

Clause Library with Print Templates Sample Spec

This article is based on a design specification from an actual customer project in Agiloft.

Design Requirement

The customer KB should be able to compile and product a blank contract file from a clause library based on a number of parameters such as the selected contract type in the contract and solicitation records. The user only needs to set the metadata in the record correctly, then click the Generate Blank Contract button. The system then assembles the contract file and gives a browser pop-up once the contract generation is complete. The total assembly process takes about 30-40 seconds.

High Level System Design

There are four tables needed for the design:

- Solicitation Table (SOL) – Major processing table where the blank contract is generated.
- Clause Library (CL) – Background table holding all of the available clauses in the system. Clauses are composed in the HTML editor.
- Contract Type (CT) – Each Contract Type is loaded with a set of Contract Type Clauses and form templates.
- Contract Type Clauses (CTC) – Intermediate table which links the Contract Type with clauses from the Clause Library. A CTC record has the form `Contract Type - Clause Name`, for example
`Maintenance Contract - Premises/Assigned Area`.

In addition, the following Excel sheets and scripts can be used to import clauses and clean Word formatting from clause text:

- Contract Language Matrix
- Contract Type Clause Import Template
- Example CTC Import Template
- Cleanse HTML Script

System Design Example

Assume that we have overall 100 clauses stored in the CL table with ID 1-100; and we have two Contract Types, TypeA and TypeB. Contract TypeA uses CL ID 1-80; and Contract TypeB uses CL ID 50-100. Each CTC record is related to one contract type and one clause library record. For CL ID 1-80, we generate 80 records in the CTC table related to Contract TypeA; and for CL ID 50-100, we generate 50 records in the CTC table related to contract TypeB.

The first differentiator is the contract type. Each contract type has a pool of clauses it can use, stored in the CTC table. When the system generates the blank contracts, it is checking the CTC records instead of the CL records based on the selected Contract Type in the SOL record.

CTC records can have one of three types of CTC records:

Usage ☐ Required ☒ Conditional Determination Method ☒ System Defines ☐ User Picks

- Required CTC: The CTC clause is always required for the related Contract Type.
 - CTC Parameter: Usage = Required
- Conditional CTC, System Defines: The CTC clause is optional, but the system determines whether to include the clause or not based on the metadata in the solicitation record. For example, clause "ACDBE – Version 1" will be included if, in the Solicitation record, we have "Minority Business = ACDBE" and "Minority Business Goal Applicable = No Goal".
 - CTC Parameter: Usage=Conditional, Determination Method=System Defines. The Condition formula field holds the conditions using the metadata in the Solicitation record.

Note: "Usage" and "Determination Method" could be combined into one parameter.
- Conditional CTC, User Picks: The CTC clause is optional, but the system does not hold the information to determine whether to include clauses. Instead, users need to select the clauses they want to include in the solicitation record before generating the blank contract.
 - CTC Parameter: Usage=Conditional, Determination Method=User Picks. All records with these parameters are included in the Solicitation "Include conditional clauses" field for users to pick.

Conditional Clause to Include

Goods Contract-Condernation
Goods Contract-Disadvantaged Business Enterprise Asurance and Paticipation - Version 1 - No Goal
Goods Contract-Disadvantaged Business Enterprise Asurance and Paticipation - Version 2 - Goal
Goods Contract-Environmental
Goods Contract-Improvements and Alterations

Each CTC record must belong to one of the three types above.

The second layer of differentiation between CTC records is based on the metadata in the Solicitation record.

Blank Contract Generation Workflow

Each CTC record has a Sequence value. The Sequence defines the workflow and so must be unique for a specific Contract Type. For example, for Contract TypeA, we cannot have two CTC records with "Sequence = 1". Instead, we can make one CTC sequence "1.0", and the other sequence "1.1".

Sequence	3.2
----------	-----

When the system generates a blank contract, it goes to the first CTC record. After judging and converting the first CTC, the system goes to the next CTC with the next highest sequence. The process goes on until all the related CTC are judged and appended to the Solicitation "File Content" field. At the end, the system takes the file content and compiles it into a file to generate the blank contract.

Features

- System generates the file completely using the Clause Library.
- Contracts can have "Required" and "Conditional" clauses
- The generated file makes use of dynamic numbering grouped by article number. For more information, see [Dynamic Numbering](#).

Limitations

- The contract must be generated completely from the Clause Library. This design is not good at inserting clauses into certain places in a contract template.
- Clauses must follow a determined sequence, distinguished by the Contract Type. Please see the [Contract Language Matrix](#) for further examples.
To work with conditional sequences, see the [example below](#) for a possible solution.
- It does not have an approval mechanism for contract clause changes, so if approvals are needed for a specific contract's language, it cannot be done directly. Instead, users have to work with the generated Word document.

Possible Enhancements

- It is possible to insert variables into the generated contract. To do so, use the generated "Blank Contract" as an intermediate file, and generate the final contract with variables. In this case, the variables should be included in the Clause Library/CTC Clauses fields.
- We could include the ability to approve changes at the CL and CTC levels.

Conditional Sequencing Suggestion

As an example, we have ClauseA and ClauseB. Under Condition A, Clause A goes in front of Clause B; Under Condition B, Clause A goes after Clause B; and only one of Condition A or Condition B will be met under any circumstance. In this case, we could set up three CTC records to work with conditional sequencing. Note that this approach is not recommended because it could make the system hard to maintain.

CTC	Related Clause	Usage	Condition	Sequence
CTC A1	Clause A	Conditional	Condition A	1
CTC B	Clause B	Conditional	Required	2
CTC A2	Clause A	Conditional	Condition B	3

System Maintenance

The sections below will help with managing certain aspects of the clause library system design.

Dynamic Numbering

Dynamic numbering of the contract articles can be set up by the system, to a limited degree. For this, you will need to store the clause text differently than how it appears in the contract. For example, a contract contains clause numbering as below:

ARTICLE 3	
<u>TERM</u>	
3.01 Effective Date	This Contr Authority. deemed as
3.02 Term	
3.03 Commencement of Fees and Charges	
3.04 Commencement of Operations	
3.05 Renewal Option	

All the "3"s in the text can be dynamically numbered, so that if the section is generated as the fourth section in the contract, it will display "Article 4", and "4.01 Effective Date".
To set up this capability, when storing the clause into the system just need to replace the heading numbers with _num, so the data stored in the Clause Library looks like:

ARTICLE _num	
<u>TERM</u>	
_num.01 Effective Date	
_num.02 Term	
_num.03 Commencement of Fees and Charges	
_num.04 Commencement of Operations	
_num.05 Renewal Option	
_num.06 Early Termination	

Update Existing Clauses for all Contract Types in the Clause Library

To update the language in an existing clause, go to the Clause Library table and edit the target record. This process must be done manually and one at a time, because attempting to update clauses using Excel will cause Word formatting to be lost.

1. Turn related CTC auto-update on or off

In the CTC record, the "Should System Auto-Update Clause" field determines whether the system will update the clause in the CTC record if the underlying Clause Library's wording changes. Modify all CTC records to hold the correct Yes or No value for this field before updating the Clause Library record.



2. Update the target Clause Library record

Go to the Clause Library record and update the Clause field by either editing it directly or copying the updated clause from a Word document.

3. Click the Cleanse HTML button then Save

The Cleanse HTML button causes the formatting to be retained in the generated contract file. Once this has been done, click Save to save the update.



4. Once saved, the system updates the clause language in all related Contract Type Clauses where the auto-update field was set to Yes, and leave the ones with auto-update set to No untouched.

Customize a Clause for a Specific Contract Type

For a specific article (clause), each contract type can hold a different version of a clause.

1. Go to the Contract Type Clause record, and modify the Clause directly in the record.
2. Remember to set Should System Auto-Update Clause to No if you want to turn off the auto-update for this CTC in the future.

Add a new Clause to the Clause Library

If we have new articles (clauses) you want to add into the system, you firstly need to create the Clause Library record for it. There are two ways of creating background clauses:

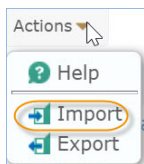
Method 1: Manually add a Clause

1. If you are adding a small number of clauses into the system, go to the Clause Library table and create a new record.
2. Give the new record a title for future reference, and copy the clause text into the Clause field.
3. Click Cleanse HTML button.
4. Save the record.
5. Continue creating records as needed.

Method 2: Batch Import a Clause

If you are adding a lot of background clauses into the system, you can use a batch import in the Clause Library table to create the records. However, there is normally formatting in the clause text, the import function can only be used to create the records, and the custom formatting must be done manually.

1. Add new Clause Library records via an imported spreadsheet.
 - a. Create a spreadsheet with Column A - Clause Title.
 - b. In the table, select **Actions > Import** to import it to create the clause library records.



2. Manually update the clause language in the Clause Library records.

Go through the clause library records and paste the text in the Clause field, generally with the Word formatting included.
3. Click the Cleanse HTML button
4. Save the record.

Add New Contract Type Clause

CTC records hold a pool of clauses that can be used for certain contract type. If you need to create a CTC record, you must make sure the target clause already exists in the Clause Library table, and the contract type already exists in the Contract Type table.

There are two ways of generating the CTC records:

Method 1: Manually create new CTC records

1. If you are adding a small number of CTC records, go to the Contract Type Clauses table and create a new record.
2. Enter the basic information in the record including:
 - a. Related Contract Type
 - b. Sequence number

Note: the sequence needs to be unique for certain Contract Types. For example, for Goods contracts there should only be one CTC records where the sequence=3.

For conditional CTCs with multiple versions, you can assign a decimal number to the sequence. For example, ACDBE - Version 1 can have sequence 18.1, while ACDBE - Version 2 can have sequence 18.2.
 - c. Related Clause Title
 - d. Usage, Determination Method and Condition.
3. Click Save. You will get an error message if the sequence you entered already exists. Modify the sequence and save again.

Method 2: Batch Import Contract Type Clauses

If you need to generate a large amount of CTC records - i.e, when you create a new Contract Type - you can use an Excel spreadsheet to create multiple records via import. Please pay attention to all the steps in this section. You can use this [Contract Type Clause import template.xlsx](#) file for the import.

1. Export all records in the Clause Library table.

When exporting the Clause Library records, only include the Clause Title and ID fields.

 - a. Copy the Excel spreadsheet into the ClauseLibrary sheet in the import template.
 - b. Column A should be Clause Title and Column B should be ID.

2. Add values to the CTC spreadsheet.

When importing the spreadsheet, each row in the CTC sheet becomes a record in the CTC table.

- a. Copy Clause Title into Column C and Sequence into Column F
- b. Put Contract Type ID into Column A - Related Contract Type ID
- c. Modify the sequence, ensuring that it is unique for a given contract type.

The All Conditional System Clause sheet in the spreadsheet contains an example with the ACDBE and WMBE clauses.

- d. Determine values for Column B - Related Clause ID
Use this function in Column B to get the system ID for the clause:

```
=VLOOKUP(C2,ClauseLibrary!$A$2:$B$500,2,FALSE)
```

Clean up all N/A values by updating the records in Column C matching the Clause Title in spreadsheet Column A.

- e. Determine values for Column D (Usage) and Column E (Determination Method).
 - i. For all CTC records that are required for the specific Contract Type use Required for Column D and leave Column E empty.
 - ii. For all CTC records that are conditional and the system makes the decision based on certain values of the SOL metadata, use Conditional for Column D and System Defines for Column E.
 - iii. For all CTC records that are conditional and the user picks clauses every time, use Conditional for Column D and User Picks for Column E.
- f. Save the Excel file.

3. Import the spreadsheet.

- a. In the CTC table, select Actions > Import to open the Import wizard.
- b. In the Data Source tab, select Local hard drive.
- c. In the File Name tab, choose and upload the file.
- d. In the Data Format tab, select Excel spreadsheet.
- e. In the Table tab...
 - i. Select CTC for the Sheet
 - ii. For Table, select Contract Type Clauses
 - iii. Choose to Skip first row
 - iv. Do not create missing choice values.
 - v. Ignore "failed to convert field" errors.
 - vi. In Workflow/Rules select Execute workflow actions and/or guards, and Run rules after import.
- f. In the Records tab, leave the default settings.

g. In the Fields tab, see the following screenshot.

Id	Sample Data	Column
1	Related Contract Type ID	Related Contract Type ID
2	Related Clause ID	Related Clause ID
3	DO NOT IMPORT (Related Clause Title)	*Not Imported*
4	Usage	Usage
5	Determination Method	Determination Method
6	Sequence	Sequence

h. Click Finish to perform the import.

4. Update the Condition fields in the new records.

When the import finishes, all CTC records are created in the table. For records with Usage=Conditional and Determination=System Defines, we need to update the Condition field in the CTC records.

- a. Edit the target record in the CTC table. In the Condition field, add the relevant clauses. For more help on possible values for this field, the Formula column of the All Conditional (System) Clause tab of the spreadsheet contains examples of possible values for each clause type. Locate the associated clauses and copy the value into the Condition field.

Contract Type Clause

Save Cancel Details History

ID: 180 Status: Active

*Related Contract Type: Concessions - Test Sequence: 3.2

Contract Type Clause Title: Concessions - Test-W/MBE - Version 1 - No Goal

Contract Type and Sequence: Concessions - Test - 3.2

Related Clause Title: W/MBE - Version 1 - No Goal

Usage: ☒ Required ☒ Conditional Determination Method: ☒ System Defines ☐ User Picks

Condition: ({\$latest_solicitation_id.minority_business=="W/MBE"&&\$latest_solicitation_id.minority_business_goal_applicable}) [Formula Help](#)

b. Save the records.

5. Refresh the Contract Type records.

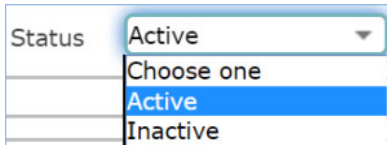
Note: This step is critical for the system to run.

- a. In the Contract Type table, open each record, select the Clauses tab, and click Refresh Contract Type Clauses.

Refresh Contract Type Clauses

Remove Clauses

1. To remove a CL record, simply change the Status value to Inactive. When this happens all associated CTC records with Auto-update=Yes will also be deactivated.



A screenshot of a web application interface showing a dropdown menu for the 'Status' field. The dropdown is open, displaying three options: 'Active' (selected), 'Choose one', and 'Inactive'. The 'Status' label is to the left of the dropdown.

2. After changing the CL status to Inactive, go into every contract type record and click the Refresh Contract Type Clauses button in the Clause tab. The system will then exclude any deactivated CTC when future blank contracts are generated.

Reactivate Clauses

To reactivate an inactive clause, change the Status value to Active. Note that associated inactive CTCs will not be reactivated automatically.

Remove CTC Records

To remove a CTC record, change the Status to Inactive. The CTC will not be included in future blank contracts, but nothing else will be affected.

Specification - Set Up and Update

This section contains implementation steps for common tasks relating to setting up and updating items, with the actions or rules that happen in each step and the purpose for each.

Note: the Sequence column should be viewed as a numbered list.

Entry Point: Modify the Clause field in CL record and click Save.

Sequence	Table	Action/Rule	Purpose
1	CL		

		Edit: When Clause Changes	
1.1	CL	L: Auto Update Linked CTC with Auto Update on	Push the newest clauses into the related CTC records with auto-updates on. Goes to CTC records using the Related Clause link in CTC, filter to those that are active and auto-update = Yes.
1.1.1	CTC	U: Refresh Clause Library Link Set	Re-import the Related Clause link set to update the clause
1.1.2	CTC	U: Pulls Contract Type Clause from Clause Library without Numbering	Copy the updated clause into the Contract Type Clause text field.

Entry Point: Modify the Status field in CL record and click Save.

Sequence	Table	Action/Rule	Purpose
1	CL	Edit: Status Changed	
1.1	CL	If status changed from Active to Inactive	
1.1.1	CL	Warn user that CTC will be deactivated.	Provide users a pass-through validation. User need to modify the auto-update option in related CTC records. All CTC with auto-updates on will also be deactivated automatically. Also remind user to go to the contract type records and click on the Refresh CTC button after all changes have been made in CL table.
1.1.2	CL	Set all linked CTC with auto-update on to be inactive	Goes to CTC records and deactivate all linked CTC with auto-update On.
1.2	CL	If status changed from Inactive to Active	
1.2.1	CL	Remind User to Reactivate CTC	Instead of doing reactivation automatically, we only provide a pass-through validation to inform user to do that manually. Check the reactivation of CTC records workflow below.

Entry Point: Edit the Clause field or Change Related CL and click Save.

Sequence	Table	Action/Rule	Purpose
1	CTC	Edit: If Clause or Related CL Changed	
1.1	CTC	If Clause Changed	
1.1.1	CTC	Warn User about Auto-Updating	Remind the user to toggle the Auto-Update setting if necessary.

1.2	CTC	If Related CL changed	
1.2.1	CTC	Pulls Contract Type Clause from Clause Library without Numbering	Refresh the Contract Type Clause field.

Entry Point: Change Status or Sequence field and click Save.

Sequence	Table	Action/Rule	Purpose
1	CTC	Edit: If Sequence or Status Change	
1.1	CTC	U: Set Sequence Int field value	
1.2	CTC	U: Trigger Related Contract Type to refresh the sequence	
1.2.1	Contract Type	Edit: Refresh CTC Sequence – L: Refresh Next CTC ID in all associated CTC	Goes down to all related CTC records and update the Next CTC ID in them.
1.2.1.1	CTC	U: Set Next Contract Type Clause	This action set the Next CTC link set within the current record to CTC with the same contract type and next higher sequence.
1.2.2	Contract Type	U: Set Trigger CTC refreshment to No.	Reset the control parameter in Contract Type.

Entry Point: Create a new CTC record and click Save.

Sequence	Table	Action/Rule	Purpose
1	CTC	Create: All Create Action (Web/API)	
1.1	CTC	U: Set Record Created to Yes	The CTC Clause is only visible after the record is created. So to avoid user copying the clauses manually when creating the record.
1.2	CTC	U: Set Sequence Int field value	Sequence Int= Sequence*1000. Used in save search comparison.
1.3	CTC	U: Pulls Contract Type Clause from Clause Library without numbering.	Copying the CL Clause into CTC Clause.
2	CTC	Create: All Create Action Web Only: Trigger CT Sequence refreshment	This function only runs if users are creating CTC manually. Importing CTC records skip this function because it wastes a lot of resources if we run it during import.
2.1	CTC		It triggers the refreshment of all CTC related to the same contract type to refresh the Next CTC ID.

		U: Trigger Related Contract Type to refresh the sequence	
2.1.1	Contract Type	Edit: Refresh CTC Sequence – L: Refresh Next CTC ID in all associated CTC	Goes down to all related CTC records and update the Next CTC ID in them.
2.1.1.1	CTC	U: Set Next Contract Type Clause	This action set the Next CTC link set within the current record to CTC with same contract type and next higher sequence.

Entry Point: import a list of CTC records from Excel.

Sequence	Table	Action/Rule	Purpose
1	CTC	Create: All Create Action (Web/API)	
1.1	CTC	U: Set Record Created to Yes	The CTC Clause is only visible after the record is created. So to avoid user copying the clauses manually when creating the record.
1.2	CTC	U: Set Sequence Int field value	Sequence Int= Sequence*1000. Used in save search comparison.
1.3	CTC	U: Pulls Contract Type Clause from Clause Library without numbering.	Copying the CL Clause into CTC Clause.
2	Contract Type	After the import finishes, user needs to go to the contract type record and click the Refresh Contract Type Clauses button	User initiate the refreshment instead of system triggers it every time a new CTC is imported.
2.1	Contract Type	U: Set Trigger CTC Refreshment to Yes	This will trigger the edit rule of refreshing the CTC's Next CTC ID link set.
2.1.1	Contract Type	Edit: Refresh CTC Sequence – L: Refresh Next CTC ID in all associated CTC	Goes down to all related CTC records and update the Next CTC ID in them.
2.1.1.1	CTC	U: Set Next Contract Type Clause	This action set the Next CTC link set within the current record to CTC with same contract type and next higher sequence.

Entry Point: User clicks the Generate Blank Contract File button in SOL.

Sequence	Table	Action/Rule	Purpose
1	Solicitation	Clause Library – Trigger first Clause	Solicitation > Contract Type/First CTC > CTC record: Set Latest Solicitation ID and Trigger Generation to Yes, and Current Article Number to 1.

1.1	CTC	Edit Generation of the Blank Contract in Solicitation	Edit rule that handles the generation of blank contract
1.1.1	CTC	Set Trigger Blank file to No	
1.1.2	CTC	If CTC needs to be generated	Checks whether the CTC needs to be included in the Solicitation blank contract based on Usage + Determination Method + Condition + Clause to Include.
1.1.2.1	CTC	Set Running CTC	Copy CTC Clause into Running Clause. Replacing the "_num" with \$current_article_number. The field value for first CTC is set to 1 within step 1.
1.1.2.2	CTC	Push Running CTC Clause into Latest Solicitation	Append the updated running CTC record into the solicitation Blank File Content field.
1.1.2.3	CTC	Set Next CTC Article Number to Plus 1	Add 1 to the current article number, and push that number to Next CTC's current article number.
1.1.3	CTC	If CTC DOES NOT need to be generated...	
1.1.3.1	CTC	Set Next CTC Article Number to Same	Copy current article number into Next CTC's current article number because the current CTC is not generated
1.1.4	CTC	If Next Contract Type Clause ID ! = Null	This means the current CTC is NOT the last clause. So we need to trigger the generation of the next CTC
1.1.4.1	CTC	Push Latest Solic ID to Next CTC and Set Trigger to Yes	This will trigger the next CTC's edit rule to run, going back to step 1.1 in the next CTC record. Then we go into a loop: Previous CTC records triggers the next CTC, until we reach the end (Step 1.1.5).
1.1.5	CTC	If Next Contract Type Clause ID= Null	This is the last clause... So we can tell the solicitation to generate the file because we no longer have any additional clauses.
1.1.5.1	CTC	Set Trigger Generation of Contract File to Yes in Solicitation	Goes up to Solicitation using the Latest Solicitation ID link, and set the trigger generation of contract file to Yes in Solic.
1.1.5.1.1	Solic	Edit: Generation of the Blank Contract File	This rule runs because the field set by 1.1.5.1

1.1.5.1.1.1	Solic	A: Generate Blank Contract	Put everything in the File Content field into a blank print template
1.1.5.1.1.2	Solic	Set Trigger Blank Contract to No and blank out file content	
1.1.5.1.1.3	Solic	B: Notify User the Blank Contract is Ready	User will get a browser pop-up letting them know the blank contract is generated.

Table and Field Specification

This section contains a description of the tables, fields and rules that were used in this design.

Note: If a rule level is LV1 - Major Function, then it is critical for the functionality. If a rule is LV2 - Assisting Function, then the rule is not critical for the functionality, but improves the user experience.

Clause Library

The Clause Library table is the major background table holding all the clauses for the customer.

Fields

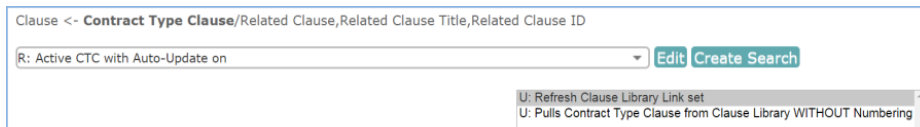
Field Label	Data Type	Notes
Clause	Text Field	HTML Only. Max size 100,000. Display 5 rows, 120 character wide.
Status	Choice: Active, Inactive	Default to Active.
Clause Title	Short Text	Max 255 characters. Title of the clause. (Used for future references)
Button: Cleanse HTML	Action Button	Running the Cleanse HTML function that converts text copied from Microsoft Word into HTML tagged text. This is needed because Word adds a lot of custom tags for formatting which need to be cleaned up for HTML. The script is attached here .

Rules and Actions

The rules and actions are listed below:

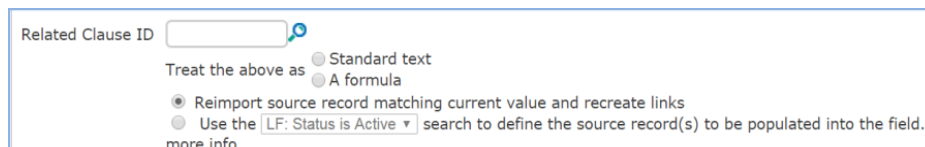
Edit: When Clause changes

- Description - When clause changes, all associated CTC with auto-update on will get updated.
- Rule level - LV1
- Condition - Edit, Web/API (if they are importing)
- Search - Clause changed
- Priority - 1
- Action - L: Auto-updated Linked CTC with Auto Update On.
 - Goes down to CTC, filter to Active + Auto-Update On

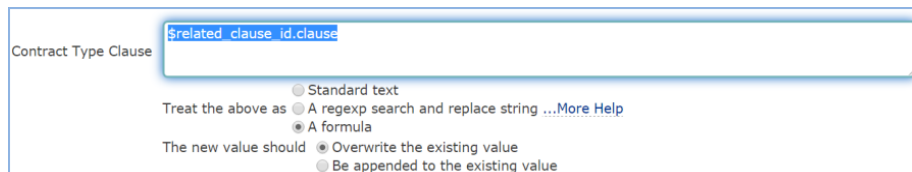


- Run:

- U: Refresh Clause Library linked set



- U: Pull CTC Clause from Clause Library without numbering



Edit: Status changed to Inactive

- Description - If status changed to Inactive, system needs to make sure there aren't any Active CTCs still linking to the Clause Library. If status changed to Active, system notifies user to reactivate the related CTC if necessary.
- Rule Level - LV2

- Condition - Edit, Web/API
- Search - R: Status changed during last user modification
- Action - I: Handle status change

```
if (Status changed from: 'Active' to: 'Inactive' last user's modification) {
    V: Warn User that CTC will be deactivated
    L: Set all linked CTC with auto-update on to be inactive
} else if (Status changed from: 'Inactive' to: 'Active' last user's modification) {
    V: Remind User to Reactivate CTC
}
```

Contract Type

The Contract Type table creates a list of contract types which each contain a selection of available clauses from the Clause Library.

Fields

Field Label	Data Type	Notes
Contract Type Name	Short Text	Name of the Contract Type
Related Contract Type Clauses	Related Table: Contract Type Clauses.	All CTC records related
First Contract Type Clauses ID, Title	Link to Contract Type Clauses	The Sequence 1 contract clauses. Used to initiate the contract generation. Set by default value, update whenever the underneath condition changed. This field is refreshed every time changes have been made to any of the CTC records.
Trigger refreshment of Clause	Choice: Y /N	Default to No. When set to Yes, it triggers the edit rule that goes down to all associated Contract Type Clauses (CTC) and refreshes the Next CTC ID in those records
Button: Refresh Contract Type Clause	Action Button	Set the Trigger refreshment of Clause to Yes. Then system will refresh the First CTC link in the CT record, as well as updating all Next CTC in CTC records. This button is used after importing CTC records from Excel. Used as the Manual Trigger refresh.

Rules and Actions

The Contract Type table contains a rule and several actions which are described here.

Button - First Contract Type Clause ID

The default setting for the first Contract Type Clause ID button takes the Set First Contract Type Clause from CTC record action.

☒ run the Set First Contract Type Clause from CTC record action when the Trigger Refreshment of Clause field is first set or edited

Create Action

U: Set First Contract Type Clause from CTC record

Rule Level: LV1

☒ Use the MT: First Contract Type Clause in Contract Type: CT matches and Active

more info

Edit Create New Search

This search has the following filters and sorting order:

Related Contract Type ID

equals, =

Variable

\$contract_type.id

and

Status

equals, =

Value

Active

now

☒ First Sort

Sequence

Ascending

Rule - Edit: Refresh CTC Sequence

- Description - LV1. Refreshes all related active CTC records, sets the Next CTC record ID in all CTC records. The Trigger Refreshment of Clause field is changed to Yes by either the edit rule in CTC table, or by pressing the Refresh Contract Type Clause button.
- Condition - Edit, Web/API
- Search - Trigger refreshment of Clause changed from No to Yes.
- Priority - 1
- Actions

- L: Refresh Next CTC ID in all associated CTC

Contract Type <- **Contract Type Clause**/Related Contract Type,Related Contract Type ID

LF: Active Records Edit Create Search

U: Set Next Contract Type Clause

- U: Set Next Contract Type Clause

Next Contract Type Clause ID

Treat the above as ☐ Standard text ☐ A formula

☐ Reimport source record matching current value and recreate links

☒ Use the MT: Next Contract Type Clause in Contract Type Clause: Same contract type, active, higher sequence more info

Edit Create New Search

☐ Choose the first record returned by the search in sort order

Sequence Int is more than, > Variable \$contract_type_clause.seq ☒ now

and Related Contract Type ID equals, = Variable \$contract_type_clause.rela ☒ now

and Status equals, = Value Active ☒ now

☒ First Sort

Sequence Ascending

- U: Set Trigger CTC Refresh back to No
 - For 1/11 - Generate Template: If first is empty, or CTC status is inactive > broadcast, abandon.

Contract Type Clauses

For a specific contract type, the sequence must be unique, so if you have two different clauses with the same sequence but different versions, you also need to make sure the sequence is different. It can be something like 3.1 and 3.2. When generating the contract, the system works like the following:

- Sequence 2 (required) – generated
- Sequence 3.1 (conditional) – check – condition not met – goes to the next directly without pushing data into SOL
- Sequence 3.2 (conditional) – check – condition met – push clause into SOL – move to sequence 4.

In addition, there are three types of CTC:

1. Required CTC
2. Conditional CTC determined by metadata in the Solicitation record
3. Conditional CTC select by user manually.

The difference between 2 and 3 is that although both have Usage=Conditional, 2 has Determination=System Defines; while 3 has Determination=User Picks.

For CTC type 3, in the Solicitation record there is a field called Conditional Clause to Include. This field holds all the CTCs that are related to the same Contract Type with Conditional and User Picks, and the user picks the ones they would like to include.

When generating the CTC, the system checks whether the Condition meets, or if the conditional clause is included in the field in Solicitation. If either of them is true, it generates the clause.

Fields

Field Label	Data Type	Notes
Related Contract Type ID, Related Contract Type	Link to Selected Fields	Link to Contract Type table. Required.
Related Clause ID, Related Clause Title, Related Clause	Link to Selected Fields	Link to Clause Library. Filter to Status=Active records only.
Contract Type and Sequence	Compound	Contract Type + Sequence. Unique. This field is critical because we need to make sure the sequence is unique for certain contract type. For System use mainly.
Contract Type Clause Title	Compound	<p>Contract Type + Clause Title. This field should be unique. We are not replacing the Contract Type Clause Sequence field because the other field is used for admins to make sure the Contract Type + Sequence combination is unique. On the other hand, this field is more user friendly so it is included in the solicitation record "user picking conditional clauses" section.</p> <p>Basically this field is for the user to user, and the other field is for administrative maintainance.</p>
Sequence	Floating Number	Clause sequence. Do not round.
Condition	Variable Formula	<p>This is the field marking whether the clause needs to be included based on the contract metadata.</p> <p>For example, clause W/MBE – Version 1 – No Goal only needs to be generated if Minority Business = WMBE and MBAApplicable = No Goal.</p> <p>Visible only if Determination Method=System Defines. Determination is only visible if usage is conditional.</p>

Status	Choice: Active /Inactive	Marking whether a record is active or not. Default to Active.
Latest Solicitation ID, Conditional Clauses to Include	Link to Solicitation	Clause is generated for the Latest Solic record.
Next Contract Type Clause ID, Clause Sequence, Title	Link to Contract Type Clauses	If empty, then system knows this is the last clause for the contract type. Filter to CTC for the same contract type with Active status
Record Created	Choice: Y /N	Default to No. Set to Yes once the record is created. This field is used to hide the Contract Type Clause field before the record first gets created, so the system can pull the clause from CL table, and prevent users from entering the Clause manually before the record is created.
Should System do Auto-Update of the Clauses	Choice: Y /N	If this field is set to Yes, the system auto-update the CTC record whenever underlying CL record changed (CL Status and CL Clause). Once there are changes made to the CTC clause, this field is set to No by the system automatically.
Contract Type Clause	Text	Max 100000. This field holds the clause for the specific contract type, without numbering updated. HTML only. Visible only if Record Created = Yes.
Usage	Choice	Required, Conditional. This field determines whether there is an alternative contract type clause that can be used.
Trigger Generation of Blank Contract	Choice: Y /N	Default to No. If set to Yes, then the running clause of this CTC will be pushed to the latest Solicitation.
Sequence Int	Int	<p>This field is built because the floating point Sequence is not working within the Search filters to populate the Next CTC ID. Hidden field, used by the system. set by Create/Edit Rule. Value equals Sequence * 1000.</p> <p>Note: for systems using later releases, the floating point comparison in search filters has been fixed, so this field is no longer needed.</p>
Determination Method	Choice: System Defines, User Picks	Visible only if Usage = Conditional. Default to system defines. If set to User picks, then the clause is selectable in Solicitation, and user needs to include the clause manually.
	Action Button	Action Bar rule. This button manually pulls the clause from the CL record.

Button: Pull Clause from Related Clause Library		
Current Article Number	Int	Because we have optional clauses, the Article Number showing in the blank contract may be different from the Sequence. This field is set dynamically every time the blank contract is generated, so all the included clauses will have a sequential article number.
Running Contract Type Clause	Text	This field holds the actual clause that will be pushed to the Solicitation record. This field is set every time blank contract is generated. The field holds the same value as the CTC clause except the <code>_num</code> is replaced by the Current Article Number.

Rules and Actions

Below are the rules and actions included in the Contract Type Clauses table:

Create: All Create Actions Web/API

- Description - LV1. This rule sets several default values as well as the Contract Type Clause - not the running clause. It just copies the CL Clause into the CTC Clause, without numbering updates, so the `_num` remains.
- Condition - Create, Web/API
- Search - None
- Priority - 1
- Actions:
 - U: Set Record Created to Yes
 - U: Set Sequence Int field value
Sequence Int=Sequence*1000
 - U: Pulls CTC from Clause Library without numbering

Contract Type Clause

\$related_clause_id.clause

☐ Standard text
☐ Treat the above as A regex search and replace string ...[More Help](#)
☒ A formula

Create: All Create Actions Web Only - Trigger CT Sequence refresh

- Description - LV1. This rule goes to the related CT and triggers a refresh of the next CTC in all associated CTC records. This rule is web only, so that if the user is importing CTCs from Excel the refresh is not triggered every time, which takes a longer time to import.
- Condition - Create, Web
- Search - None
- Priority - 2
- Actions:
 - U: Set Related Contract Type's trigger refresh of Clauses to Yes
Contract Type Clause/Related Contract Type ID > Contract Type. Not a CTC local field, but a link to the CT record.

Trigger Refreshment of Clauses

☒ Yes ☐ No

Treat the above as

☒ Standard text ☐ A formula

Edit: If Sequence changed, or status changed

- Description - LV1. Handle the CTC record if the CTC parameter changes.
- Condition - Edit, Web
- Search - Status or Sequence changed during last modification
- Priority - 3
- Actions:
 - U: Set Sequence Int field value
 - Trigger Related Contract Type to refresh the sequence
Important: the user cannot change the related Contract Type after the CTC record is created, because there is not a field and rule to trigger a refresh of the previous contract type.

Edit: If Clause or Related CL record changed

- Description - LV2. Handle the action if the CTC clause or the Related CL change. Web only because of the validation rule. Also prevents the Warn User about Auto-Updating action from running if the user changes the Related Clause.

- Condition - Edit, Web
- Search - CTC Clause or Related CL changes
- Priority - 4
- Actions:
 - I: Handle Clause changes

```

if (Contract Type Clause changed from: any value to: any value last user's modification) {
  V: Warn User about Auto-Updating
} else if (Related Clause ID changed from: any value to: any value last user's modification) {
  U: Pulls Contract Type Clause from Clause Library WITHOUT Numbering
}

```

Edit: Generation of the Blank Contract in Solicitation

- Description - LV1. Major action used to generate the blank contract
- Condition - Edit, Web/API

Important: this rule runs in background, with looping protection disable. Check the two boxes under the Apply this rule section on the Rule Type tab.
- Search - Trigger Generation of Blank Contract change from No to Yes during Last Modification
- Priority - 5
- Actions:
 - I: Actions Generating Blank Contract

```

Set Trigger Blank File to No
if (Usage='Required' or Usage='Conditional' and Determination Method='System Defines' and Condition='TRUE' or Usage='Conditional' and Determination Method='User Picks' and Contract Type Clause Title<<Latest Solicitation Conditional Clause to Include) {
  Set Running CTC
  Push Running CTC Clause into Latest Solicitation
  NEW: Set Next CTC Article Number to Plus 1
} else {
  if (Next Contract Type Clause ID!=NULL) {
    NEW: Set Next CTC Article Number to Same
  }
}
if (Next Contract Type Clause ID!=NULL) {
  Push Latest Solicitation ID to Next CTC and Set Trigger to Yes in Next CTC
} else {
  Set Trigger Generation of Contract File to Yes in Solicitation
}

```

	Usage	equals, =	Value	Required	<input checked="" type="checkbox"/> now	
or	(Usage	equals, =	Value	Conditional
					<input checked="" type="checkbox"/> now	
and			Determination Method	equals, =	Value	System Defines
					<input checked="" type="checkbox"/> now	
and			Condition	equals, =	Value	TRUE
					<input checked="" type="checkbox"/> now	
or	(Usage	equals, =	Value	Conditional
					<input checked="" type="checkbox"/> now	
and			Determination Method	equals, =	Value	User Picks
					<input checked="" type="checkbox"/> now	
and			Contract Type Clause Title	equals, =	Variable	\$latest_solicitation_conditio
					<input checked="" type="checkbox"/> now	

- U: Set Running CTC - this rule sets the running CTC field that will be pushed to Solicitation. It replaces the `_num` with the Current Article Number value.

Running Contract Type Clause

`replace($contract_type_clause,"_num",concat($current_article_number))`

Treat the above as ☐ Standard text ☒ A regexp search and replace string [...More Help](#)

The new value should ☒ Overwrite the existing value

- L: Push Running CTC Clause into Latest Solicitation. This action pushes the updated Running CTC value with updated numbering into the latest Solicitation ID field.

Contract Type Clause/Latest Solicitation ID -> **Solicitation**

---None--- [Edit](#) [Create Search](#)

U: Pull Running CTC Clause into File Content -

- L: New: Set Next CTC Article Number to Plus1. Because the current CTC is included in the blank contract, the next article number should be increased by 1.

Contract Type Clause/Next Contract Type Clause ID -> **Contract Type Clause**

---None--- [Edit](#) [Create Search](#)

U: Set Article Number to Article Number plus 1 -

- U: Set Article Number to Article Number Plus 1

Current Article Number

`$contract_type_clause.current_article_number+1`

Treat the above as ☐ Standard text ☒ A formula

- L: New: Set Next CTC Article Number to Same. Because the current CTC is skipped, the article number should remain the same for the next CTC.

Contract Type Clause/Next Contract Type Clause ID -> **Contract Type Clause**

---None--- [Edit](#) [Create Search](#)

U: Set Article Number to Article Number to Same -

- L: Push Latest Solicitation ID to Next CTC and Set Trigger to Yes in Next CTC. If there is a next CTC, trigger the blank contract generation of the next CTC record.

Contract Type Clause/Next Contract Type Clause ID -> **Contract Type Clause**

---None--- [Edit](#) [Create Search](#)

U: Pull Latest Solicitation ID from previous CTC and set Trigger to Yes -

- U: Set Trigger Generation of Contract File to Yes in Solicitation. If there is no next CTC it means the current CTC is the last clause that needs to be checked. As a result, we can notify the Solicitation record to generate the file from the file content.

Contract Type Clause/Latest Solicitation ID > Solicitation - not local CTC field

Trigger Generation of Contract File

Treat the above as ☒ Standard text ☐ A formula

Solicitation

The Solicitation table is the main processing table where contracts are generated from the the clauses.

Fields

Field Label	Data Type	Notes
Contract Type ID, Contract Type Name	Link to selected fields: Contract Type	Link to Contract Type table. User selects the contract type.
Blank Contract File	File with versioning	File field. This is the field holding the generated template based on the clause library. DO NOT allow multiple files.
Blank Contract File Content	Text Field	Field to hold the texts for the template. Used to hold all the clauses temporarily. Max 1,000,000 characters.
Generate Contract Template	Action Button	<p>Visible only if Ready for Generation=Yes. This is the button that goes through all related Contract Clauses and appends the clauses in the sequence to generate the template.</p> <p>Note: In order to prevent the initial clause from being overwritten, this button must be set up to save the record before executing the action, and then close the record without saving after executing.</p> <div><p>Before executing actions:</p><p><input type="radio"/> Do nothing</p><p><input checked="" type="radio"/> Save record</p><p>then</p><p><input checked="" type="checkbox"/> Execute Actions</p><div><input type="text" value="Clause Library - Trigger first Clause"/> <input type="button" value="Add Action"/></div></div> <div><p>After executing actions:</p><p><input type="radio"/> Do nothing</p><p><input checked="" type="radio"/> Close record without saving</p><p><input type="radio"/> Save record</p><p><input type="radio"/> Save and Open record for edit</p></div>
	Choice: Y/N	Default to No. Set to Yes by the last CTC. Then it triggers the print template function to put the blank contract file content into a blank template.

Trigger Generation of Contract File		
Conditional Clause to Include	Link to Single field with MVE	Link to CTC's Contract Type Clause Title field. Filter to CTC with Same Contract Type + Usage=Conditional + Determination=User Picks+ Active. User picks the CTC they want to include manually.

Rules and Actions

Below are the rules, actions and action buttons included in the Solicitation table:

Button: Generate Contract Template

- Description - LV1
 - If the user clicks on this button, the system will push the current Solicitation ID to the Solicitation > Contract Type > First CTC's latest Solicitation ID field. This initiates the blank file generation. The first CTC will push the file content to Solicitation.
 - Also, it checks whether the next CTC ID is set. If it is, the CTC will trigger the next CTC; if empty, meaning it is the latest CTC, the process stops.
 - The latest CTC will update the trigger in Solicitation that copies the file content into a blank print template.
 - If the first CTC in Contract Type is null, nothing happens.
- Action:
 - L: Clause Library - Trigger First Clause

Edit: Generation of the Blank Contract File

- Description - LV1. The Trigger Generation of Blank Contract field in the Solicitation record is set to Yes by the last CTC - by the very end of the process, when all clauses are in the File Content Field. When this field is set to Yes, it triggers this rule.
- Condition - Edit, Web/API
- Search - R: Trigger Blank File Field set from No to Yes during last modification

- Priority - 12

- Actions:

- A: Generate Blank Contract
- U: Set Trigger Generation of Contract File to No and Blank out file content
- B: Notify User the Blank Contract is ready