# Field Codes Reference Guide

## Type conversion

6/10/2026

# Contents

- 
- Administrator

Learn about the type conversions you can use in field codes.

## Related documentation:

-

Field codes can use the type conversions detailed on this page.

# Bool

**Bool(*Arg*, [*Default*])**

Returns a Boolean converted from a number or a string.

## Arguments

| Name | Description |
|---|---|
| Arg | If a number, then converts 0 to False and nonzero to True.<br><br>If a string, then converts Off, No, and False to False, and On, Yes, and True to True. If another string, then returns Default. If Default is omitted, then returns False. |

## Examples

| Example | Result |
|---|---|
| | False |
| | True |
| | True |
| | False |
| | False |
| | True |

# Num

**Num (*String*,*[String]*)**

Returns a number converted from a string.

## Arguments

| Name | Description |
|---|---|
| First argument | The string to be converted. May be expressed in scientific notation. Returns 0 if the string is not recognizable as a number. Ignores nonnumeric characters following the number. |
| Second argument | Optional. The locale that must be used to parse the first argument. Some examples include: en_US for English (United States), en_GB for English (United Kingdom), and fr_FR for French (France). If omitted, the value of an internal configuration option is used, if present - contact your Genesys representative for details. Otherwise, the platform locale is used. |

## Examples

For clarity, the results shown in the table appear with three digits after the decimal point and always in the en_US format. Default number formatting shows no digits after the decimal point. Use the Text function (see Text) or format operator (%) to override the default formatting.

| Example | Result |
|---|---|
| | 10.000 |
| | 10.000<br>(Assuming the locale is en_US.) |
| | 10.000<br>(Note the comma-decimal separator in the first argument.) |
| | 0.120 |
| | 1220.000<br>(Assuming the locale is en_US.) |
| | 1220.000<br>(Note the comma-decimal separator in the first argument.) |
| | 0.000 |

## Text

**Text (*Arg[,Pattern[,String]]*)** or **Text (*Arg:Pattern*)**

Returns a string converted from an argument of any data type. Use the format operator (:) as shorthand for this function.

### Arguments

| Name | Description |
|---|---|
| Arg | The value to be converted |
| Pattern | Optional. The picture string to use for formatting. If omitted, default formatting is used. The syntax of the picture string depends on the data type. See Number Formatting (Arg is a Number). |
| String | Optional. The locale that must be used to parse the first argument. Some examples include: en_US for English (United States), en_GB for English (United Kingdom), and fr_FR for French (France). If omitted, the value of an internal configuration option is used, if present - contact your Genesys representative for details. Otherwise, the platform locale is used. |

## Number Formatting (Arg is a Number)

If Arg is a number, then the regular expression syntax of the optional pattern string is as follows:

`#*.?#*`

Where:

The pound sign (#) represents a digit. Any number of #s, including 0 may appear before the decimal character. Specify the minimum number of digits that should appear to the left of the decimal. If the integer part of the formatted number contains fewer than the specified number of digits, the number is padded with leading zeros.

Any number of #s, including 0, may appear after the decimal character. Specify the precision of the fractional part of the number. The number is rounded to the specified precision.

Only the decimal separator in the result is locale dependent (there is no grouping separator).

The Examples of Number Formatting table contains some examples.

Examples of Number Formatting

| Pattern | Arg Value | Locale | Result |
|---|---|---|---|
| "" | 0 | en_US | "0" |
| "" | 123.456 | en_US | "123" |
| "#" | 0 | en_US | "0" |
| "##" | 0 | en_US | "00" |
| "##" | 123.456 | en_US | "123" |
| "#." | 0 | en_US | "0. " |
| "#." | 123.456 | en_US | "123. " |
| ".##" | 0 | en_US | ".00" |
| ".##" | 0.456 | en_US | ".46" |
| ".##" | 123.456 | en_US | "123.46" |
| ".##" | 20000.456 | en_US | "20000.46"<br>(Note the decimal point separator in the result.) |
| ".##" | 123.456 | fr_FR | "123,46"<br>(Note the comma-decimal separator in the result.) |
| ".##" | 20000.456 | fr_FR | "20000,46"<br>(Note the comma-decimal separator in the result.) |

## Duration Formatting (Arg is a Number)

If Arg is a number, then the regular expression syntax of the optional pattern string is as follows:

`()?#*`

Where:

represents a duration and can be any of the sequences in the following list. Upper- or lowercase letters are accepted.

- HH
- HH:MM
- HH:MM:SS
- MM
- MM:SS
- SS
- H
- H:MM
- H:MM:SS
- M
- M:SS
- S

may be followed by a .## string, which specifies the precision of the last element of the duration. Any C or % suffixes are ignored. When you format a value as a duration, the value is always assumed to be expressed in days.

The pound sign (#) represents a digit. Any number of #s, including 0, may appear before the decimal character and specify the minimum number of digits that should appear to the left of the decimal. If the integer part of the formatted number contains fewer than the specified number of digits, the number is padded with leading zeroes.

Any number of #s, including 0, may appear after the decimal character and specify the precision of the fractional part of the number. The number is rounded to the specified precision.

The Examples of Duration Formatting table contains some examples.

Examples of Duration Formatting

| Pattern | Arg Value | Locale | Result |
|---|---|---|---|
| "HH" | 10.5083 | en_US | "11" |
| "HH.## " | 10.5083 | en_US | "10.51" |
| "HH:MM" | 10.5083 | en_US | "10:30" |
| "HH:MM.# " | 10.5083 | en_US | "10:30.5" |

| Pattern | Arg Value | Locale | Result |
|---|---|---|---|
| "HH:MM:SS" | 10.5083 | en_US | "10:30:30" |
| "MM" | 10.5083 | en_US | "630" |
| "MM.## " | 10.5083 | en_US | "630.50" |
| "MM:SS" | 10.5083 | en_US | "630:30" |
| "SS" | 10.5083 | en_US | "37830" |

## Currency Formatting (Arg is a Number)

If Arg is a number, then the regular expression syntax of the optional parameter string is as follows:

`#*.?#*[Cc]`

Where:

A C or a c means format as currency. The grouping separator, the decimal separator, and the currency sign in the result are locale dependent.

The Examples of Currency Formatting table contains some examples.

Examples of Currency Formatting

| Pattern | Arg Value | Locale | Result |
|---|---|---|---|
| "C" | 12.34 | en_US | "$12.34" |
| "C" | -12.34 | en_US | "($12.34)" |
| "#.#C" | 12.34 | en_US | "$12.3" |
| "#.#C" | -12.34 | en_US | "($12.3)" |
| "C" | 12.34 | en_GB | "£12.34 " |
| "C" | -12.34 | en_GB | "-£12.34 " |

| Pattern | Arg Value | Locale | Result |
|---|---|---|---|
| "#.#C" | 12.34 | en_GB | "£12.3 " |
| "#.#C" | -12.34 | en_GB | "-£12.3 " |
| ".##C" | 20000.456 | en_US | "$20,000.46"<br>(Note the comma grouping separator and point decimal separator in the result.) |
| ".##C" | 20000.456 | fr_FR | "20 000,46 €" (Note the decimal comma separator in the result.) |

## Percentage Formatting (Arg is a Number)

If Arg is a number, then the regular expression syntax of the optional pattern string is as follows:

`#*.?#*%`

Where:

The percent sign (%) means multiply by 100 and append the locale-dependent sign for percent values. If the % appears by itself, the formatter rounds to the nearest integral value and omits a decimal point (equivalent to the format #%).

The grouping separator, the decimal separator, and the percent sign in the result are locale dependent.

The Examples of Percentage Formatting table contains some examples.

Examples of Percentage Formatting

| Pattern | Arg Value | Locale | Result |
|---|---|---|---|
| "%" | 0 | en_US | "0%" |
| "%" | 0.123456 | en_US | "12%" |
| "#.##%" | 0.123456 | en_US | "12.35%" |
| "#.##%" | 0.123456 | fr_FR | "12,35%"<br>(Note the comma-decimal |

| Pattern | Arg Value | Locale | Result |
|---|---|---|---|
| | | | separator in the result.) |

## Date/Time Formatting

Use elements shown in the Date/Time Pattern Letters table to construct a Date/Time pattern string. The letters must be in uppercase or lowercase, as shown in the table (for example, MM not mm). Characters that are not picture elements, or that are enclosed in single quotation marks, will appear in the same location and unchanged in the output string.

### Date/Time Pattern Letters

| Element | Meaning |
|---|---|
| d | Day of month as digits, with no leading zero for single-digit days |
| dd | Day of month as digits, with leading zero for single-digit days |
| ddd | Day of week as a three-letter abbreviation |
| dddd | Day of week as its full name |
| M | Month as digits, with no leading zero for single-digit months |
| MM | Month as digits, with leading zero for single-digit months |
| MMM | Month as a three-letter abbreviation |
| MMMM | Month as its full name |
| y | Year as last two digits, but with no leading zero for years less than 10 |
| yy | Year as last two digits, but with leading zero for years less than 10 |
| yyyy | Year represented by full four digits |

| Element | Meaning |
|---|---|
| h | Hours, with no leading zero for single-digit hours; 12-hour clock |
| hh | Hours, with leading zero for single-digit hours; 12-hour clock |
| H | Hours, with no leading zero for single-digit hours; 24-hour clock |
| HH | Hours, with leading zero for single-digit hours; 24-hour clock |
| m | Minutes, with no leading zero for single-digit minutes |
| mm | Minutes, with leading zero for single-digit minutes |
| s | Seconds, with no leading zero for single-digit seconds |
| ss | Seconds, with leading zero for single-digit seconds |
| tt | Time-marker string, such as AM or PM |

The examples in the Examples of Date/Time Formatting table assume that the date being formatted is August 6, 2003, @ 15:05:10:

Examples of Date/Time Formatting

| Pattern | Locale | Result |
|---|---|---|
| "MMMM d, yyyy @ hh:mm:ss tt" | en_US | "August 6, 2003 @ 03:05:10 PM" |
| "MMMM dd, yyyy @ HH:mm:ss" | en_US | "August 06, 2003 @ 15:05:10" |
| "dd MMMM yyyy HH:mm:ss" | fr_FR | "06 aout 2003 15:05:10" |
| "MMM d, yy @ h:mm:ss tt" | en_US | "Aug 6, 03 @ 3:05:10 PM" |
| "M/dd/yy" | en_US | "8/06/03" |

## Boolean Formatting

A Boolean picture string is simply two words separated by a comma. The first word is used if the Boolean value is True, and the second is used otherwise.

The Examples of Boolean Formatting table shows some examples:

Examples of Boolean Formatting

| Field Code | Result |
|---|---|

| | |
|---|---|
| | "Yup" |
| | "No" |
| | "peach" |

## String Formatting

Picture strings do not apply to string values. Strings are always output unchanged. If you want to output a piece of a string, or change the case, then you can use one of the string-manipulation functions previously described.