Document Template Syntax Reference

This reference is designed to assist you when creating document templates, formerly called print templates. The syntax required for inserting field variables, formulas, conditions, and other elements are listed below.

> If your syntax includes quotation marks, make sure you're using plain quotes rather than smart quotes. To disable automatic smart quote formatting, go to File > Options, open the Proofing section, click AutoCorrect Options, go to the AutoFormat tab, and clear the "Straight quotes" with "smart quotes" checkbox.

Field Variables

Field variables can be inserted into document templates so that the field's value is displayed in the document when the document template runs.

The general form for field variables is `$FieldLabel`. For example, in the People table the variable for the Login field is `$login`.

You can construct chains of field variables to find values in other tables linked to the current record. These are in the form `$Field1.Field2` where Field2 references a field in another table linked through Field1. Field variable chains must be constructed manually by locating the appropriate field names in each table.

Syntax

```
$formula($field)
$formula($field1.field2)
```

Examples

`$formula($login)`	Insert the Login value.
`$formula($account_rep.backup_person)`	Insert the name of the Account Rep's backup person.
`$formula($related123123123)`	Insert a related table from the current record.

Note that if you are putting dollar signs ($) into a document template to format values in a number field, you can enter a backslash (\) to ensure that the dollar sign does not get eliminated by the system while it is processing the variables and formulas in the document template. For example, \$25,000 will appear as $25,000. You can configure the system so that it does not eliminate the backslash by changing the value of the Keep slash prior escaped dollar sign global variable to Yes.

View or Download Field Variables

A list of field variables for the current table can be displayed when creating or editing a document template.

- Click Show Field Variables to open the Formula Help wizard with a list of the Fields, Global Variables, and Functions.
- Click Download Template to save an MS Word file which lists the field name and field label for each field in the current table.

Formulas and Conditions

Formulas and conditions are used in document templates to calculate totals, insert different fields based on the values, and insert or delete text when a condition is met. A longer list of available formulas can be found in the Formula Help window, available from the Mass Edit Wizard and other system locations.

The following formulas and examples represent the most common uses and will help you create complex document templates.

Basic Formulas

Standard mathematical operations can be used with numeric data types such as integer, floating point, and calculated result fields.

Syntax

```
$formula($equation)
```

Examples

`$formula ($contract_amount*$discount_percentage)`	Multiply the *Contract Amount* by the *Discount Percentage*
`$formula($total_cost - $refund_amount)`	Subtract the *Refund Amount* from the *Total Cost*

Concatenate Strings

concatenatestringsexcerpt

The concatenate strings command allows you to combine field values with text strings, field variables, or other formulas that use variables. For example, if you use `$formula(concat("x",$field_name,"z",...))`, the "x" and "z" placeholders are where you insert text strings, field variables, or formulas.

It can also be used when inserting a Choice field value, if the value is rendered as code such as 1@2 rather than the text of the value. To do so, enter the value as the only parameter.

Syntax

```
$formula(concat("x",$field,"z",...))
```

The variables may be text strings, field variables, or other formulas and variables. Text strings must be surrounded in double quotation marks (""), while variables and formulas do not.

If the first piece is a text string, you can use shorthand to combine variables and strings, but $formula() is still required in email templates:

```
$formula("x"+$variable+"z"+...)
```

Examples

Example formula	Output
`$formula(concat("Your account representative is ",$account_rep,"."))`	Your account representative is Hector Gomez.
`$formula(concat($company_name," Support Contract"))`	Agiloft Support Contract.
`$formula(""+$field1*$field2+" Total")`	2,142 Total
`$formula(concat($contract_start_date," to ",$contract_end_date))`	09/01/19 to 08/31/2020

Dateformat

dateformatexcerpt

`Dateformat()` is used to display a date/time field in a particular format, often for localization. `Dateformat()` takes two arguments: the desired pattern format, and the field variable. Date time patterns are indicated with a series of letters that represent elements such as month, day in month, day of week, year, hour, and minute. For a full list of possible formats see this page from Java about SimpleDateFormat().

Syntax

```
dateformat("output pattern","$field")
```

Date/time Pattern	Example output
"MM/dd/yy"	05/31/16
"yyyy/dd/MM"	2016/22/09
"MMMMM"	July (name of month)
"d"	10 (day of month)

Examples

The following results are expected when `$contract_start_date` evaluates to February 10, 2016 at 01:00:00.

Formula	Sample output
$formula(dateformat ("d",$contract_start_date))	10. *Inserts which day of the month the contract starts.*
$formula(dateformat ("MMMMM",$contract_start_date))	February. *Inserts the full text name of the month.*
$formula(dateformat ("yyyy",$contract_start_date))	2016. *Inserts the year of the contract start date.*
$formula(dateformat("MM/dd/yyyy hh:mm", NOW()))	*Inserts the current date and timestamp when the document template runs.*

Look through the Sample Output column for the format you want to use, then use the corresponding Formula entry to use this format in your document templates. These examples use `$contract_start_date` with a value of February 10, 2016, so if you're referring to a different Date field, replace `$contract_start_date` with the logical name of the new field.

Sample Output	Formula
2/10/16	$formula(dateformat("M/d/yy",$contract_start_date))
2/10/2016	$formula(dateformat("M/d/yyyy",$contract_start_date))

10/2/16	$formula(dateformat("d/M/yy",$contract_start_date))
10/2/2016	$formula(dateformat("d/M/yyyy",$contract_start_date))
February 10, 2016	$formula(dateformat ("MMMMM d, yyyy",$contract_start_date))
February 10th, 2016	$formula(dateformat("MMMM",$contract_start_date)+" "+dateformat("d",$contract_start_date)+((dateformat("dd",$contract_start_date))=="11"||(dateformat("dd",$contract_start_date))=="12"\|\|(dateformat("dd",$contract_start_date))=="13"?"th":(((replace(dateformat("dd",$contract_start_date),".*(.)","$1"))=="1"?"st":((replace(dateformat("dd",$contract_start_date),".*(.)","$1"))=="2"?"nd":((replace(dateformat("dd",$contract_start_date),".*(.)","$1"))=="3"?"rd":"th")))))+", "+dateformat("yyyy",$contract_start_date))
February 10th, 2016 (alternate formula)	$formula(dateformat("MMMM",$contract_start_date)+" "+(dateformat("d",$contract_start_date))+((dateformat("dd",$contract_start_date))=="01"||(dateformat("dd",$contract_start_date))=="21"\|\|(dateformat("dd",$contract_start_date))=="31" ?"st":((dateformat("dd",$contract_start_date))=="02"\|\|(dateformat("dd",$contract_start_date))=="22"?"nd":((dateformat("dd",$contract_start_date))=="03"\|\|(dateformat("dd",$contract_start_date))=="23" ? "rd" : "th")))+", "+dateformat("yyyy",$contract_start_date))
February 10th, 2016	$formula(dateformat("MMMM",$contract_start_date)) $formula(dateformat("d",$contract_start_date))$formula(((dateformat("dd",$contract_start_date))=="11"||(dateformat("dd",$contract_start_date))=="12"\|\|(dateformat("dd",$contract_start_date))=="13"?"th":(((replace(dateformat("dd",$contract_start_date),".*(.)","$1"))=="1"?"st":((replace(dateformat("dd",$contract_start_date),".*(.)","$1"))=="2"?"nd":((replace(dateformat("dd",$contract_start_date),".*(.)","$1"))=="3"?"rd":"th")))))), $formula(dateformat("yyyy",$contract_start_date))
10th day of February, 2016	$formula(dateformat("d",$contract_start_date)+((dateformat("dd",$contract_start_date))=="11"\|\|(dateformat("dd",$contract_start_date))=="12"||(dateformat("dd",$contract_start_date))=="13"?"th":(((replace(dateformat("dd",$contract_start_date),".*(.)","$1"))=="1"?"st":((replace(dateformat("dd",$contract_start_date),".*(.)","$1"))=="2"?"nd":((replace(dateformat("dd",$contract_start_date),".*(.)","$1"))=="3"?"rd":"th")))))+" day of "+dateformat("MMMM",$contract_start_date)), $formula(dateformat("yyyy",$contract_start_date))

num2words

num2wordsexcerpt

The num2words formula converts numerals into words. For example, you can use the function to render "12" as "twelve," "twelfth," or "12th," or you can format it as currency. If you want to output words in another language, you can specify an alternate language code as the last parameter. The currency option requires a language to be specified in order to identify the appropriate currency.

The format options are cardinal; ordinal; ordinal_num, which renders the ordinal format with numerals; and currency, which requires a language code.

The first input parameter must be a numerical value. For ordinal formatting, the number must be an integer. The default format is cardinal, and the default language is U.S. English.

Syntax

```
num2words(value, "format", "language")
```

Examples

`num2words(12.76, "cardinal")`	twelve point seven six
`num2words(39, "ordinal")`	thirty-ninth
`num2words(39, "ordinal_num")`	39th
`num2words(39, "currency", "es")`	treinta y nueve euros
`num2words(39, "currency", "en")`	thirty-nine U.S. dollars

Document Template Examples

In a document template, use the $formula wrapper. These examples render a contract value of $1,000.20, with the second example capitalizing the first word.

$formula(num2words($contract_amount, "currency", "en"))	one thousand U.S. dollars and twenty cents
$formula(upper(replace(num2words($contract_amount, "currency", "en"), "(^[a-z]).*", "$1")) + replace(num2words($contract_amount, "currency", "en"), "^[a-z](.*)", "$1"))	One thousand U.S. dollars and twenty cents

Format Formula Outputs

formatexcerpt

`format()` is used to determine the output format of a formula based on the display characteristics of a chosen field. It is most commonly used to set the appropriate display for the output of a formula that includes Currency fields.

Syntax

```
$formula(format("table_name.field",$value))
```

In the syntax above, table_name.field is the reference to the field whose formatting you want to use. For instance, contract.dollar_amount could indicate using the $1.00 standard US currency display format. The formatting field can be referenced from any table. The $value may be any valid formula such as a field or formula including multiple field variables.

Examples

The following example formulas will help you create your own formulas.

Formula	Output
`$formula(format("contract.display_field",$contract_amount))`	Inserts the contract amount field formatted with the display characteristics of Display_Field.
`$formula(format("quote.base_currency",$total_amount*$discount_percentage))`	Multiplies the Total Amount times the Discount Percentage and formats the output based on the Base Currency field's settings.
`$formula(format("pricing_plan.monthly_price", (($number_of_users*16) + ($number_of_units*5))))`	Formats this formula using the Monthly Price field in the Pricing Plan table: multiply the number of users times 16 and add five times the number of units.

Conditional Text, Paragraphs, or Clauses

A common requirement is whether to show words, multiple lines, or paragraphs of text based on the values in a record or a condition based on those values. Conditions can use a full range of logical operators such as == (equivalency), != (not equal), >, <, >=, <=.

Ternary Operator

ternaryoperatorsexcerpt

Short conditional statements can be inserted with the ternary operator "a ? b : c." This works like an if-else statement: if the condition "a" evaluates true, then insert "b;" if "a" is false, insert "c." You must provide all parameters for the ternary operator to function, even if one parameter simply inserts a blank space. If you want to output a string in the b or c parameters, include quotation marks around the string.

The ternary operator is useful in formulas that divide by a variable, where that variable might sometimes equal zero. Otherwise, the attempt to divide by zero results in an error. Use the ternary operator to check whether that variable is zero and then provide formulas for each situation. For example, if you want to calculate a percentage using two fields, but $field2 might sometimes equal zero, you could use "$field2 == 0 ? 0 : 100*($field1/$field2)" to output zero if $field2 is zero and otherwise output the calculated percentage.

In some cases, the ternary operator might unintentionally render a Choice field value as code, such as 1@2, instead of the text of the value. To show the text instead, use the concat() function inside the ternary syntax. For example: $formula($choice_field=="Choice A" ? concat($preselected_value_in_choice_field2) : concat($ preselected_value_in_choice_field3))

Syntax

```
$formula($condition ? "True Output" : "False output")
```

Example

In this example the statement compares the variable $n to the value 1. If it is 1, "$n is 1" is shown. If it's not 1, "$n is not 1" is shown.

```
($n==1) ? "$n is 1" : "$n is not 1"
```

In this example, the statement uses functions inside the ternary formula. This outputs information about the delegation timeline, if the delegation is time-based, and otherwise states that the delegation isn't time-based. Notice that inside the ternary function, $formula is not used a second time.

```
$formula(($delegation_type=="Time-Based") ? "Delegation from " + dateformat("MMM
dd, yyyy", $delegation_start_date) + " to " dateformat("MMM dd, yyyy",
$delegation_end_date) : "Delegation is not time-based.")
```

Additional Examples

Formula	Explanation / Output
$formula($authorized_contract=="Yes" ? "'Authorized contractor' means a third party contracted to develop or assist with the development of an Authorized Application." :	Inserts the sentence "'Authorized contractor' means a third party contracted to develop or assist with the development of an Authorized Application." if the *Authorized Contract* field value is Yes; otherwise, a blank space is inserted.

"")	
$formula((isEmpty($company_id.fax)) ? "" : "Fax: " + $company_id.fax)	If the company's linked Fax field is empty, insert nothing. If the Fax field is not empty, insert "Fax: 555-111-2134."
$formula($country=="USA"&&$city==" Redwood" ? "The company is located in Redwood, USA" : "")	Inserts the sentence "The company is located in Redwood, USA" if the *Country* is USA and the *City* is Redwood; otherwise, a blank space is inserted.

If-Then-Else Conditions to Display Field Variables

The `$if` statement evaluates a condition to decide which field variable to display. It is similar to the ternary operator, but the true and false outputs may only be field variables or the $merge command with an attached file field.

Syntax

```
$if(condition) ? $fieldTrue : $fieldFalse
```

The second (else) field variable may be left out to display blank if the condition is not met.

```
$if(condition) ? $fieldTrue
```

Or, the `$merge($filefield)` command may be used as an output:

```
$if(condition) ? $merge($filefield) : $fieldFalse
```

Examples

$if ($discount_applied == "Yes") ? $percentage_discount	If a discount is applied, show the Percentage Discount field.
$if ($contract_amount > 100000) ? $related123456 : $customer. related345678	If the contract amount is greater than $100,000, show the field $related123456; else, show the field $related45678 in the linked customer record.
$if ($include_appendix == "Yes") :	If an appendix should be included, merge the file held in the Attached

$merge($attached_file)	File field.

Conditional Paragraphs using $startif

Use the `$startif()` command in a document template if some paragraphs of text should only be included under certain conditions. For example, suppose a certain liability clause only appears in a final contract when the contract amount is over $50,000. The conditional text comes after the `$startif($condition)` and is followed by `$endif` to close the statement.

> If your Clause Library Table stores clauses in Microsoft Word format, and you want to use $startif in your document templates, create a new global variable named print_template_preprocess_startifs and set it to Yes. This eliminates some formatting issues that can occur during processing.

Syntax

```
$startif($condition)
Paragraph 1
Paragraph 2
$endif
```

Examples

$startif ($contract_amount > 50000) Contracts over $50,000 require an additional insurance certificate. $endif	If the *Contract Amount* is over 50,000, the paragraph appears in the final document. If the *Contract Amount* is less than 50,000, the paragraph is deleted.
$startif ($extended_warranty==" Yes") Your extended warranty is in effect until $formula ($warranty_end_date).	If the *Extended Warranty* field is Yes, insert the sentence listing the *Warranty End Date.*

$endif	
$startif(($service_name ~= "Installation")&&($flag = "Yes")) Lorem ipsum dolor sit amet. $endif	If the Service Name field contains "Installation" and the Flag field is set to Yes, then show the paragraph. This example combines conditions using && and extra parentheses surrounding the conditions.
$startif(find("Amended", concat("", $contract_updates)) Lorem ipsum dolor sit amet. $endif	Checks if the value "Amended" is found in a multi-value choice field "Contract Updates". The `concat` function gets a string representation of the values in the multi-value field, and if it contains the value "Amended", returns the text below. To handle cases where none of the values are selected, you can add the following after the startif: || isEmpty($contract_updates)) `isEmpty` returns "true" and outputs the text if none of the values are selected.
$startif(!($territory~="310")) Territory is not in Japan. $endif	If the Territory field does not contain "310", the system outputs the text.
$startif ($training_completed=="True") Please remember to upload your training completion certificate. $endif	If the Training Completed field is set to True, the system outputs the text. In this case, the Training Completed field is a Singleton Check datatype, which requires a capital "T" in "True" or capital "F" in "False" to work correctly in a `$startif()` command.

isEmpty

isemptyexcerpt

The `isEmpty` operator checks whether a field has a value or not, and is formatted as follows:

```
isEmpty($field)
```

This returns `true` when the $field is null or empty and `false` if the field has a value.

New line or insert line break

linebreakexcerpt

To insert a new line or a line break, use `\n` within a document template or field value formula. This can help start new paragraphs in certain cases.

Merge Documents

The `$merge` command can be used to merge multiple files held in a single field, or files held in multiple fields into one final document.

Using the `$merge($fieldname)` command within a document template will merge all attached .docx/.html files from the field in the current record into the resulting document template's .docx file. Documents held in a versioned file field will be merged in the order in which they were uploaded or appended to the field.

Syntax

To merge multiple documents held in a single file field with versioning or multiple files enabled:

```
$merge($fieldname)
```

To merge multiple documents held in separate file fields, use the format:

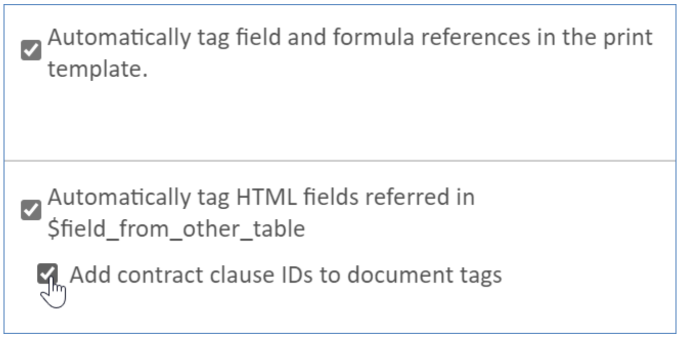
```
$merge($field1) $merge($field2) $merge($field3) ...
```

This makes it possible to compile a large PDF or Word document from several attachments.

The $merge command must be placed within the body of the document template. This function will not work if the $merge function is placed within the header or footer of the template, or within a table.

Integration

You can choose to configure a document template to automatically push changes that occur to field values or formulas of the Word document back to the pre-existing record in the KB. You can tag field and formula references, as well as references to fields from other tables. This is a valuable feature that ensures the value in the document template and the value in the actual record stay synchronized. These tags are also recognized as annotations when the generated documents are used with the Agiloft add-in for Microsoft Word. The options for tagging data can be found beneath the Published section of the Word/PDF Template:



☑ Automatically tag field and formula references in the print template.

☑ Automatically tag HTML fields referred in $field_from_other_table

☑ Add contract clause IDs to document tags

Agiloft Contract Assistant print template settings

You can select one or more options:

- **Automatically tag field and formula references in the document template**: this option is used to tag metadata in the document template.
- **Automatically tag references to $field_from_other_table**: this option is used to tag contract clauses in the document template. You can choose to use Contract Clause table variables within tags for identification purposes and additional detail.

The "tags" mentioned by the system text are actually ContentControl objects, which are the colored boxes you see in Microsoft Word when something has been labeled. When the document is opened in Microsoft Word with the Agiloft add-in installed, the tags hidden in the document's text by the options above are identified and listed as annotations on the Annotations page of the add-in. This can be used to automatically generate annotations in the add-in without the use of AI.

Bookmarks in Templates

You can add bookmarks to Word templates and configure your system to preserve the bookmarks in the finished output. To do this, go to **Setup > System > Manage Global Variables**, open the Customized Variables tab, and select **New > Choice Constant**. Create a new global variable named `print_template_remove_bookmark` and set it to No.

Bookmarks can only be preserved in Word-format document templates, not PDF or HTML formats.