

The degree sign is a non-ASCII character. It is encoded as follows:

MacRoman	0xA1
Windows-1252	0xB0
Shift JIS (Japanese)	0x81 0x8B
UTF-8	0xC2 0xB0

To simplify the discussion, we will ignore the issue of Japanese.

If you created the procedure file in Igor6 on Macintosh, it would work correctly in Igor6 on Macintosh but would print the wrong character in Igor6 on Windows. If you opened the procedure file in Igor7 or later with "Western" selected in Misc→Text Encoding→Default Text Encoding, it would work correctly on Macintosh but would print the wrong character on Windows. If you added this to the procedure file:

```
#pragma TextEncoding = "MacRoman"  
it would work correctly in Igor7 or later on both Macintosh and Windows.
```

Literal Strings in Igor6/Igor7 Procedures

If you want it to work in Igor6 and Igor7 or later, on Macintosh and Windows, and irrespective of the text encoding of the procedure file and the Misc→Text Encoding→Default Text Encoding setting, you will need to do this:

```
#if IgorVersion() >= 7.00  
    static StrConstant kDegreeSignCharacter = "\xC2\xB0"  
#else  
    #ifdef MACINTOSH  
        static StrConstant kDegreeSignCharacter = "\u241"\n    #else  
        static StrConstant kDegreeSignCharacter = "\u260"  
    #endif  
#endif  
  
Function PrintTemperature(temperature)  
    Variable temperature  
  
    // The character after %g is the degree sign  
    Printf "The temperature is %g%s\r", temperature, kDegreeSignCharacter  
End
```

This code uses escape sequences instead of literal characters to keep non-ASCII characters out of the procedure file. This allows it to work regardless of the text encoding of the procedure file in which it appears. For Igor7 or later, it uses the \xxx hexadecimal escape sequence. Igor6 does not support \xxx so it uses octal (\ooo) instead.

The section **Determining the Encodings of a Particular Character** on page III-482 shows how we determined the hex and octal codes.

The reason for making the StrConstants static is that other programmers may include the same code, causing a compile error if a user uses both sets of procedures.

The string constant kDegreeSignCharacter produces the correct string in Igor6 on Macintosh and Windows, and in Igor7 or later on Macintosh and Windows.

This will not print the right character for an Igor6 user using a Japanese font for the history area. Since you can not know the user's history area font, there is no way to cope with that.

If you have code that assumes that degree sign is one byte, you will need to modify it, since it is two bytes in UTF-8.