Popdock Guide for a Calculated Field

Here is a list of example operators and methods that can be used in a calculated field. This includes a definition of what the operator or method does and a few syntax examples for them.

Arithmetic Calculations

These operators are mostly used when the field types are numeric values and the assigned type for the calculated field is set to number.

+ (ADD) Adds two numeric values.

Examples:

- {Field1} + {Field2}
- {Field1} + 5

- (SUBTRACT) Subtracts one numeric value from another numeric value.

Examples:

- {Field1} - {Field2}
- {Field1} - 5

*** (MULTIPLY)** Multiplies two numeric values together.

Examples:

- {Field1} * {Field2}
- {Field1} * 5

/ (DIVIDE) Divides one numeric value by another numeric value.

Examples:

- {Field1} / {Field2}
- {Field1} / 5

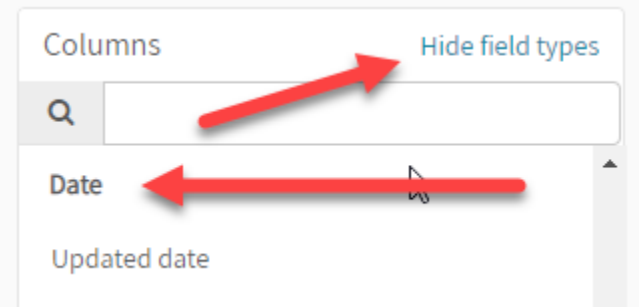
% (REMAINDER) If the values are integers, this returns the remainder of dividing one numeric value by another numeric value.

Examples:

- {Field1} % {Field2}
- {Field1} % 5

Date Type Calculations and Conversions

These operators are mostly used when the field type is a date. The calculated field type will be based on the type of calculation or conversion you are performing.



Date Properties:

DateTime.Now

This expression will provide the current date.

Example: DateTime.Now
Calculated Field Type: Date

Created date	Calculated Field Test
10/7/2016	9/18/2018

{Field1}.Day

Gets the day of the month.

Example: {Create date}.Day
Calculated Field Type: Number

Created date	Calculated Field Test
10/7/2016	7

{Field1}.DayOfWeek

Gets the day of the week.

Example: {Create date}.DayOfWeek
Calculated Field Type: String

Created date	Calculated Field Test
10/7/2016	Friday

{Field1}.DayOfYear

Gets the day of the year.

Example: {Create date}.DayOfYear

Calculated Field Type: Number

Created date	Calculated Field Test
10/7/2016	281

{Field1}.Month

Gets the month of the date.

Example: {Create date}.Month

Calculated Field Type: Number, String

Created date	Calculated Field Test
10/7/2016	10

{Field1}.Year

Gets the year of the date.

Example: {Create date}.Year

Calculated Field Type: Number, String

Created date	Calculated Field Test
10/7/2016	2016

DateTime.DaysInMonth({Field1}.Year,{Field1}.Month)

Gets the number of days for the Month of that Year.

Example: DateTime.DaysInMonth({Created date}.Year,{Created date}.Month)

Calculated Field Type: Number, String

Created date	Calculated Field Test
10/7/2016	31

Date Methods:

{Field1}.AddDays(*x*)

Adds *x* days to the date.

Example: {Create date}.AddDays(5)

Calculated Field Type: Date

Created date	Calculated Field Test
10/7/2016	10/12/2016

{Field1}.AddMonths(*x*)

Adds *x* months to the date.

Example: {Create date}. AddMonths(5)

Calculated Field Type: Date

Created date	Calculated Field Test
10/7/2016	3/7/2017

{Field1}.AddYears(*x*)

Adds *x* years to the date.

Example: {Create date}. AddYears(5)

Calculated Field Type: Date

Created date	Calculated Field Test
10/7/2016	10/7/2021

DateTime.Compare({Field1}, {Field2})

Compares two dates and returns an integer that indicates whether the first date is earlier than (-1), the same as(0), or later than(1) the second instance.

Example: Compare({Create date}, {Updated date})

Calculated Field Type: String

Created date	Update date	Calculated Field Test
7/19/2016	12/27/2017	-1
11/2/2016	11/2/2016	0

Date Format Specifier:

({Field1}).ToString("x")

This will convert a date field to a string in the specified format *x*. Here is a list of some of the formats available. Use one of the following letters in quotes to replace the *x* to see the format displayed:

```
d: 6/15/2008
D: Sunday, June 15, 2008
f: Sunday, June 15, 2008 9:15 PM
F: Sunday, June 15, 2008 9:15:07 PM
g: 6/15/2008 9:15 PM
G: 6/15/2008 9:15:07 PM
m: June 15
o: 2008-06-15T21:15:07.0000000
R: Sun, 15 Jun 2008 21:15:07 GMT
s: 2008-06-15T21:15:07
t: 9:15 PM
T: 9:15:07 PM
u: 2008-06-15 21:15:07Z
U: Monday, June 16, 2008 4:15:07 AM
y: June, 2008
```

You can also customize how you want to display your date string.

Example: ({Created date}).ToString("yyyy-MM-dd")
Calculated Field Type: String

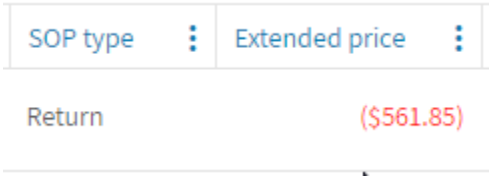
Created date	Calculated Field Test
7/25/2017	2017-07-25

If-Else Calculated Field

{Field1} == "Return" ? {Field2} * -1 : {Field2}

This is a more complicated calculation where it uses an if-else statement to accomplish a calculation. The way the formula is constructed is comparing a field and whether the compare is true or false to complete a specific calculation. In the above example if {Field1} is equal to "Return" then take {Field2} and multiply it by a -1 and that will be the data it shows for the calculated field. Now if {Field1} does not equal "Return" it performs the calculation that is after the colon which in the above example is to just keep the same value as {Field2}.

Example: {SOP type} == "Return" ? {Extended price} * -1 : {Extended price}
Calculated Field Type: Will vary depending on the fields in the calculation.



SOP type	Extended price
Return	($561.85)