

Chapter III-16 — Text Encodings

If the degree sign is the only non-ASCII character that you need to represent in Igor6 and Igor7 or later, this problem is manageable. If not, you will need similar code for other characters. The additional complexity and kludginess will get out of hand. That's why we recommend against supporting Igor6 and Igor7 or later with the same set of procedures.

Literal Strings in Igor7-only Code

If your procedure file requires Igor7 or later, you can write the PrintTemperature function in the obvious way:

```
#pragma TextEncoding = "UTF-8"
#pragma IgorVersion = 7.00

Function PrintTemperature(temperature)
    Variable temperature

    // The character after %g is the degree sign
    Printf "The temperature is %g°\r", temperature
End
```

This will work in Igor7 or later on Macintosh and Windows, and without regard to the user's history font or the Misc→Text Encoding→Default Text Encoding setting.

Unlike the previous approaches, this technique allows you to use any Unicode character, not just the small subset available in MacRoman or Windows-1252.

Determining the Encodings of a Particular Character

This section shows a method for determining how a given character is encoded in MacRoman, Windows-1252, and UTF-8. The commands must be executed in Igor7 or later because they use the ConvertTextEncoding function which does not exist in Igor6.

```
// Print MacRoman code for degree sign as octal - Prints 241
Printf "%03o\r", char2num(ConvertTextEncoding("°", 1, 2, 1, 0)) & 0xFF

// Print Windows-1252 code for degree sign as octal - Prints 260
Printf "%03o\r", char2num(ConvertTextEncoding("°", 1, 3, 1, 0)) & 0xFF

// Print number of bytes of UTF-8 degree sign character - Prints 2
Printf "%d\r", strlen("°")

// Print first byte of UTF-8 for degree sign as hex - Prints C2
Printf "%02X\r", char2num("°"[0]) & 0xFF

// Print second byte of UTF-8 for degree sign as hex - Prints B0
Printf "%02X\r", char2num("°"[1]) & 0xFF
```

Using the printed information, we created the following Igor6/Igor7 code. It uses a hex escape sequence in Igor7 or later and an octal escape sequence in Igor6 because Igor6 does not support hex escape sequences.

```
#if IgorVersion() >= 7.00
    static StrConstant kDegreeSignCharacter = "\xC2\xB0"
#else
    #ifdef MACINTOSH
        static StrConstant kDegreeSignCharacter = "\241"
    #else
        static StrConstant kDegreeSignCharacter = "\260"
    #endif
#endif

Function PrintTemperature(temperature)
    Variable temperature
```