

Standard numeric format strings are used to format common numeric types. A standard format string takes the form *Axx* where *A* is a single alphabetic character called the format specifier, and *xx* is an optional integer called the precision specifier. The format specifier must be one of the built-in format characters. The precision specifier ranges from 0 to 99 and controls the number of significant digits or zeros to the right of a decimal. The format string cannot contain white spaces.

If the format string does not contain one of the standard format specifiers, then a `FormatException` is thrown. For example, the format string "z" is interpreted as a standard numeric format string because it contains one alphabetic character, but the alphabetic character is not one of the standard numeric format specifiers so a **FormatException** is thrown. Any numeric format string that does not fit the definition of a standard numeric format string is interpreted as a [custom numeric format string](#). The format string "c!" is interpreted as a custom format string because it contains two alphabetic characters, even though the character "c" is a standard numeric format specifier.

The following table describes the standard numeric format strings. Note that the result string produced by these format specifiers is influenced by the settings in the Regional Options control panel. Computers using different settings will generate different result strings.

Format specifier	Name	Description
C or c	Currency	The number is converted to a string that represents a currency amount. The conversion is controlled by the currency format information of the <code>NumberFormatInfo</code> object used to format the number. The precision specifier indicates the desired number of decimal places. If the precision specifier is omitted, the default currency precision given by the <b>NumberFormatInfo</b> is used.
D or d	Decimal	This format is supported for integral types only. The number is converted to a string of decimal digits (0-9), prefixed by a minus sign if the number is negative. The precision specifier indicates the minimum number of digits desired in the resulting string. If required, the number is padded with zeros to its left to produce the number of digits given by the precision specifier.
E or e	Scientific (exponential)	The number is converted to a string of the form "-d.ddd...E+ddd" or "-d.ddd...e+ddd", where each 'd' indicates a digit (0-9). The string starts with a minus sign if the number is negative. One digit always precedes the decimal point. The precision specifier indicates the desired number of digits after the decimal point. If the precision specifier is omitted, a default of six digits after the decimal point is used. The case of the format specifier indicates whether to prefix the exponent with an 'E' or an 'e'. The exponent always consists of a plus or minus sign and a minimum of three digits. The exponent is padded with zeros to meet this minimum, if required.
F or f	Fixed-point	The number is converted to a string of the form "-ddd.ddd..." where each 'd' indicates a digit (0-9). The string starts with a minus sign if the number is negative. The precision specifier indicates the desired number of decimal places. If the precision specifier is omitted, the default numeric precision given by the <b>NumberFormatInfo</b> is used.
G or g	General	The number is converted to the most compact of either fixed-point or scientific notation, depending on the type of the number and whether a precision specifier is present. If the precision specifier is

		<p>omitted or zero, the type of the number determines the default precision, as indicated by the following list.</p> <ul style="list-style-type: none"> <li>• <b>Byte</b> or <b>SByte</b>: 3</li> <li>• <b>Int16</b> or <b>UInt16</b>: 5</li> <li>• <b>Int32</b> or <b>UInt32</b>: 10</li> <li>• <b>Int64</b> or <b>UInt64</b>: 19</li> <li>• <b>Single</b>: 7</li> <li>• <b>Double</b>: 15</li> <li>• <b>Decimal</b>: 29</li> </ul> <p>Fixed-point notation is used if the exponent that would result from expressing the number in scientific notation is greater than -5 and less than the precision specifier; otherwise, scientific notation is used. The result contains a decimal point if required and trailing zeroes are omitted. If the precision specifier is present and the number of significant digits in the result exceeds the specified precision, then the excess trailing digits are removed by rounding. If scientific notation is used, the exponent in the result is prefixed with 'E' if the format specifier is 'G', or 'e' if the format specifier is 'g'.</p> <p>The exception to the preceding rule is if the number is a <b>Decimal</b> and the precision specifier is omitted. In that case, fixed-point notation is always used and trailing zeroes are preserved.</p>
N or n	Number	<p>The number is converted to a string of the form "-d,ddd,ddd.ddd...", where each 'd' indicates a digit (0-9). The string starts with a minus sign if the number is negative. Thousand separators are inserted between each group of three digits to the left of the decimal point. The precision specifier indicates the desired number of decimal places. If the precision specifier is omitted, the default numeric precision given by the <b>NumberFormatInfo</b> is used.</p>
P or p	Percent	<p>The number is converted to a string that represents a percent as defined by the <b>NumberFormatInfo.PercentNegativePattern</b> property or the <b>NumberFormatInfo.PercentPositivePattern</b> property. If the number is negative, the string produced is defined by the <b>PercentNegativePattern</b> and starts with a minus sign. The converted number is multiplied by 100 in order to be presented as a percentage. The precision specifier indicates the desired number of decimal places. If the precision specifier is omitted, the default numeric precision given by <b>NumberFormatInfo</b> is used.</p>
R or r	Round-trip	<p>The round-trip specifier guarantees that a numeric value converted to a string will be parsed back into the same numeric value. When a numeric value is formatted using this specifier, it is first tested using the general format, with 15 spaces of precision for a <b>Double</b> and 7 spaces of precision for a <b>Single</b>. If the value is successfully parsed back to the same numeric value, it is formatted using the general format specifier. However, if the value is not successfully parsed back to the same numeric value, then the value is formatted using 17 digits of precision for a <b>Double</b> and 9 digits of precision for a <b>Single</b>. Although a precision specifier can be appended to the round-trip format specifier, it is ignored. Round trips are given precedence over precision when using this specifier. This format is</p>

		supported by floating-point types only.
X or x	Hexadecimal	The number is converted to a string of hexadecimal digits. The case of the format specifier indicates whether to use uppercase or lowercase characters for the hexadecimal digits greater than 9. For example, use 'X' to produce "ABCDEF", and 'x' to produce "abcdef". The precision specifier indicates the minimum number of digits desired in the resulting string. If required, the number is padded with zeros to its left to produce the number of digits given by the precision specifier. This format is supported for integral types only.