Parameters

wave is a wave reference to a numeric wave.

Details

Only the wave's data is present in the returned string. Other information, such as scaling, dimension labels, etc., is not included. You may need to use the Redimension operation to change the type of the wave or to perform an endian swap before you use WaveDataToString.

While Igor strings can contain embeded nulls, some parts of Igor are not prepared to handle them. For example printing a string with a null will only print the part of the string before the null. For more information, see **Embedded Nulls in Literal Strings** on page IV-16.

Example

```
Function WaveDataToStringDemo1()
    Make/FREE/B/U w1 = {49, 50, 51}

    // This requires Igor 9.00
    String s1 = WaveDataToString(w1)
    Print s1                // Prints 123

    // This approach works in older versions of Igor
    Variable np = numpnts(w1)
    String s2 = ""
    s2 = PadString(s2, np, 0)
    Variable n
    For (n=0; n < np; n++)
        s2[n,n] = num2char(w1[n])
    EndFor
    Print s2                // Prints 123
End

// Round trip using WaveDataToString and StringToUnsignedByteWave
Function WaveDataToStringDemo2()
    Make/FREE/D w2 = {1}
    String w2Str = WaveDataToString(w2)
    Print w2Str             // Prints nothing because w2Str contains leading null bytes
    Print strlen(w2Str)
    WAVE/B/U w2ByteWave = StringToUnsignedByteWave(w2Str)
    Print w2ByteWave        // Prints {0,0,0,0,0,0,240,63}

    // Redimension the byte wave to a double precision floating point wave
    Redimension/E=1/D/N=1 w2ByteWave
    Print w2ByteWave        // Prints {1}
End

// Generate mixed-case random letters
Function WaveDataToStringDemo3()
    Make/O/FREE/N=(1e3) letters
    MultiThread letters = trunc(abs(enoise(52)))

    // 0-25 uppercase, 26-51 become lowercase
    MultiThread letters = letters[p] < 26 ? letters[p] + 65 : letters[p] + 71
    Redimension/B/U letters

    // Create a string with all the letters.
    String lettersStr = WaveDataToString(letters)
    Print lettersStr[0,100]
End
```

See Also

StringToUnsignedByteWave, **wfprintf**, **Working With Binary String Data** on page IV-175

WaveDims

WaveDims(*wave*)

The WaveDims function returns the number of dimensions used by *wave*.

Returns zero if wave reference is null. See **WaveExists** for a discussion of null wave references.

Also returns zero if wave has zero rows. A matrix will return 2.