

Re: Non-blocking method for reading writing objects to sockets

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2005-05/msg00678.html>

- *From:* Esmond Pitt <esmond.nospam.pitt@xxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 09 May 2005 08:31:04 GMT
-

Joseph Dionne wrote:

As you may or may not know, not knowing the level of expertise you bring to this discussion, socket stream of non ASCII STREAMS, use two to four bytes preceding the data allowing the received to know the message size and prepare applications to receive the data.

I've only been doing sockets since BSD 4.1, about 1983-4, but if this is supposed to be a statement that is true of all non ASCII streams it just isn't. Some application protocols do it. Some application protocols e.g. XDR do something else. TCP doesn't do **anything** except put the required length field in the IP header. Quite a few application protocols don't prefix message lengths either: Telnet, FTP, and HTTP for starters.

> My original point was that there exist no single socket interface method that removes the responsibility to manage the socket STREAMS like garbage collection has done for memory management.

Well, I thought your original point was 'Technically there are not non blocking socket write methods in any language. There is either room enough in the socket write MCB to hold the data, allowing the method to return, or there is not enough room, and an error is returned to indicate a failure.' There's now so much irrelevance being added such as lectures on threads and further incorrect claims that the original incorrect statement is being obscured, but it's still incorrect.

.