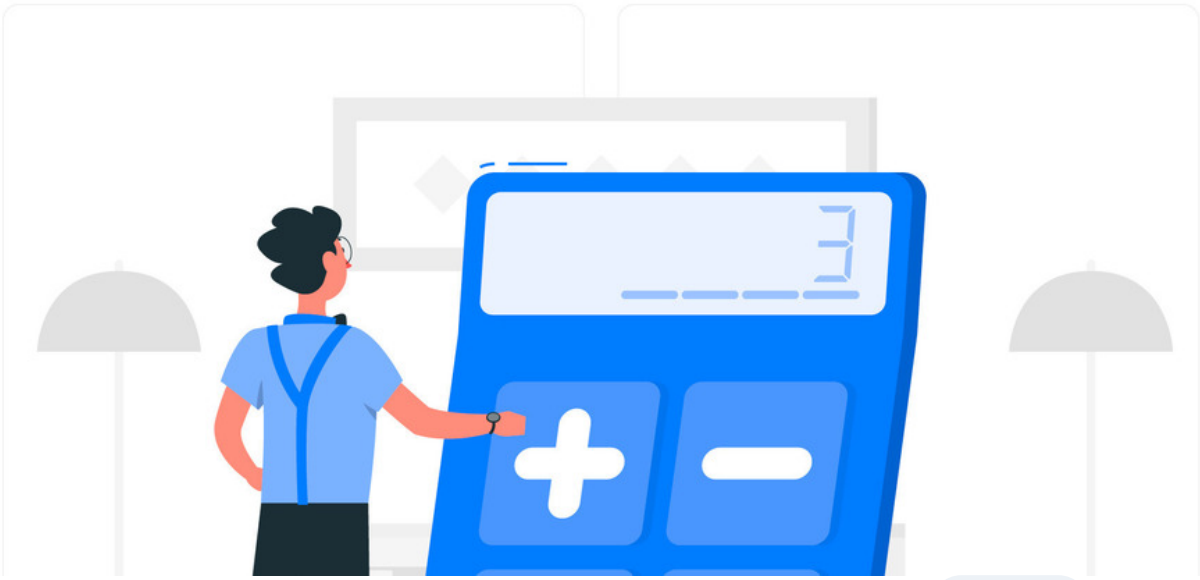
Link to article: https://discover.claromentis.com/knowledgebase/articles/59/expressions-calculating-fields-in-infocapture



Expressions & Calculating Fields in Infocapture Endorsed

If you have a query about expressions, please raise asupport ticket for the team to assist your further :)

Introduction

Expressions can be used within Infocapture, to produce calculated values. These expressions should be defined as the default value of the field which should store and display the calculated value. Such values will be unchangeable and therefore it's recommended that these fields be configured as disabled, to avoid any confusion.

Syntax

Expressions must begin with...

```
{#
```

...and end with...

```
}
```

Variables

Symbolic names of fields act as the variables and must be prefixed by $.

For example, a field with the symbolic name 'location' would be:

```
$location
```

The output value of a select field (including single select and radio buttons) will be its numerical ID.

For example, you may have a select field labelled Location, with the following values:

- Brighton
- London
- New York
- Miami

Each value is assigned a numerical ID:

- Brighton,#1
- London,#2
- New York,#3
- Miami,#4

If this select field is defined within an expression using $location the output value would be its numerical ID.

For example, if London was selected, the output value would be 2.

If the string value is required, replace $ with @:

```
@location
```

If London was selected, the output value would then be London instead of 2.

If a checkbox field is defined within an expression, the output value would either be 0 (not ticked) or 1 (ticked).

Functions

Calculations

By default, all values have text format. To convert these values to a numeric format, functions must be used.

To calculate a floating point, use the following function:

```
parseFloat()
```

To calculate an integer, use the following function:

```
parseInt()
```

Concatenation

To concatenate fields, use the following function:

```
concat()
```

Examples

Copying fields

The expression below copies the numerical ID of the option selected in the Expense Type field to the Expense Code field. This expression has been defined as the default value of the Expense Code field:

```
{#$expense_type}
```

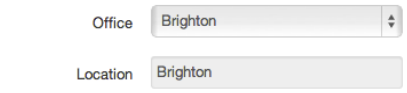
This is a good example of where numerical IDs of field values can come in handy.

By default, they are auto-generated chronologically, but in this instance, they have been changed to the relevant expense codes. The values of the Expense Type field are:

- Accomodation,#4500
- Business Mileage,#4505
- Client Entertainment,#4510
- Employee Entertainment,#4515
- Gift,#4520
- Miscellaneous,#4525
- Stationery,#4530
- Subscriptions,#4535
- Subsistence,#4540
- Taxis/Parking/Toll charges,#4545
- Telephone/Broadband/Mobile,#4550

The expression below copies the string value of the Office field to the Location field. This expression has been defined as the default value of the Location field:

```
{#@office}
```



Notice that @ has been used to prefix the symbolic name of the field instead of $. This is because the string value is required, rather than the numerical ID.

Calculations

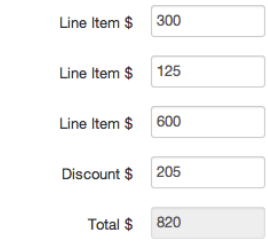
Multiplication

The expression below calculates the value of the Cost field multiplied by the Quantity field. This expression has been defined as the default value of the Total field:

```
{#parseFloat($cost)*parseFloat($quantity)}
```

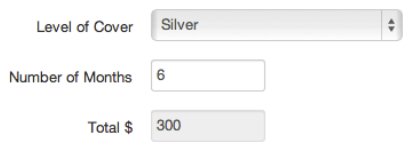
The expression below calculates the value of the 3 Line Item fields and then deducts the value of the Discount field. This expression has been defined as the default value of the Total field:

```
{#parseFloat($line_item_1)+parseFloat($line_item_2)+parseFloat($line_item_3)-parseFloat($discount)}
```



The expression below calculates the value of the numerical ID of the option selected in the Level of Cover field, multiplied by the value of the Number of Months field. This expression has been defined as the default value of the Total field:

```
{#parseInt($option)*parseInt($number_months)}
```



Average

The expression below calculates the average of `3` fields `field1`, `field2`, and `field` 3

```
{# (parseInt($field1) + parseInt($field2) + parseInt($field3) ) / 3 }
```

Your browser doesn't support video.
Please download the file. video/mp4

Decimal places

It is possible to limit a numeric value to two decimal places with the following:

```
{#parseInt( ("your expression here") * 100) / 100}
```

Or, if you are using Claromentis 8, you will have a "Financial" field type that will round to two decimal places for you, without needing to use this formula.

Show a percentage

```
{#concat((parseFloat(($NameOfFieldOne/$NameOfFieldTwo)*100)),'%')}
```

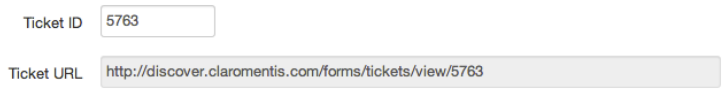
Concatenation

The expression below concatenates the values of First Name and Last Name to output Full Name, with a space in between. This expression has been defined as the default value of the Full Name field:

```
{#concat($first_name,' ',$last_name)}
```

The expression below concatenates a pre-defined URL with the value of the Ticket ID field. This expression has been defined as the default value of the Ticket URL field:

```
{#concat('http://discover.claromentis.com/forms/tickets/view/',(parseFloat($ticket_id)))}
```



Created on 8 November 2013 by Hannah Door. Last modified on 3 August 2023
Tags: expression, calculation, infocapture, bpm, sums, equations