

5. The BOM issue is, frankly, news to me. If I limit myself to UTF-8 strings only, and stick to standard Win/Java api at both client & server end, do I need to worry about BOM ?

I doubt it. The important thing is to have made a conscious (and documented) decision. I would probably decide that a BOM must not be used, unless there's something in your project's requirements that I don't know about.

— chris