will use either freeware or commercial toolkits to build his/her HL7 messaging application. However, even when dealing with toolkits, a basic understanding of sockets, the MLLP protocol as well as message acknowledgement fundamentals that were covered in this tutorial is required. I am planning on writing a whole series of articles on HL7 programming using a popular HL7 library called "HAPI". So, watch for that on my blog soon.

You can find the full source code used in this tutorial on GitHub here

Conclusion

Building robust HL7 systems capable of running 24/7 and unattended is not easy. For most programmers, once they get past the initial socket-programming hurdles, the bulk of the issues in HL7 2.x-related programming will be around supporting any parsing, translating, formatting and storage requirements. With HL7 3.0* standard starting to come into use increasingly, these issues should go slowly away. However, HL7 2.x standard isn't going anywhere for now. We should expect it to stay at least for 10 more years before they are fully retired. Until then, we will have to continue our quest to build more elegant and sturdier solutions for HL7 messaging.

Tip: You will want to design the system so that the MLLP character groupings are configurable. This way, the customer can change them to whatever they require them to be. You should also permit some configuration around the message acknowledgment functionality such as information that is transmitted about the processing system and its location (goes in the "MSH" segment). Also, a configuration to indicate whether the system is in "test" mode versus "production mode". In the past, things that I have made configurable also include message control id starting number or number pattern, email addresses and error log locations (for error notifications), as well as connection retry attempts.

* - HL7 3.0 (RIM) never came to achieve the wide adoption that many people imagined to occur in the early 2000s and beyond including the replacement of V2-based message interfaces. Please see my article on the V3 standard for more information on what this standard attempted to do, its strengths, weaknesses and its legacy in the now long history of healthcare informatics.
