ASTM E1394 Message Parsing

Theron W. Genaux

Calculating the checksum of a UTF8 encoded LIS01 Frame

* [LIS01A2E](https://clsi.org/standards/products/automation-and-informatics/documents/lis01/), Specification for Low-Level Protocol to Transfer Messages Between Clinical Laboratory Instruments and Computer Systems

<STX>6P|1|PID123456|||Brown^Bobby^B|White|196501020304|U<CR><EXT>62<CR><LF>

An LIS1 frame includes the data and an LIS1 wrapper for transmitting the frame using the LIS1 protocol. Below is the call to create the frame and the parameters in frame number (line 2)

1 byte[] frameBtyes = LIS01\_Helper.CreateFrame(  
2 frameNumber:6,  
3 text:"P|1|PID123456|||Brown^Bobby^B|White|196501020304|U" + C.CR,  
4 frameType:C.ETX,  
5 encoding:Encoding.UTF8);  
  
Frame:  
<STX>6P|1|PID123456|||Brown^Bobby^B|White|196501020304|U<CR><EXT>62<CR><LF>  
  
Bytes:  
02-36-50-7C-31-7C-50-49-44-31-32-33-34-35-36-7C-7C-7C-42-72-6F-77-6E-5E-42-6F-62-62-79-5E-42-7C-57-68-69-74-65-7C-31-39-36-35-30-31-30-32-30-33-30-34-7C-55-0D-3C-45-58-54-3E-36-32-0D-0A

/// <summary>  
 /// Creates an LIS1-A frame  
 /// </summary>  
 /// <param name="frameNumber">single digit Frame Number 0 to 7</param>  
 /// <param name="text">data content of frame</param>  
 /// <param name="frameType">ETX or ETB</param>  
 /// <param name="encoding">Character encoding</param>  
 /// <returns>Returns the encoded frame as byte[]</returns>  
01 public static byte[] CreateFrame(int frameNumber, string text, char frameType, Encoding encoding)  
02 {  
03 string checksum = "00";  
04  
05 string frame = C.STX + frameNumber.ToString() + text + frameType + checksum + C.CR + C.LF;  
06  
07 byte[] frameBytes = encoding.GetBytes(frame);  
08  
09 checksum = LIS01\_Helper.CalculateFrameChecksum(frameBytes);  
10 frameBytes[frameBytes.Length - 4] = (byte)checksum[0];  
11 frameBytes[frameBytes.Length - 3] = (byte)checksum[1];  
12  
13 return frameBytes;  
14 }

byte[] frameBtyes = LIS01\_Helper.CreateFrame(  
 2,  
 "P|1|666|||頭いい^素晴らしい|||U" + C.CR,  
 C.ETX,   
 Encoding.UTF8);  
  
Frame:  
<STX>2P|1|666|||頭いい^素晴らしい|||U<CR><EXT>3D<CR><LF>  
  
Bytes:  
02-32-50-7C-31-7C-36-36-36-7C-7C-7C-E9-A0-AD-E3-81-84-E3-81-84-5E-E7-B4-A0-E6-99-B4-E3-82-89-E3-81-97-E3-81-84-7C-7C-7C-55-0D-03-33-44-0D-0A

byte[] frameBtyes = LIS01\_Helper.CreateFrame(  
 2,  
 "P|1|666|||頭いい^素晴らしい|||U" + C.CR,  
 C.ETX,   
 Encoding.GetEncoding("Shift-JIS"));  
  
Frame:  
<STX>2P|1|666|||頭いい^素晴らしい|||U<CR><EXT>78<CR><LF>  
  
Bytes:  
02-32-50-7C-31-7C-36-36-36-7C-7C-7C-93-AA-82-A2-82-A2-5E-91-66-90-B0-82-E7-82-B5-82-A2-7C-7C-7C-55-0D-03-37-38-0D-0A

/// ASCII Byte constants  
public static class B  
{  
 public const byte ENQ = 5;  
 public const byte ACK = 6;  
 public const byte NAK = 21;  
 public const byte EOT = 4;  
 public const byte ETX = 3;  
 public const byte ETB = 23;  
 public const byte STX = 2;  
 public const byte CR = 13;  
 public const byte LF = 10;  
}

/// ASCII char constants (default host encoding)  
public static class C  
{  
 public const char ENQ = (char)B.ACK;  
 public const char ACK = (char)B.ACK;  
 public const char NAK = (char)B.NAK;  
 public const char EOT = (char)B.EOT;  
 public const char ETX = (char)B.ETX;  
 public const char ETB = (char)B.ETB;  
 public const char STX = (char)B.STX;  
 public const char CR = (char)B.CR;  
 public const char LF = (char)B.LF;  
}