# Clause Library Table

This table holds clauses that can be referenced while creating new contracts in the Word app. You can compare Contract Clauses to similar clauses in the Clause Library table, or even insert clauses from the Clause Library table into a contract. It can be thought of as a database of approved, vetted clauses commonly used by your company. There are many ways to use a clause library that vary in complexity. The out-of-the-box setup includes examples of several reasonably simple approaches.

There are different reasons for using a clause library to hold contract language rather than simply storing all text and variables in the Microsoft Word document template. Here are a few of them:

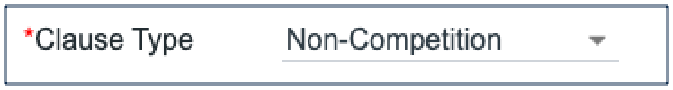
- If all source clauses are stored in a library, then if the clause text needs to be changed, it must only be changed in one place, and all document templates or records that use that clause will be updated for future contracts.
- Clauses may have specific conditions associated with them so that they are only used if some particular meta data exists in the contract.
    - Note that it is also possible to add such conditions to document templates, so that this does not necessarily require a clause library.
- Workflow can be configured to handle changes to standard language, with approvals, statuses, and so on. Responsibility for maintaining and updating the clause can be assigned to a particular team or clause owner.
    - Alternatively, document templates could also have approvals and statuses if language is managed directly in the document templates.
- With a clause library, users may be presented with a list of alternative clauses for particular sections or a list of optional clauses that they can select to construct a given contract. A simple version of such selection is built into the contract template to demonstrate these approaches.
- Some companies want to identify approvers who will need to review a contract based on a particular clause that is changed for a given contract. Approval information can be stored in the source clause to provide approval automation. This typically requires a more complex setup and is not included in the out-of-the-box Clause Library table.

Our out-of-the-box setup is focused on providing the first four benefits. It provides a simple repository for clause language that can then be inserted into document templates, either automatically, or with some user selection.

# Use Case

The clause library has several fields to assist in managing the use of language throughout your contracts. The key fields and their usage is as follows:

- **Clause Type -** Categorizing clauses makes it possible to filter them if you want to allow users to select from alternative clauses for a specific section of the contract, for instance the termination provisions, or the indemnification provisions.

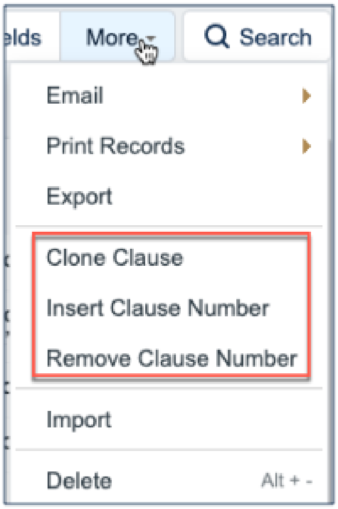


*Clause Type Non-Competition

- **Status** - Statuses of Active, Pending Approval, Retired, Replaced or Rejected can be used to filter available clauses and to indicate where they are in any approval workflow.
- **Default Risk Rating** - indicates the level of risk for this clause type on a scale of 0 (no risk) to 1 (highest risk).
- **Clause Priority** - is used to sort a drop-down list of clauses that may be filtered and presented to the user. If you have four variants for indemnification, for instance, and you want to display them in a particular order, you would set the Clause Priority with the appropriate values (1 for the highest to appear on the list).
- **Approval Team** - can be used to define the approval team with responsibility for updating this clause.
- **Clause Owner** - can be used to indicate a specific person responsible for changes to the clause.
- **Language** - allows you to select the language of the clause text
- **Has Multiple Languages** - selecting Yes displays a related table where you can link multiple child clauses in other languages.

# Action Buttons on the Action Bar

The action bar above the clause table has three special options under the More drop-down. These options are Clone Clause, Insert Clause Number, and Remove Clause Number



- **Clone Clause** can be used with one or more clauses to clone them and create a copy. The copies will map most of the field values and will add the date cloned to the clause title. This makes it easy to create copies of a clause for some purpose such as revision, and then to edit the resulting clause.

- **Insert Clause Number** adds MS Word code that results in auto-numbering in front of that clause when the clause is used in a document template. This is one method of managing clause numbering within your contracts. See below for more details.
- **Remove Clause Number** removes the MS Word code for auto-numbering from the selected clause(s). This is handy if you change your mind about the method to use for numbering.

# HTML Field Conversion into MS Word

The Clause Text field is an HTML based field and can use standard HTML formatting tags. To see all formatting options, click the Edit button below the clause text field to bring up the HTML editor.

If you copy and paste clauses into the clause text field from a Microsoft office file such as MS Word or Excel, it is best to use the Cleanse HTML button to strip out some of the excess code that is typically inserted by Microsoft. This will give a cleaner result when the clause is inserted into a document template and converted back into MS Word.

> HTML text fields are treated in a special way in Microsoft Word. Depending on specific HTML field characteristics, content may be translated in the Word document to either the Normal or Normal(Web) formatting style. Including <p> codes in the paragraph results in the Normal(web) formatting. It is important to configure any document template in Word to set the same font, size, and spacing style for Normal (Web) as for Normal, or for whatever other main style is used in your template, for consistency. For instance, if your template has some hard coded text, followed by variables that point to specific HTML clauses, you would want to define Normal (Web) to look the same as the hard coded text style. For more information, see Troubleshooting document templates.

# Methods for Auto-Numbering Contract Paragraphs

In example document templates, the numbering and indentation style is set in the document template, applied to the relevant clause variable. However, sometimes the HTML text that replaces the variable in the template does not retain the numbering or indentation.

If document template formatting doesn't work for you, you can choose to insert the auto-numbering formatting inside the clause text, instead of in the document template. To do so, edit the clause in Source HTML and insert the MS Word auto-numbering code at the front of the clause text. When the document is generated, this code automatically assigns a sequential number and applies list indentation. Note that if you use this method, these numbers are reassigned every time the document is opened, which means the clauses are always shown as recently modified when a contract is edited.

**Auto-numbering Code**

```
<agiloft_tag_fc fc="al_fc_start LISTNUM LegalDefault \l 1 al_fc_end"><
/agiloft_tag_fc>
```

In the standard implementation, we handle numbering in the document template, not the clause text. Whichever method you choose, make sure to test it thoroughly.

# Using the Clause Library in Contract document templates

The standard setup of the clause library assumes that clauses may be created for any language that may be used in one or more contract types and document templates. It may make sense to put most of your contract language into the document template and just use the clause library for user-selectable alternative text or optional text. The choice is yours.

> ⓘ If you do not reuse language in multiple document templates, there is really not much advantage to storing it in the clause library. In that case it is simpler to just put your language directly in the document template, where it does not need to be translated back and forth between Word and HTML and where users looking at the document template can see exactly what will be produced. It is easy to provide conditions directly in the document template for alternative or conditional clauses.

As you define your clauses, you can link them to the specific document templates where you plan to use them. Then they will appear within the document template in a related table of Clauses Included, for reference.

Once the clauses are created, you can create a document template that inserts the conditions and the specific clauses you want. Please review Document Template Syntax Reference for details on the syntax to use to insert a clause into a document template.

# Allow Users to Select Alternate Clauses

Sometimes, you may want contract managers to be able to select from multiple clause options. There are a few different ways to do this, depending on how many choices they will need to make.

- If there are only a few user decisions, it may be simplest to add a link to the selected clauses within the contract record, filtered to the choices available. Then whichever clause is selected can be inserted into the document template.
- If there are many decisions, it may be easiest to just provide questions for the user to answer and then have conditions built into the document template based on the answers given. This makes maintenance of the document template more complex, but may make it easier for the users making the decisions.

We have provided two examples of user selection in the out-of-the-box setup.

## Selecting Optional Clauses to Add to the Contract

For the same contract types, the user can also select clauses that have been defined in the document template Clauses table using the Select Optional Clause(s) field. The user can select the clauses he wants to add, and when the document is generated, those clauses will be added under the section called Additional Terms. This field is only visible if the document template record has a value of Yes in the "Has Optional Clauses" field. For more information on document template Clause records, visit Document Template Clauses Table.

These basic approaches can be expanded upon for a given implementation, depending on the needs of your company.

### Conditional Clauses Based on Other Answers

The sample templates also include conditional clauses based on the choice made for the Renewal Type field. If Evergreen is selected, one clause is used, and if a different value is selected, an alternative clause is used that provides both the start and end date of the contract.

## More Complex Configurations

If greater control is needed over changes made to clauses within a specific contract, this requires one or two extra tables. In that case, it is generally necessary to have a "Contract Clauses" table that auto-generates new records for each clause linked to a specific contract type, so the contract clause record itself is edited, and then the document template just inserts all the clauses from the table of Contract Clauses.

For such implementations, we recommend that you contract with our professional services team for assistance. There are many variables in terms of requirements that help determine the optimal design.

# Workflow

There is currently no automation built out in the clause library, but there are fields that could be used to route approvals for new language to the approval team or the clause owner.