

Number Format Strings

You can specify how number values are formatted for any measure or numeric-type dimension (`INT`, `INTEGER`, `LONG`, `FLOAT`, `DOUBLE`, or `DECIMAL` data types).

Built-In Numeric Formats

| Format Name | Description | Example Values |
|----------------|--|-------------------------------------|
| None | No format applied. Values will be sent to the client as is, and displayed according to the BI client's defaults. | N/A |
| General Number | Displays the number with no thousand separator. | 46193766.985470295 46193766.9855 |
| Fixed | Displays at least one digit to the left and two digits to the right of the decimal separator. | 46,193,766.99 |
| Standard | Displays the number with thousand separator, at least one digit to the left and two digits to the right of the decimal separator. | 46,193,766.99 |
| Percent | Displays the number multiplied by 100 with a percent sign (%) appended to the right. Always displays two digits to the right of the decimal separator. | 4619376698.55% |
| Scientific | Uses standard scientific notation. | 4.619E+7 |

Custom Numeric Formats

If you choose Custom as the format, you can supply a format string to specify how to format numeric values.

A customer format expression for numeric values contains one to four sections separated by semicolons.



Note: Some Business Intelligence tools support short-cut custom formats that expand to fully qualified formats. For example, Microsoft Excel will expand the custom format of `#,#` to use a thousands grouping size of `3`, resulting in `#,###`. Similarly Excel will expand `#,#.##` to `#,###.##`. AtScale does not perform automatic thousands-group expansion, therefore you must specify the exact custom format string.

| Number of Sections | Result |
|--------------------|---|
| One | The format expression applies to all values. |
| Two | The first section applies to positive values and zeros. The second section applies to negative values. |
| Three | The first section applies to positive values. The second section applies to negative values. The third section applies to zeros. |
| Four | The first section applies to positive values. The second section applies to negative values. The third section applies to zeros. The fourth section applies to NULL values. |

For example, if there are two sections, the first section defines the format for positive values and zeros, and the second section defines the format for negative values:

```
$#,##0;($#,##0)
```

If a section is empty (nothing between two semi-colons), the missing section prints using the format of the positive value. For example, the following format displays positive and negative values using the format in the first section and displays "Zero" if the value is zero:

```
$#,##0;;\Z\er\o
```

The following table identifies the characters that can appear in the format string for numeric formats.

none

Displays the number without any formatting.

0

Represents a digit placeholder that displays a digit or a zero (0).

If the number has a digit in the position where the zero appears in the format string, the formatted value displays the digit. Otherwise, the formatted value displays a zero in that position.

If the number has fewer digits than there are zeros (on either side of the decimal) in the format string, the formatted value displays leading or trailing zeros.

If the number has more digits to the right of the decimal separator than there are zeros to the right of the decimal separator in the format expression, the formatted value rounds the number to as many decimal places as there are zeros.

If the number has more digits to the left of the decimal separator than there are zeros to the left of the decimal separator in the format expression, the formatted value displays the additional digits without modification.

#

Represents a digit placeholder that displays a digit or nothing.

If the expression has a digit in the position where the number sign (#) appears in the format string, the formatted value displays the digit. Otherwise, the formatted value displays nothing in that position.

The number sign (#) placeholder works like the zero (0) digit placeholder except that leading and trailing zeros are not displayed if the number has the same or fewer digits than there are # characters on either side of the decimal separator in the format expression.

.

Represents a decimal placeholder that determines how many digits are displayed to the left and right of the decimal separator.

If the format expression contains only number sign (#) characters to the left of the period (.), numbers smaller than 1 start with a decimal separator. To display a leading zero displayed with fractional numbers, use zero (0) as the first digit placeholder to the left of the decimal separator.

The actual character used as a decimal placeholder in the formatted output depends on the number format recognized by the computer system.

Note: In some locales, a comma is used as the decimal separator.

%

Represents a percentage placeholder. The expression is multiplied by 100. The percent character (%) is inserted in the position where the percentage appears in the format string.

,

Represents a thousand separator that separates thousands from hundreds within a number that has four or more places to the left of the decimal separator.

Standard use of the thousand separator is specified if the format contains a thousand separator enclosed in digit placeholders (0 or #).

Two adjacent thousand separators, or a thousand separator immediately to the left of the decimal separator (whether or not a decimal is specified), means "scale the number by dividing the number by 1000, rounding as required." For example, you can use the format string "##0,," to represent 100 million as 100. Numbers smaller than 1 million are displayed as 0. Two adjacent thousand separators in any position other than immediately to the left of the decimal separator are treated as specifying the use of a thousand separator.

The actual character used as the thousand separator in the formatted output depends on the number format recognized by the computer system.

Note: In some locales, a period is used as the thousand separator.

:

Represents a time separator that separates hours, minutes, and seconds when time values are formatted.

Note: In some locales, other characters may be used as the time separator.

The actual character used as the time separator in formatted output is determined by the system settings on the computer.

/

Represents a date separator that separates the day, month, and year when date values are formatted.

The actual character used as the date separator in formatted output is determined by the system settings on the computer.

Note: In some locales, other characters may be used as the date separator.

E- E+ e- e+

Represents scientific format.

If the format expression contains at least one digit placeholder (0 or #) to the right of `E-`, `E+`, `e-`, or `e+`, the formatted value displays in scientific format and E or e is inserted between the number and the number's exponent. The number of digit placeholders to the right determines the number of digits in the exponent. Use E- or e- to include a minus sign next to negative exponents. Use E+ or e+ to include a minus sign next to negative exponents and a plus sign next to positive exponents.

- + \$ ()

Displays a literal character.

To display a character other than one of those listed, put a backslash () before the character or enclose the character in double quotation marks (" ").

\

Displays the next character in the format string.

To display a character that has special meaning as a literal character, put a backslash (\) before the character. The backslash itself is not displayed. Using a backslash is the same as enclosing the next character in double quotation marks. To display a backslash, use two backslashes (\). Examples of characters that cannot be displayed as literal characters include the following characters:

The date-formatting and time-formatting characters-a, c, d, h, m, n, p, q, s, t, w, y, /, and :

The numeric-formatting characters-#, 0, %, E, e, comma, and period

The string-formatting characters-@, &, <, >, and !

"ABC"

Displays the string inside the double quotation marks (" ").

To include a string in format from within code, use Chr(34) to enclose the text. (The character code for a double quotation mark is 34.)