

Embedding Tables and Fields from Tables

Embedded tables, such as [Related Tables](#), [Link to Selected Fields from Other Table](#), or [Embedded Search Results](#) can be included in document templates. You can allow the template to automatically format the embedded table, or you can enforce custom formatting explicitly in the document template. Both methods are described below.

This article also explains how to include an embedded table with a specific saved search, and how to embed fields from other tables.

Automatic Table Formatting

If you place the related table like any other field, the template automatically generates a table for you. Insert the formula and embedded table on a separate line from the body of the text.

```
$formula($related1231231231)
```

When you use this method, the system formats the table with the columns from the embedded table's view, and it automatically calculates appropriate cell widths. With automatic formatting, you don't need to draw a table or define columns in the document template at all. However, if you aren't happy with the automatic formatting in the final product, you need to use [custom formatting](#).

Custom Table Formatting

You can define your own table format, labels, and columns, and set properties like cell width in the document template file.



When building an embedded table with custom formatting, you can use the [Agiloft Contract Assistant for Word](#) to insert field variables, but it can't construct the whole table for you.

Look Up Variable Names

Before you begin, you can save time by looking up the variable names of the fields you want to use.

- For related tables or embedded search results, look in the target table for the logical field name used to insert the table. Then, look at the source table to find the logical field names used to define the columns.
- For links to selected fields, look in the target table for the logical field name used to insert the table. Stay in the source table, edit the linked field, and go to the Fields tab to find the logical names you can use to define the columns.

Add the Custom Table

To add a table with custom formatting:

1. Insert the field variable for the embedded table, such as `$related123123123` or `$link_to_selected_fields`. Note that for a link to selected fields, you need to use the link column name found on the Fields tab of the field wizard.
2. Directly below the variable, create a table in Word. The system will use this table to format the information in your embedded table.



If you want to include a header row, go to the table in Agiloft and go to **Print Records > Manage** in the action bar. Edit the Main Template, select "Include column headings in table," and click Finish. If you don't want to use a header row, select "Include only table data without headings" instead. This controls the default setting for table headings in all document templates in the Contracts table.

Table Column Headings

☒ Include column headings in table

☐ Include only table data without headings

☐ Expand table to fit page width

3. Add and format one, two, three, or more rows as needed:

- One row: The entire embedded table is formatted to match this row. You must manually add the related table variable, `$related1231231231`, but you can use the Agiloft app for Word to insert the variables inside each cell.

\$related1231231231					
Sid	Sitem_name	Sunit	Sprice	Squantity	\$total_price

- Two rows: If you select **Include column headings in table**, the first row is used as a header row in the embedded table and the rest of the table is formatted to match the second row. You must manually add the related table variable, `$related1231231231`, but you can use the ACA for Word to insert the variables inside each cell.

\$related1231231231					
Sid	Sitem_name	Sunit	Sprice	Squantity	\$total_price
Sid	Sitem_name	Sunit	Sprice	Squantity	\$total_price

If you select **Include only table data without headings**, odd rows are formatted to match the first row and even rows are formatted to match the second row.

\$related1231231231					
\$id	\$item_name	\$unit	\$price	\$quantity	\$total_price
\$id	\$item_name	\$unit	\$price	\$quantity	\$total_price

You can also create a static header, where the table headings from Agiloft are ignored. Consider this example of a link to selected fields with MVE:

\$contract_to_contract_party			
Company Name	Street Address	City	Country
\$company_name	\$address_1_and_2_comma_formatted	\$city	\$country

- Three rows: The first row is used as a header row in the embedded table, odd rows are formatted like the second row, and even rows are formatted like the third row. Below is an example of the template with a static header, where the second and third rows define the format for odd and even rows. You must manually add the related table variable, \$related1231231231, but you can use the ACA for Word to insert the variables inside each cell.

\$related1231231231					
ID	CUSTOM LABEL		PRICE	N	SUM
\$id	\$item_name	\$unit	\$price	\$quantity	\$total_price
\$id	\$item_name	\$unit	\$price	\$quantity	\$total_price

- Additional rows: Any extraneous rows are included in the final document, but not used to format the embedded table. For example, you might include a row with the total cost. You can also include field variables in extra rows, but only for fields in the same table.
- Position the fields by inserting the appropriate field variable in each column heading.
 - Test your template to make sure the table size, design, color, and column width look appropriate with data. If you include File with Versioning fields in the table, make sure the image sizes are appropriate to the table size.

Nesting Embedded Tables

You can nest embedded tables if necessary, using the same syntax.

Example

In the example below, the first related table, \$related123456789, then contains data from a second related table, \$related456789123.

This example also includes an advanced LISTNUM function to make sure the nested table numbering always starts over at (i) in each alpha row. You can add this function from the Insert ribbon using Quick Parts > Field. In the list of options, select LISTNUM, click Field Codes, and add \1 1 \s 0 after the LISTNUM text. After the function is inserted, simply set the text color to white and the font size to a very small value so that the codes doesn't appear in final documents.

\$related123456789	
	Column Header
(a)	From the \$alpha_name \$related456789123
	Column Header
	(i) \$bravo_name
	(ii) \$bravo_name
	{ LISTNUM \1 1 \s 0 }
(b)	From the \$alpha_name \$related456789123
	Column Header
	(i) \$bravo_name
	(ii) \$bravo_name
	{ LISTNUM \1 1 \s 0 }

Saved Searches

You can add a saved search to the related table in the document template using the following format:

```
$related123456789 ( search: LABEL_NAME )
```

To find the logical label of a saved search, edit the search, go to the Options tab, and locate the Label field:

Save Search	
<input checked="" type="radio"/> Yes	
*Name	attachment Active Attachments Only
*Label	All Active Attachments - Ignore Superseded

An example of this would be:

```
$related123456789 (search:All Active Attachments - Ignore Superseded)
```

Insert Fields From Other Tables

Use the following syntax to insert variables from records in other tables. This syntax differs from standard field variable chains because the inserted variable does not have to be connected by a linked record. For example, this makes it possible to insert text or other data from a specific record into a document template or formula in any other record, such as adding a contract clause to contract templates using variables from a Clause Library table. You can even insert File with Versioning fields to add Microsoft Word .docx files that contain clause text.

When you add a clause using a Word file, you can optionally add a parameter at the end. Available parameters include:

- `customaltchunk=true` to use `AltChunk` and preserve the clause's text formatting instead of using the document template's formatting; this is also useful if the clause uses HTML-formatted bullets or numbering, which are not recognized by Word
- `use_clause_word_file_num_style=true` to preserve the clause's numbering style and order instead of matching the document template's numbering
- `inherit_num_levels=[n n+1 n+2 ... n+m]` in conjunction with `use_clause_word_file_num_style=true` to continue a numbered list in the document template using numbered content in the clause; make sure the numbers listed are positive values, and note that the numbers listed confer the level of the numbering, not the numerical value shown in the template
- `nodelempypara`, not set to true or false, to preserve empty paragraphs, if any are created by your formulas and must be preserved for readability or other formatting reasons; this is rarely used
- `skipformulatagging=true` to preserve formatting like bold text that is placed on only part of a formula; this is rarely used

Syntax

```
$field_from_other_table($table,$id,$field_name,param)
```

Word File Examples

Consider the following table setup:

- Primary table: Table running the document template
 - Table name: `contract`
 - Fields: `contract_term_in_months`, `contract_start_date`, `contract_end_date`.

- Secondary table: Table referenced by main table
 - Table name: clause
 - Fields: clause_ms_word_file, clause_title
 - Record IDs: 44, 46, 48

Note: In the above example, the clause_text and clause_title fields are not directly or indirectly linked to the contract.

Example	Description
<code>\$field_from_other_table(clause, 44, clause_ms_word_file)</code>	Insert the file contents of the clause_ms_word_file field, from record ID 44 of the Clause table, into the printed document of a contract. Depending on the document template settings, the document template chooses either the formatting styled in the document template itself, or it retains the styles used in the clause file itself.
<code>\$startif (\$contract_term_in_months > 12) \$field_from_other_table (clause, 48, clause_ms_word_file, customaltchunk=true) \$endif</code>	Show clause text from record ID 48 if the Contract Term in Months is greater than 12, using a \$startif condition to reference the local variable. In this case, the customaltchunk=true variable is used to manually apply AltChunk to this specific clause, which means the formatting in the clause's Word file will be used instead of the document template formatting, regardless of the setting at the document template level and the global variable settings.
<code>\$field_from_other_table (clause, \$linked_field_clause_id, clause_ms_word_file)</code>	Insert the file contents of the clause_ms_word_file field, from the Clause record selected by the user in the linked_field_clause_id field. Depending on the document template settings, the document template chooses either the formatting styled in the document template itself, or it retains the styles used in the clause file itself.

HTML Text Examples

Consider the following table setup:

- Primary table: Table running the document template
 - Table name: contract
 - Fields: contract_term_in_months, contract_start_date, contract_end_date.
- Secondary table: Table referenced by main table
 - Table name: clause
 - Fields: clause_text, clause_title
 - Record IDs: 44, 46, 48

Note: In the above example, the clause_text and clause_title fields are not directly or indirectly linked to the contract.

Example	Description
---------	-------------

<code>\$field_from_other_table(clause,44,clause_text)</code>	Insert the value of the <code>clause_text</code> field, from record ID 44 of the <code>Clause</code> table, into the printed document of a contract.
<code>\$startif(\$contract_term_in_months >12)\$field_from_other_table(clause,48,clause_text)\$endif</code>	Show clause text from record ID 48 if the Contract Term in Months is greater than 12, using a <code>\$startif</code> condition to reference the local variable.
<code>\$field_from_other_table(clause,\$linked_field_clause_id,clause_text)</code>	Insert the value of the <code>clause_text</code> field, from the <code>Clause</code> record selected by the user in the <code>linked_field_clause_id</code> field.

Notes

The `$field_from_other_table(clause,44,clause_text)` string cannot be used for statements that use the `$if` condition. For example:

```
$if($contract_term_in_months < 12)? $field_from_other_table(clause,44,clause_text):
$field_from_other_table(clause,48,clause_text)
```

However, you can use two pairs of `$startif` and `$endif` statements instead. For example:

```
$startif($contract_term_in_months < 12)$field_from_other_table(clause,44,
clause_text)$endif $startif($contract_term_in_months >=12)$field_from_other_table
(clause,48,clause_text) $endif
```

- HTML fields inserted into document template, use the Normal Web style and will eliminate any numbering or bullet styles applied in the document template document.
 - To insert text with numbers or bullets, use a table in MS Word with a column for the numbering or bullets and a column for the clause text.
 - To create consistent styles in your document template, choose the appropriate font and make it your Normal Web style.
- When you use field variables from a non-linked table, such as the `clause_text` field in this example, that field variable may contain variables that reference the main table, such as `$contract_start_date` or `$contract_manager.cell_phone`.
 - When a document template on the `Contract` table uses a statement that references a field in a different table, such as `$field_from_other_table(clause,44,clause_text)`, the system first evaluates the `clause_text` field from `Clause` table's record 44, and then resolves the included reference to `$contract_start_date` from the contract record where the document template runs.
- While resolving variables from a secondary table record, the view permissions of the field are checked. If the user running a document template does not have view permissions for the field being referenced, then that field value is not resolved in the resulting document.

- If a formula is not resolved, the line is left blank. For example, if you use the `$startif` condition and the condition is not met, the line will be left empty in the resulting template.