

2.9 - Document Templates

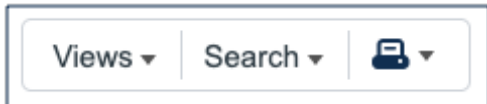
CONTENTS

1. Document Templates for Word and PDF	3
1.1 Creating Document Templates	5
1.2 Embedding Tables and Fields from Tables	14
1.3 Document Template Syntax Reference	22
1.4 Troubleshooting Document Templates	35

Document Templates for Word and PDF

Document templates, formerly called print templates, can be used to create letters, invoices, contracts, or other documents in Word, PDF, or HTML format. Your browser either opens the document or asks to save it, depending on your settings. Templates must be created and published before they are available for most users. A user's [group permissions](#) settings determine whether they can generate documents from templates.

Power users can access existing document templates, create new templates, and manage all templates from the Printer icon in the action bar.



For information about document templates in the default Contracts table, see the Create Contracts section of the [Contracts Table](#) article.

Uses for Document Templates

After you've set up a document template, you can use it to:

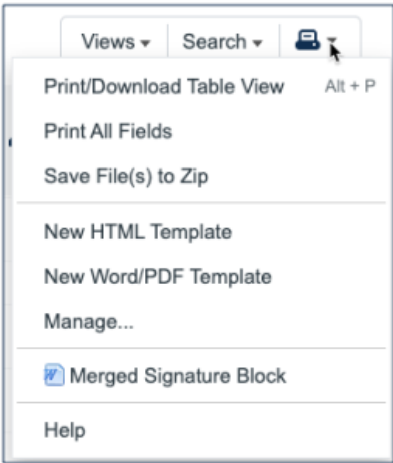
- Create a document and attach it to a record
- Run a template to create a document preview, to be manually saved or attached
- Run a template on multiple records and combine the output into a single document, to be manually saved or attached
- Merge attached files into a new document

Document templates can be run manually, but more commonly they are run automatically or by clicking an action button. For more information about printing with an action, refer to [Document Actions](#).

To manually create documents from a template...

1. Select one or more records in a table view.

2. Then, click the print icon and select the template name.



Creating Document Templates

Document templates, formerly print templates, make it easy to create documents using consistent, standardized language while inserting some information specific to the situation. For example, you might have an NDA template that includes standard and approved contractual text, with the parties' names, addresses, and other information inserted based on the data in the record. This makes it easy to generate standard documents quickly and with confidence that they follow the necessary procedures.

Document templates are usually managed in the [Document Templates Table](#), within the Agiloft app for Word, or from the action bar of each table. Hover over the printer icon from the table view of records, then select New HTML Template or New Word/PDF Template to create one, or click Manage to see the templates already created for that table.

If you want to create a PDF output, you'll still use a Word file to create the template, although the PDF output generally alters the formatting. When creating a PDF template, be sure to test the output thoroughly.



This article discusses how to create a document template in any given table. For information about templates included in the default Contracts table, see the Create Contracts section of the [Contracts Table](#) article.

Format Considerations

Each format has benefits and drawbacks. Here are some common considerations for each format:

	Word	PDF	HTML
Pros	<ul style="list-style-type: none">▪ Output is fully editable and easy for power users to revise▪ Can include images▪ Can include headers and footers▪ Can include mathematical formulas based on field values in the record▪ Can include bookmarks using the <code>print_template_remove_bookmark</code> custom variable▪ Can use content controls to label field and formula data▪ Can be built with the Agiloft Contract Assistant for Word	<ul style="list-style-type: none">▪ Output is not editable and therefore more secure▪ Can include headers and footers	<ul style="list-style-type: none">▪ Highly configurable▪ Can include images▪ Can create barcodes▪ Can include mathematical formulas based on field values in the record

Cons	<ul style="list-style-type: none"> ▪ Editable output is less secure, but you can use a locked template to automatically track any changes to the final document 	<ul style="list-style-type: none"> ▪ Non-editable output is difficult to revise ▪ Formatting is more difficult, and some formatting is impossible ▪ Must use Word tables with invisible borders to align text, rather than using Tab 	<ul style="list-style-type: none"> ▪ Requires knowledge of HTML to set up the template well ▪ Not a standard format for contracts and other common documents
------	--	---	--

Building a Template

Document templates are usually used to insert information from the record into the greater context of a written document. For example, you might have a standard NDA template, which you then use to generate NDAs for specific parties by inserting the party information from the record.

When you build a new document template, you usually begin with one of these source materials: an existing finalized sample document, an existing document template, or a blank document. No matter where you begin, we recommend using the [Agiloft Contract Assistant for Word](#) to build the template. The steps here assume you have the Word app installed.

The Word app's template editor introduces the concepts of syntax mode and label mode, and can convert your document between those two modes. In order for the document to be recognized by the Agiloft template engine, the document must contain \$variable syntax. You can use the Scan and Convert button to identify any recognizable document template syntax, and synthesize it into labels. In label mode, it is easier to work with and understand the template, particularly to add or edit expressions, as the \$variable syntax expressions are replaced with their descriptive labels, and their coloration and visibility is improved. Whenever the Word app saves your document back to the KB, the document is converted to syntax mode if it isn't already.

Turning a Document into a Template

One of the easier methods for creating a new document template from scratch is to use an existing, finalized contract sample that is a good representation of best practices. For example, you might begin with an executed NDA that uses standard language for your organization, and does not have unusual modifications. You can also use these steps for a template created for a different system, or filled out by hand.

1. Save the sample document as a local file and open it in Microsoft Word.
2. Click the Agiloft logo in the Home ribbon.
3. Sign in to the KB where you want to create the document template.

4. Click Create new template.
5. Complete the basic information about the template:
 - a. Document Type: Ensure that the Template radio button is selected.
 - b. Title: Enter the title of the template.
 - c. Description: Give an optional description of what sort of contract this template represents, how it should be used, etc.
 - d. Status: Determine whether this document template should be Active or Inactive.
 - e. Publication Level: Determine who can view this document template.
 - f. Related To: Determine what records this document template is related to. If you select Contracts, you can then link this document template to specific contract types. This way, you can be sure you are only using a document template with the right kind of contracts.
 - g. eSignature Type: Select whether this document template is signed using Adobe Sign or DocuSign.
6. Click Create. After you click Create, you can still edit the details and link additional contract types, either here in Word or in the KB itself.
7. The pane now shows the template information you entered, and offers the option to Scan and Convert syntax into labels. Real-world documents don't generally include any document template syntax, so you don't need to use this option.

NONCOMPETITION AND NONDISCLOSURE AGREEMENT

From Clause Library (ID 93): This non-disclosure and non-competition Agreement (the "Agreement") \$startif(\$set_contract_start_date_to=="Specific start date")is entered into this \$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", YYYY",\$contract_start_date))\$endif\$startif(\$set_contract_start_date_to=="Date of last signature")will be effective as of the date of last signature\$endif, between \$internal_legal_entity, a Delaware corporation ("Buyer") and \$company_name, a Pennsylvania corporation ("Stockholder")

RECITALS

A. From Clause Library (ID 94): The Stockholder has acquired extensive knowledge of and experience in the business and/or operations of the Company through his status as a stockholder in the Company;

B. From Clause Library (ID 95): Concurrently herewith, the

contributed significantly to the creation and maintenance of the good will of the Company, and that nondisclosure of confidential information and a period of noncompetition are, therefore, necessary to safeguard the Buyer's interests arising in connection with the transaction which is the subject of the Purchase Agreement; and

D. From Clause Library (ID 97): Buyer and the Stockholder desire to enter into a noncompetition agreement upon the terms and subject to the conditions set forth herein.

From Clause Library (ID 98): NOW, THEREFORE, in consideration of the mutual promises and agreements contained herein, the adequacy and sufficiency of which are hereby acknowledged by the Stockholder, the parties agree as follows:

1. From Clause Library (ID 99): Noncompetition.

a. From Clause Library (ID 100): During the term commencing on the date of this Agreement and ending on the fifth (5th) anniversary of the date hereof:

i. From Clause Library (ID 101): the Stockholder will not, directly or indirectly, own, manage, operate, control, or participate in the ownership, management, operation or control of, or be

Contract Assistant

NDA with Complex Clause Nu...
Active • 12

Overview Tags

Description
NDA with Clauses and DocuSign Tags.

Publication Level
Internal

Related To
Contracts

Linked to Contract Types
Non-Disclosure Agreement

eSignature type
DocuSign

Tags [Tag syntax](#)

Conditions	0
Variables	6
Clauses	26
Signatures	4

[VIEW TAGS](#)

Add Save

8. Instead, start highlighting content in the document that needs to be generic for the template, or otherwise needs to be connected to the KB so that it can be kept up to date. These might include:
- Variables. These represent field values from the record. For example, you might tag variables like Company Name, Contract Start Date, or Contract Amount. To tag a variable, highlight the text in the document and then click Add > Variable.
 - Clauses. These represent standard clauses, or blocks of text, from your clause library. For example, you might tag a Limitation of Liability clause, or Force Majeure clause. To tag a clause, highlight the entire clause in the document and then click Add > Clause. Be careful with your highlighting, because all the highlighted text is replaced by the contents of the tagged clause.
 - Signatures. These support electronic signature by the document's recipients, constructed according to the syntax of the e-signature vendor you selected in step 5. To tag a signature, highlight the signature line and click Add > Signature.
9. As you go, you might encounter content, tagged or otherwise, that should only be included under certain circumstances. For these:
- a. First, enter or locate the text you want to show only under certain conditions.
 - b. Select the text, being careful to select or not select punctuation as desired, and then click Add > Condition.



This adds a \$startif function, for those familiar with document template syntax.

- c. Give a clear name to the condition so you can identify it in the ACA pane.
 - d. Select the variable, or field, that determines whether the text is shown.
 - e. Select the appropriate operator, such as "is" or "is greater than".
 - f. Enter the value you're evaluating. For example, to show text only when the Amended field is Yes, you would enter Yes for the value.
 - g. Click Save. The selected text is now included only if the specified condition is met. If the condition is not met, the text is not included in the output file.
10. Select or clear the checkboxes for different element types to show or hide them in the ACA. For large templates with many tags, this makes editing easier. For example, you might want to hide the variable tags when you finish with them, since there are often many discrete variables tagged in a template.
11. Below the element types, you can switch between Showing Labels or Showing Syntax.



Some complex expressions are not yet supported by the template designer and must be entered manually while Showing Syntax.

12. When you finish tagging, click Save a final time in the pane. If you want to save the file, you can, but a copy has already been saved to the document template record.
13. Finally, test your template to make sure the output looks correct. Document templates often require some [trial and error](#) to get the formatting to look exactly right.

Editing an Existing Template

If you want to edit an existing template using the ACA for Word, you can easily tag the existing document template syntax and make additional changes directly in the ACA.

1. Open the document template file in Word.
2. Click the Agiloft logo in the Home ribbon.
3. Sign in to the KB where you want to create the document template, and wait while the Word app retrieves information about the template.
4. Below the template information from the KB, click Scan and Convert. The Scan and Convert button appears whenever a template is open in the template editor, if the Word app isn't yet aware of any expressions in the document. Pressing the button scans the document for all \$variable syntax expressions, and then converts them into labels. Label view makes it easier to add or edit expressions as \$variable syntax expressions are replaced with their descriptive labels. Note that if you are using the Showing Syntax feature, this process does not change the appearance of the syntax.
5. In the Tags pane, you can see all the identified tags. Switch to Showing Labels to see natural language labels for all the identified tags. Keep in mind that some syntax isn't supported with natural language labels, but that syntax can still be parsed by the system even if you don't tag it, so you don't need to remove it or change it.
6. Add any variables, tags, signatures, or conditions as you go.
7. When you finish tagging, click Save a final time in the pane. This saves your changes to the document template record.
8. Finally, test your template to make sure the output looks correct. Document templates often require some [trial and error](#) to get the formatting to look exactly right.

Starting from Scratch

The Word app makes it easier to build a document template from a blank page, although it does depend somewhat on whether your system has a robust [Clause Library Table](#). Clause libraries offer a repository of approved, reusable clauses, which you can drop directly into a new template to borrow already-approved language. The Word app connects to the clause library to make inserting clauses very simple.

1. Open a new, blank document in Word. Save the file locally.
2. Click the Agiloft logo in the Home ribbon.
3. Sign in to the KB where you want to create the document template.
4. Click Create new template.

5. Complete the basic information about the template:
 - a. Document Type: Ensure that the Template radio button is selected.
 - b. Title: Enter the title of the template.
 - c. Description: Give an optional description of what sort of contract this template represents, how it should be used, etc.
 - d. Status: Determine whether this document template should be Active or Inactive.
 - e. Publication Level: Determine who can view this document template.
 - f. Related To: Determine what records this document template is related to. If you select Contracts, you can then link this document template to specific contract types. This way, you can be sure you are only using a document template with the right kind of contracts.
 - g. eSignature Type: Select whether this document template is signed using Adobe Sign or DocuSign.
6. Click Create. After you click Create, you can still edit the details and link additional contract types, either here in Word or in the KB itself.
7. The pane now shows the template information you entered, and offers the option to Scan and Convert syntax into tags. This option isn't useful with a blank document, so ignore it.
8. In the document, start adding content. Text that should be static (text that is never updated based on the record or the clause library) can be entered directly in the document. When you want to insert content that will update dynamically based on the record or on updates to the clause library, add them using the ACA.

These might include:

- Variables. These represent field values from the record. For example, you might insert variables like Company Name, Contract Start Date, or Contract Amount. To insert a variable, place the text cursor at the right location and then click Add > Variable.
- Clauses. These represent standard clauses, or blocks of text, from your clause library. For example, you might insert a Limitation of Liability clause, or Force Majeure clause. To tag a clause, place the text cursor at the right location and then click Add > Clause. Be careful where you place your cursor so that the clause is inserted correctly, particularly regarding paragraph breaks.



This feature uses the Clause Text field of the clause library records, so it is not compatible with clauses that are formatted as Microsoft Word files.

- Signatures. These support electronic signature by the document's recipients, constructed according to the syntax of the e-signature vendor you selected in step 5. To insert a signature line, place the text cursor at the right location and then click Add > Signature.
- Conditions. These use a variable value, or field value, from a record to determine whether selected text is included in the final output. Enter the text you want to show or hide dynamically, and then click Add > Condition. Name the condition for your reference in the ACA pane, then select the variable, operator, and the value you are looking for. For example, to show text only when the Amended field is Yes, you would enter Yes for the value.



This adds a \$startif function, for those familiar with document template syntax.

9. Select or clear the checkboxes for different element types to show or hide them in the ACA. For large templates with many tags, this makes editing easier. For example, you might want to hide the variable tags when you finish with them, since there are often many discrete variables tagged in a template.
10. Below the element types, you can switch between Showing Labels and Showing Syntax. Some complex expressions are not yet supported by the template designer and must be entered manually while Showing Syntax.
11. When you finish tagging, click Save a final time in the pane. This saves your changes to the document template record.
12. Finally, test your template to make sure the output looks correct. Document templates often require some [trial and error](#) to get the formatting to look exactly right.

Showing Syntax

The Word app assists you in building document templates with working syntax, but if you prefer, you can enter all the syntax yourself manually. In the Word app, Showing Syntax shows this syntax for editing and troubleshooting purposes.

You can include information from the selected record in the document template output. This ranges from ordinary fields to complex embedded related tables and attached files. If you're building a Word or PDF template, you can click Show Field Variables or Download Template to see all the available fields in the appropriate syntax.

For details about syntax, refer to [Document Template Syntax Reference](#). For details about embedding related tables, refer to [Embedding Tables and Fields from Tables](#).

Selectable Templates

If different records will use different formats, you can create a document template that pulls its formatting from a field instead of hard-coded file. With this setup, you can use a single document template configuration while accommodating different document formats.

For example, in a typical contract management scenario, the Contract Types table has an attached file field called Document Template File that holds or links to the available document templates. In each contract type, the relevant document template is selected in this field. When users select a contract type for a new contract, you can automatically pull in the selected format as part of the linked set.

1. Set up an attached file field in the table, either as a local field or a link to another table that stores all available document templates.
2. When you create the document template, load the template from a **Field** instead of a **File**, and select the attached file field. In this example, the attached file field would be Document Template File.

Locked Word Templates

If a PDF doesn't suit your needs, but you don't want a Word template to be freely edited, you can lock the template so that all changes are tracked. When the document is ready to be finalized, the person with access to the password can review all changes and accept or reject them as needed. This is also useful when you already have a Word template designed, but the template doesn't look right when you apply it to PDF formatting.

To lock your Word template for editing, configure your template as desired and then:

1. Go to the **Review** tab and click **Restrict Editing**.
2. Configure the options as desired:
 - Formatting restrictions. You can use this to limit formatting changes to selecting an allowed saved style.
 - Editing restrictions. Select one of these options to limit or track editing. To follow the example above and track all changes for later review, select Tracked Changes.
3. Click **Yes, Start Enforcing Protection**.
4. Set a password to unlock the restrictions. This is the password you'll need to distribute to people responsible for reviewing the changes and producing a final version of the document. You might also store the password in a field if you use a table to store all the document templates; in the default Document Templates table, you can use the Password for Locked Word Document field.

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.

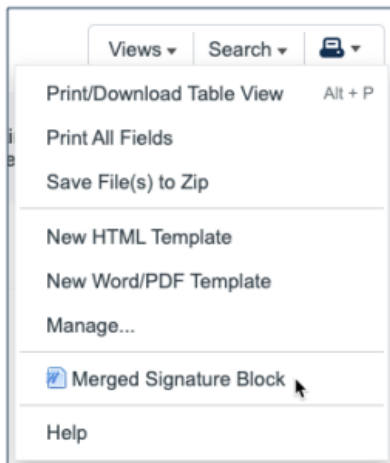
Bookmarks can only be preserved in Word-format document templates, not PDF or HTML formats.

Managing Document Templates

Document templates are accessed through a table's action bar. By hovering over the Printer icon, you can bring up the document template options. From here, you have a few different choices:

- New HTML Template and New Word/PDF Template open the wizards shown at the top of this page.
- Manage produces a screen with a list of existing templates and allows you edit them. You can also create new templates from here.

- Clicking an existing template's name runs the template with any selected records.



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, format the first row as desired and select Include column headings in table in the document template [wizard](#).

<p>Table Column Headings</p> <p><input checked="" type="radio"/> Include column headings in table</p> <p><input type="radio"/> Include only table data without headings</p> <p><input type="checkbox"/> Expand table to fit page width</p>
--

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

4. 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	
	<i>Column Header</i>
(a)	From the \$alpha_name \$related456789123
	<i>Column Header</i>
	(i) \$bravo_name
	(ii) \$bravo_name
	{ LISTNUM \1 1 \s 0 }
(b)	From the \$alpha_name \$related456789123
	<i>Column Header</i>
	(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
☒ 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 include the `customaltchunk=true` parameter to use AltChunk and preserve the clause's text formatting instead of using the document template's formatting.

Syntax

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

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 \$startif 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.

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.

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

<code>\$formula(\$login)</code>	Insert the Login value.
<code>\$formula(\$account_rep.backup_person)</code>	Insert the name of the Account Rep's backup person.
<code>\$formula(\$related123123123)</code>	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

<code>\$formula (\$contract_amount*\$discount_percentage)</code>	Multiply the <i>Contract Amount</i> by the <i>Discount Percentage</i>
--	---

```
$formula($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
<code>\$formula(concat("Your account representative is ", \$account_rep, "."))</code>	Your account representative is Hector Gomez.
<code>\$formula(concat(\$company_name, " Support Contract"))</code>	Agiloft Support Contract.
<code>\$formula(" " + \$field1 * \$field2 + " Total")</code>	2,142 Total
<code>\$formula(concat(\$contract_start_date, " to ", \$contract_end_date))</code>	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
"MMMMMM"	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
<code>\$formula(dateformat("d",\$contract_start_date))</code>	10. <i>Inserts which day of the month the contract starts.</i>
<code>\$formula(dateformat("MMMMMM",\$contract_start_date))</code>	February. <i>Inserts the full text name of the month.</i>
<code>\$formula(dateformat("yyyy",\$contract_start_date))</code>	2016. <i>Inserts the year of the contract start date.</i>
<code>\$formula(dateformat("yyyy",NOW()))</code>	<i>Inserts the current year when the document template runs.</i>

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
<code>\$formula(format("contract. display_field",\$contract_amount))</code>	Inserts the contract amount field formatted with the display characteristics of Display_Field.
<code>\$formula(format("quote. base_currency",\$total_amount*\$discount_percentage))</code>	Multiplies the Total Amount times the Discount Percentage and formats the output based on the Base Currency field's settings.
<code>\$formula(format("pricing_plan.monthly_price", ((\$number_of_users*16) + (\$number_of_units*5))))</code>	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" ?	Inserts the sentence "Authorized contractor' means a third

"Authorized contractor' means a third party contracted to develop or assist with the development of an Authorized Application." : "")	party contracted to develop or assist with the development of an Authorized Application." if the <i>Authorized Contract</i> 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 <i>Country</i> is USA and the <i>City</i> 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.	If the contract amount is greater than \$100,000, show the field \$related123456; else, show the field \$related45678 in the linked

related345678	customer record.
\$if (\$include_appendix == "Yes") : \$merge(\$attached_file)	If an appendix should be included, merge the file held in the Attached 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

<code>\$startif</code> <code>(\$contract_amount > 50000)</code> Contracts over \$50,000 require an additional insurance certificate. <code>\$endif</code>	If the <i>Contract Amount</i> is over 50,000, the paragraph appears in the final document. If the <i>Contract Amount</i> is less than 50,000, the paragraph is deleted.
<code>\$startif</code> <code>(\$extended_warranty=="Yes")</code>	If the <i>Extended Warranty</i> field is Yes, insert the sentence listing the <i>Warranty End Date</i> .

Your extended warranty is in effect until \$formula (\$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 <code>concat</code> 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 <code>startif</code> : <pre> isEmpty(\$contract_updates)</pre> <code>isEmpty</code> 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 <code>\$startif()</code> 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:

<input checked="" type="checkbox"/> Automatically tag field and formula references in the print template.
<input checked="" type="checkbox"/> Automatically tag HTML fields referred in \$field_from_other_table
<input checked="" type="checkbox"/> Add contract clause IDs to document tags

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.

Troubleshooting Document Templates

As you create and test your document templates, formerly called print templates, use these suggestions to help resolve problems. Note that many of these apply specifically to Word and PDF templates, rather than HTML-based templates.

Syntax Errors

Check for these common syntax issues:

- Parentheses and brackets. Make sure each one is part of a pair, with one opening and one closing the phrase. If you're working with many nested sets of parentheses or brackets, you can copy and paste the text into an HTML editor like Notepad++. These editors highlight the other half of a pair as you move your cursor, so it's easier to tell when a pair is unfinished.
- Smart quotes (""). Make sure only plain quotes (") are used inside formulas. Formulas won't work correctly with 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.

- Formula page breaks. Check each page break to make sure formulas aren't spilling across the break. Formulas broken between pages do not work correctly. If you have a formula near the end of the page, and you aren't sure whether it breaks across the page during testing, you can follow the steps in the Broken Paragraphs section below to force the formula to stay together.
- Nested \$startif() commands. If the page has a \$startif() call inside another \$startif(), either on purpose or due to missing parentheses, they won't work correctly.
- Microsoft Word-formatted clauses in a clause library used inside a \$startif() function. If you need to use this combination, 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.
- Formula validity. Test your formulas to make sure they work as expected. If you're using a formula for the first time, try testing it with simpler inputs, or test your inputs one at a time.
- Reenter the field. If you can't identify the problem, try deleting and retyping the field name, rather than copying and pasting. Pasting can include formatting and field codes behind the scenes that sometimes cause problems.
- Enforce formatting. Make sure the document has consistent formatting and font.

You can also try copying and pasting the entire document into a new Word document to remove ghost formatting.

Broken Paragraphs

There are often sections of document templates that need to stay together. For example, you might want to keep all signature lines together on the same page, so that if the preceding text gets too long, the entire signature section jumps to the next page together. You can make sure sections of the template stay together on the same page.

1. Highlight the entire section that you need to keep together.
2. Right-click the selection and click **Paragraph**.
3. In the Paragraph window, click the Line and Page Breaks tab.
4. Select the **Keep with next** and **Keep lines together** checkboxes.
5. Click **OK**.

Now, the selected paragraphs will always stay together as a whole.

Tables

Refer to [Embedding Tables and Fields from Tables](#) for information about including tables. If you're building a complex table, here are some formatting tips for working with tables in Word:

- To move an individual cell outside the alignment of its neighbors, double-click the cell before you drag its borders.
- If the ruler is enabled (View ribbon > Ruler checkbox), hold **Alt** while dragging a border to show the table measurements on the ruler. This makes it easier to center and distribute cells evenly.
- Tables tend to change shape when you enter long field names, or when the field contents is longer than the field name. To prevent this, right-click the table and select **Table Properties**. Click **Options** at the bottom and clear the **Automatically resize to fit contents** checkbox.

Styles

Make sure to use the Format Painter tool to apply a consistent style to your document template. In particular, use the Format Painter to apply a consistent style to formulas, starting at the \$. For example, if you have a \$startif formula with an \$endif, you should highlight everything from the first \$ to the final £ in \$endif.

If the document template's final output has style problems, but not the original document template, check the style applied to the text inserted to replace formulas and variables. If the style is different than the rest of the document, you might need to modify that style to make it match the style used in the rest of the text, or apply that style to the

text in the original document template. For example, inserting text from an HTML field is often automatically set to the Normal (Web) style in Microsoft Word, while the document template body is usually written in the Normal style. Setting the document template body to Normal (Web) or setting Normal (Web) to the same formatting as Normal can fix this issue.

- ✔ For best results, store clauses as Microsoft Word files and use the Document Template option to prioritize the template's formatting, instead of Use Global Settings or Field. For complex clauses, or clauses that contain images, set the specific clause to use AltChunk using the `customaltchunk=true` parameter. If you need a single clause to both use automatic numbering and include an image, the best method is to split the clause into two parts, so that the numbering can be handled using the document template formatting and the image can be handled using AltChunk.

If the text or list style in the final product still doesn't match the style in the document template, you might need to review the style preference settings for your system, the specific document template, and potentially the individual clause being inserted.

Troubleshooting Styles

First, if the issue comes from inserting a clause, and the clause is stored in a Microsoft Word file rather than an HTML text field, review the text of the document template to locate the variable pointing to that clause text.

- If it includes the `customaltchunk=true` parameter, that means the clause is rendered using AltChunk, a feature of Word that preserves the formatting in the clause file rather than converting it to match the template's formatting. AltChunk is necessary if you want to include images, or if the clause is very complex, but it is not compatible with automatic list numbering at the template level. In the template, look for this syntax: `$field_from_other_table(clause,161,clause_ms_word_file,customaltchunk=true)`
- If that parameter is included, try removing it; if it isn't included, try adding it.

Next, edit the Word/PDF template itself and check the style preferences drop-down setting at the bottom. This controls the style preference for the whole document template.

- Use Global Settings. Follows the settings in the global variables to determine how to format the contents of the selected field.
- Document Template. Applies the document template styles to any inserted fields. If a clause contains a bulleted or numbered list, that list is converted to the styles in the document template. This option does not support inserting images from clauses.
- Field. Uses Microsoft Word's AltChunk paradigm to embed the field's contents with the field's original styles applied. This option does not support automatic numbering.

Finally, if the lower-level settings weren't the problem, check the global variable settings. Log in as an administrator and go to **Setup > System > Manage Global Variables**.

- ⚠ Remember that any change to global variable settings affects all document templates in the system, not just the one you're troubleshooting.

- Locate the "Style preference for paragraphs in the output of a document template" global variable. Set this variable to `printtemplate` to use the style in the document template, or set it to `field` to use the style coded in the field.
 - If you set this variable to `printtemplate` and you still see problems with styling, check that all the styles used in the document template have the same font and font size. This includes both the problem text and the Microsoft Word style definitions, which you can access by right-clicking the active Style and clicking Modify.



When this variable is set to `printtemplate`, the document template still respects use of bold, italics, and underlining in the field content. If you need to override that behavior, go to the Customized Variables tab and select **New > Choice Constant**. Create a new global variable named `print_template_pref_useRunRpr` and set its value to No.

- If you set this variable to `field`, the formatted contents of an HTML field is retained in the final product. This is generally only used when users copy and paste formatted content from another program like Word directly into the field. If you use this setting, test your document templates thoroughly because Word doesn't recognize all HTML constructs, even if they are correct in HTML.
- Locate the "Use AltChunk while running a Generate Document Action" global variable. Set it to No to use the bullet and numbering lists formatted by Microsoft Word, or set it to Yes to ignore Word's automatic list functions and embed field content exactly as-is. In most cases, you want to set this variable to No, which allows usage of automatic numbering in document templates.

If you need to include bookmarks in the final product of a Microsoft Word template, you can configure the system to preserve bookmarks from document templates. To do so, go to the Customized Variables tab and select **New > Choice Constant**. Create a new global variable named `print_template_remove_bookmark` and set it to No. This setting applies only to Word document templates, not PDF or HTML formats.