SAP Ariba

PUBLIC

Document Version: 2502 – 2025-02

Enhanced Contract Authoring

SAP Ariba Contracts

THE BEST RUN SAP

Content

Enhanced Contract Authoring

This guide is for SAP Ariba users and administrators who use SAP Ariba Contracts for authoring contracts, and describes how to use the enhanced contract authoring features of SAP Ariba Contracts with assembled and non-assembled contract documents.

SAP Ariba Contracts includes enhanced contract authoring features such as, drag-and-drop document imports, seamless editing, and simplified clause management.

This guide applies to:

- SAP Ariba Contracts
- SAP Ariba Strategic Sourcing Suite

Related Guides

Managing Projects, Teams, Documents, and Tasks

Contract authoring guide

Contract Administration Guide

Project Template Guide

Common Data Import and Administration Guide for SAP Ariba Strategic Sourcing Solutions and SAP Ariba Supplier Management Solutions

About Enhanced Contract Authoring

SAP Ariba Contracts supports three types of contract documents:

- Non-assembled documents
  Non-assembled documents can be created with any application (including Microsoft Word, Microsoft Excel, Microsoft PowerPoint, or a plain text editor). SAP Ariba Contracts handles each non-assembled document as a single component or a flat file with no internal structure or subcomponents.
- Assembled contract documents
  Assembled contract documents (also referred to as assembled documents) can be created with any OpenXML based editor. An assembled document can have multiple subcomponents, or sections and clauses. A clause is one or more paragraphs treated as a single unit. Clauses can be stored and retrieved from the SAP Ariba Contracts Clause Library. You can create sections in an assembled document to organize the clauses. SAP Ariba Contracts provides an outline view that shows the section structure and clauses and enables you to edit the section structure, and to move, add, replace, or delete clauses.
  When you load or create an assembled document, you select if it is a Main Agreement or a Contract Addendum. A template or a contract workspace can have only one Main Agreement documents, but can have multiple Contract Addendum documents.
  Using assembled documents in a contract workspace is not required. You can load all documents in a workspace as non-assembled documents. However, assembled documents enable you to use contract authoring features. Contract authoring features are a set of features that provide additional functions for working with assembled documents and are supported only with assembled documents.
- Assembled PDF documents
  Assembled PDF documents (also referred to as PDF documents) can be created from single or multiple documents in a contract workspace.

The enhanced contract authoring features are not supported for Ariba Contract Management Basic users. This guide describes how to use contract authoring features with assembled documents. For information about features that can be used with all document types, see Working with Projects and Documents.

Overview of Enhanced Contract Authoring Features

SAP Ariba supports the following contract authoring features:

- Clause Library
  SAP Ariba Contracts enables you to create and manage a Clause Library. The Clause Library is a tool for storing and managing paragraphs of text for contracts as contract clauses. The Clause Library enables your organization to standardize and regulate text used in contracts while enabling contract authors to select and exchange clauses as needed.
  The Clause Library enables users to:
  - Provide a single source for standardized contract text. Text can be written and approved once, then reused in multiple contracts. For example, you can create a clause with your company's legal disclaimer. All users can incorporate this disclaimer from the Clause Library when writing contracts.
  - Provide fallback or alternate versions for contract clauses that users can select according to their needs. For example, you can create one clause for legal disclaimers in the European Union and an alternative clause for legal disclaimers in the United States.

- Manage clause versions and automatically require approvals when text in clauses is changed.
- Determine the source of clauses in a contract document. The outline view indicates which clauses are inherited from the template document, replacement clauses from the Clause Library, edited, and added from external sources.
- Use conditions to create templates that automatically select clauses according to information such as contract region or commodity type.
- Find all templates and workspaces that use a specified clause. SAP Ariba Contracts provides a "Search Where Used" feature that enables you to find all workspaces and templates that use a given clause.
- Create reports about clause with information such as how frequently a clause is used.

> Note
>
> Your company's current contract clauses might have a clause/subclause structure. This structure is not supported by SAP Ariba Contracts.

- Document Properties
  Document properties automatically generate and update contract text from contract workspace fields. For example, you set up the supplier's or customer's name and address as document properties and automatically import this information from the contract workspace and include it in the text of the Main Agreement.
  You can also use document properties to change contract workspace field values by changing text fields in Main Agreements or Contract Addenda.
- Formatting to achieve a uniform style
  By formatting contract documents and templates using simple styles available in any OpenXML editor, you can enforce a uniform corporate look and feel for all contract documents. You can link paragraph styles to section titles and content in a Main Agreement or Contract Addendum. You can also apply automatic numbering to paragraph styles, which enables the system to correctly number clauses regardless of where they are used in different contract documents.

Enabling Enhanced Contract Authoring

This feature is disabled by default. To enable this feature, please have your Designated Support Contact log a service request. A SAP Ariba Customer Support representative will follow up to complete the request. If you enable this feature, you will be redirected to the new contract authoring authoring workflow with drag-and-drop document imports, seamless editing, and simplified clause management.

Related Information

Contract Authoring Requirements and Recommendations

Enhanced Workflow, Use-Case, and Recommendations

Enhanced Contract Authoring Features

SAP Ariba provides enhanced contract authoring with drag-and-drop document imports, seamless editing, and simplified clause management.

The enhanced contract authoring workflow provides a simplified, intuitive, platform-independent interface while eliminating many of the limitatons inherent in the previous workflow. The enhanced contract authoring features are not supported for Ariba Contract Management Basic users. Following are the highlights of this enhanced interface:

- Open-XML based, bookmark-free document formatting
  Previously, you needed to add bookmarks to your contract documents (main agreement and contract addendums) to identify clause boundaries. Now, you can format documents using any Open-XML based editor and use document styles to identify clause boundaries.
- Drag-and-drop document imports
  Previously, you needed to create a Desktop File Sync (DFS) folder on your system and periodically sync document updates from the local folder to the server. Now, you only need to drag-and-drop documents directly into the application, and your dropped files are automatically available on the server.
- Simple, in-application redlining and editing
  Previously, comparing and merging contents of main agreements after internal or external edits invoked an external application with a three-way merge window. Now, the three-way merge window is replaced with a simple in-application redlining view that allows you to view and document changes quickly.
- Server-side document assembling and generation with no Desktop File Sync (DFS) or dependency on the Internet Explorer browser
  The use of DFS in the previous contract authoring workflow required ActiveX controls, which presented security vulnerabilities and required Internet Explorer. Since DFS and ActiveX are no longer used, you are free to use a variety of browsers.

Typical Contract Authoring Use Case

This section describes a typical contract authoring use case to illustrate how the enhanced contract authoring features make contract authoring intuitive and simple to use.

John has content for the main agreement of a procurement contract.

Situation: Quickly Upload, Update, and Review Contract Documents for a Contract

John wants to quickly upload the document to his contract workspace. After he uploads the document, he might decide to update the document to include information he had missed adding when he uploaded the document. He wants to re-upload the updated document and get it reviewed by his manager. John and his manager have used Microsoft Word but not its formatting features such as bookmarks.

Solution: Use SAP Ariba's Enhanced Contract Authoring

John uses SAP Ariba's enhanced contract authoring to quickly upload, edit, and review documents:

1. John structures the content of his main agreement by using heading styles to indicate sections and content in normal font to indicate clauses.
2. Using a browser (Chrome) on his system, John logs in to SAP Ariba's contract management workspace and creates a new template.
3. On the template screen, under **Actions** > **Template** > **Select Project type**, John selects a type of project, for example, `Contract workspace (procurement)`.
4. On the workspace UI, John selects **Actions** > **Create** > **Main Agreement**, and uploads the document he created.
5. John verifies that the **Outline View** shows that the clauses are generated correctly from his uploaded document.
6. John uses the **Actions** > **Add Clauses** or **Actions** > **Set up Document Properties** menu to set up initial document properties.
7. John needs to make changes to the template, so he updates his original document with the changes. In the drag-and-drop area in the top-left corner of the application, he drags and drops the updated document. He is immediately presented with the **Outline View** for the updated document. He verifies that the **Outline View** reflects his updates.

   > Note
   >
   > If a user publishes the template before making changes to the document, then the user gets an alert asking him whether he wants to review the changes.

8. John updates the document properties by clicking **Actions** > **Add Clauses** or **Actions** > **Set up Document Properties**.
9. John selects **Actions** > **Publish** to publish the template.
10. John creates a new contract workspace by going to **Create** > **Contract Workspace (Procurement)**. He sees the document he uploaded in the **Documents** section of the **Contract Workspace** view.
11. John once again reviews the **Outline View** to ensure that his document shows the clauses correctly.
12. John wants his manager to review the document, so he creates a review task by clicking the down arrow button next to the document and select **Create Task** > **Review**. He adds his manager as the reviewer and clicks Submit to send across the task for review.
13. John's manager logs in to SAP Ariba Contracts, opens **My Tasks**, and clicks **Review Task Details** to review the document. He downloads the document and makes changes to some sections and clauses.
14. John's manager goes to the review window, selects **Attach Revised Document**, and browses to and uploads the revised document.

15. John opens **Tasks** ▸ **Review for xxx.docx** ▸ and clicks **View Task Details**.
16. John clicks **Review Changes** next to the document.
17. The **Review Changes** window is a simple window that displays the number of changes and highlights the current change. John simply traverses the changes one by one by clicking **Previous** or **Next**, and clicks **Accept Changes** to accept the changes he agrees with.
18. After reviewing the changes, John can save the document as a new version or overwrite the exisitng version.
19. After John saves the changes and exits, he can revisit the document later and see that his accepted changes appear in the document after the review. He can also view all revisions in the **Version History** section and see changes between the versions.

Contract Authoring Workflow

Procedure

1. If you are working with an assembled contract document (Main Agreement or Contract Addendum) that is already loaded into a project template or workspace, open the **outline view** for the document. The outline view also enables you to move, replace, and delete sections and clauses using menu options and drag-and-drop operations. See Working with Assembled Contract Documents [page 20].
2. Plan your contract document structure. Determine if you will add to or use clauses in the Clause Library and which documents will be assembled documents. See Planning Contract Document Structures [page 11].
3. In any OpenXML editor, add your content using the available styles to demarcate clauses. Apply styles to the paragraphs in the assembled documents, as described in Assembled Contract Document Creation in the Enhanced Workflow [page 12].
4. Load the documents to a template or contract workspace as Main Agreement or Contract Addendum documents. See Loading Assembled Contract Documents [page 44].
5. (Optional) Set up the Clause Library. A designated user must populate the Clause Library with contract clauses for other users to incorporate into contract workspaces. See Adding and Modifying Clauses in the Clause Library [page 45].
6. (Optional) Add or replace clauses in the assembled documents with clauses from the Clause Library. See About Working with Assembled Contract Documents [page 20] and Replacing (Substituting) Clauses in Assembled Contract Documents [page 23].
7. (Optional) Set up document properties and add document property fields to the assembled documents. See Using Contract Fields as Document Properties [page 69].
8. Modify the document as needed and generate new versions as needed. See Working with Assembled Contract Documents [page 20].

Related Information

About Contract Authoring [page 6]

Contract Document Structures

To plan the structure for an assembled contract document (Main Agreement or Contract Addendum) in a template or workspace:

Determine if you will use the Clause Library and select text you want to include in the Clause Library. Clauses in the library are typically sections of text that are reused in multiple contracts.

Determine clause boundaries. Determine fallback and alternative clauses (clauses with the same or similar function, but with alternative text for differences such as contract location, contract term, or commodity type).

Determine which documents will be created or loaded as assembled documents (Main Agreement or Contract Addendum documents). Keep in mind the following points:

- A Main Agreement or Contract Addendum must be a document saved in the `docx` format.
- A template or project workspace can have only one Main Agreement document and can have multiple Contract Addendum documents.
- If you will not use any contract authoring features with a document, you can load the document as a non-assembled document.

Determine file boundaries. Keep in mind the following points:

- The maximum size file that you can upload is 20MB.
- The time needed to load a Microsoft Word contract document is proportional to the size of the document, so longer documents can take longer to load.
- If your company negotiates very large contracts, consider breaking the documents up into smaller documents to improve performance.

Enhanced Contract Authoring Workflow

Assembled Contract Document Creation in the Enhanced Workflow

Preparing assembled contract documents in the enhanced contract authoring workflow is simple and intuitive. Using simple styles available in any OpenXML editor, you can enforce a uniform corporate look and feel for all contract documents.

With enhanced contract authoring, you can use any OpenXML editor to quickly create an assembled document such as a main agreement or a contract addendum document. Creating an assembled document is as simple as setting up styles for the document. SAP Ariba recommends that you use Microsoft Word to create styles for your document. For details on creating styles in Microsoft Word, search for "Create a new style" in the Microsoft Help. For complete information on styles in Microsoft Word, consult the Microsoft documentation.

Any paragraph containing text that does not use a heading style (for example, a Microsoft Word style such as `Heading 1` or `Heading 2`) automatically becomes a clause. A paragraph that contains a heading style is a section or sub-section heading depending on its outline level depth. For example, the `Heading 1` heading style in a document created using Microsoft Word becomes a section title. Similarly, the `Heading 2` heading style in a document creating using Microsoft Word becomes a sub-section title.

To structure a document, you demarcate sections and content using styles. To combine more than one paragraph into a single clause, you can move the clauses around as described in Clause Library [page 45].

You can link styles in your OpenXML editor to section titles and content in a Main Agreement or Contract Addendum document. You do this in one of two ways:

- Define a label for the title and content for each section and link it to a pre-existing style in Microsoft Word (such as `Heading 1`) or another OpenXML editor. For example, to specify a sub-section title, you can map your label to the `Heading 2` style in Microsoft Word.
- Define a label for the title and content for each section and select an outline level that indicates whether the content applies to a section, sub-section, or clause, and its depth in the navigation tree. To specify the outline level, select **Format** > **Paragraph** > **Modify** > **Outline Level**. For example, for a first-level sub-section, you create a label and manually modify the style by setting **Modify** > **Format** > **Paragraph** > **Outline Level** to `Level 2`.

Essentially, you choose your labels based on the section levels, which you can see in the document's **Navigation View** in your editor or the document's **Outline View** after it is uploaded to your contract workspace. For example, a contract document with two section levels within the global section can have the following labels:

- Title Level 1 (titles of first-level sections)
- Content Level 1 (content within first-level sections)
- Title Level 2 (titles of all second-level sections)
- Content Level 2 (content within second-level sections)

A document with additional section levels would have a Title Level *n* and Content Level *n* label (where *n* is the level number) for each level. You can map only one style only per section title or content level. You cannot map section titles or content at the same level with different styles.

The paragraph style in your OpenXML editor can specify characteristics such as:

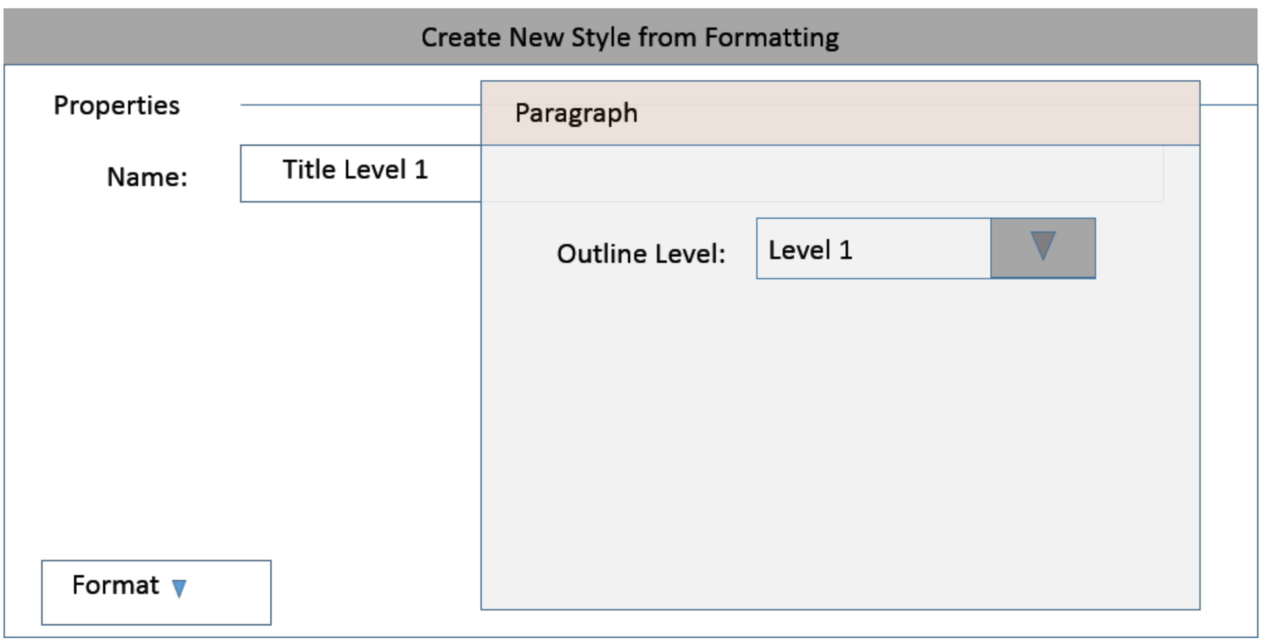
Numbering

Spacing

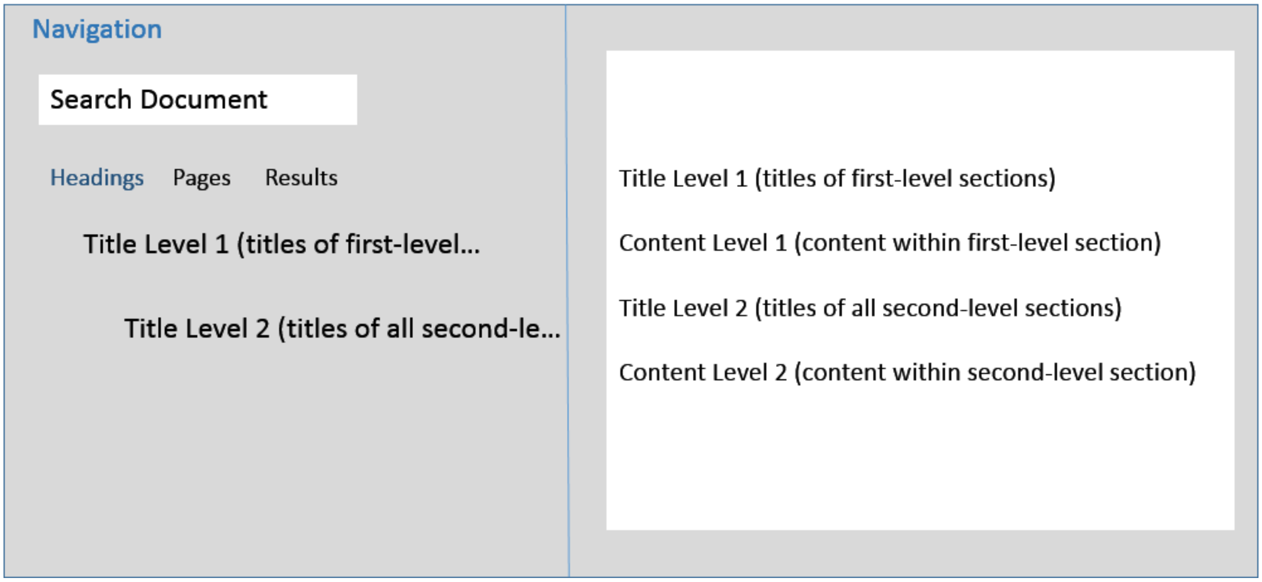
Fonts

In OpenXML editors that support multilevel lists (such as Microsoft Word 2010 and Microsoft Word 2013), you can create a template document with a multilevel list and appropriate styles to automatically apply styles and numbering to all documents that inherit the template. Example: Using Styles to Number Clauses [page 15] describes how a multilevel list can be used to automatically number clauses in your assembled documents.

The following view illustrates how the `Title Level 1` label is mapped to `Outline Level 1` in Microsoft Word by selecting **Create a Style**, naming the style with the label name, selecting **Modify** > **Format** > **Paragraph**, and setting **Outline Level** to `Level 1`.



Similarly, the `Title Level 2` label can be mapped to `Outline Level 2` in Microsoft Word. The following is a view of how a document structured in this way appears in the **Navigation View** in Microsoft Word.



The **Navigation View** reflects how the clauses and sections will appear in the **Outline View** of the document when you upload the document to your contract workspace.

> ⓘ Note
>
> The only exception is that sections and sub-sections that have `Ignore` (not case-sensitive) pre-pended to the label are shown in the **Navigation View** but not in the **Outline View**.

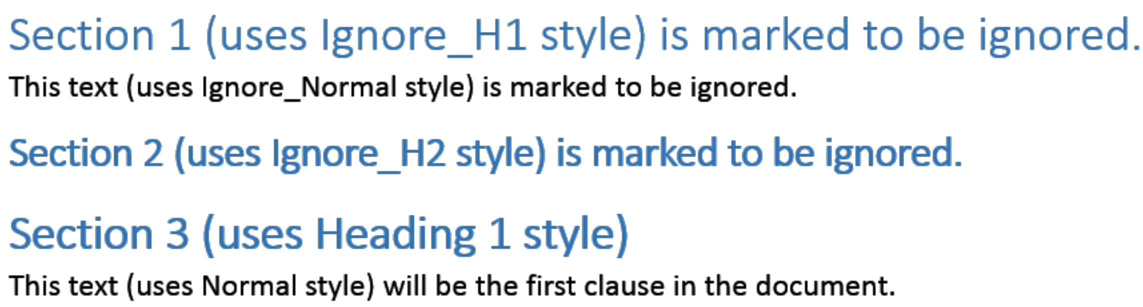
SAP Ariba therefore recommends that you use the **Navigation View** as a guidepost for your contract document structure even before you upload the document to your contract workspace.

If you want the application to combine more than one paragraph into a clause (by default, each paragraph is treated as a separate clause), then you must create a new paragraph-level style and append `combine` to the start of the style name. In this way, the paragraph does not get converted into a clause on its own, but is appended to the immediately previous clause or section, and multiple paragraphs can be combined into a single clause.

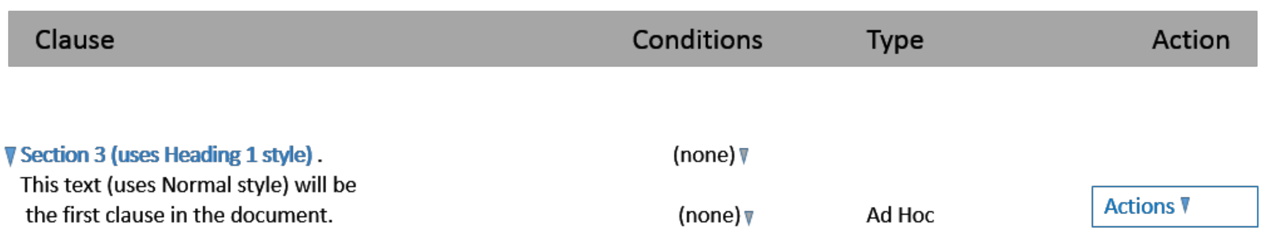
If you want the application to ignore certain sections of the document, then you must create a new style and append `Ignore` to the start of the style name. To create a style for a section that must be ignored, base its **Style based on** element on the original style (say `Heading 1` for a section heading that must be ignored). This allows any formatting changes you make to the original style to propagate to paragraphs that use both the original styles and the ignored styles.

You can apply the `Ignore` style to any content in your document that you want the application to exclude from the **Outline View**. Images are excluded by default from the **Outline View**.

The following is a view of a document created using Microsoft Word that uses the `Ignore` style to exclude content from the **Outline View**.



The styles for Section 1, the text within Section 1, and Section 2, are modified using Microsoft Word to append `Ignore_` to them. The modified styles are `Ignore_H1`, `Ignore_Normal`, and `Ignore_H2`. These sections and content are not shown in the Outline View. Section 3 uses the `Heading 1` style in Microsoft Word, and the text after it uses the `Normal` style in Microsoft Word. This section and content is shown in the **Outline View**. The following is a view of the **Outline View** for the document structured using the above styles.



In addition to using styles to define fonts and styles, you can use styles to automate the numbering of clauses and sections.

Keep the following rules in mind while defining styles and modeling your business's visual style of clauses and sections in contract documents:

- Define any specific character formatting (for example, bold or italics) in the clause that is located the Clause Library. This helps ensure consistent clause appearances in generated contract documents.
- Define paragraph formatting only at the contract document level by applying style mappings to the entire document.
- Update any cross references manually in your assembled contract documents.

The new way of styling documents makes it easy to debug a document that was accidently modified for structure, as there is no use of or dependency on bookmarks or accidental inclusion of special characters or carriage returns. Structuring assembled contract documents in the enhanced workflow is thus simple and efficient.

Example: Using Styles to Number Clauses

To use style mapping with automatic paragraph numbering, you must define list levels in an OpenXML editor multilevel list or in Microsoft. If you are using a version of Microsoft Word that does not support multilevel lists, you must define numbering properties for the Microsoft Word paragraph styles. Do not use directly-applied numbering for individual paragraphs.

In Microsoft Word versions that support multilevel lists (such as Microsoft Word 2010 and Microsoft Word 2013), you can use the following workflow to use a multilevel list and style mapping to automatically number the clauses in your document and reset the clause numbering after each section title.

> ⓘ Note
>
> The Microsoft Word field names and workflow can vary depending on your Microsoft Word version and language.

1. Create unnumbered Microsoft Word paragraph styles that you will map to the SAP Ariba labels generated for section titles and content or clauses.
2. Create a Microsoft Word multilevel list with the **Start at** and **Restart list after** fields to automatically re-start numbering.
3. Link each level in the list to a paragraph style that you created in step 1.
4. Set up styles as described in Assembled Contract Document Creation in the Enhanced Workflow [page 12] and link the paragraph styles that you created in step 1 to the SAP Ariba labels generated for section titles and content.

For example, the user creates the following paragraph styles:

- `My Title Level 1 Style`
- `My Content Level 1 Style`

The user defines a new Microsoft Word multilevel list. The levels have the following field values:

- Level 1
  - **Enter formatting for number:** `1`
  - **Start at:** `1`
  - **Link level to style:** `My Title Level 1 Style`
- Level 2
  - **Enter formatting for number:** 1.1
    To create this numbering format:
    1. In the **Define new Multilevel list** Microsoft Word dialog box, click the link level **2**. Click **More >>** to open the full dialog box.
    2. Click the drop-down menu for **Include level number from** and select **Level 1**. A `1` appears in the **Enter formatting for number** field.
    3. In the **Enter formatting for number** field, enter a dot (`.`) after the `1`.
    4. Click the drop-down menu for **Number style for this level** and select `1, 2, 3 ...`.
  - **Start at:** `1`
  - **Restart list after:** `Level 1`
  - **Link level to style:** `My Content Level 1 Style`

The user configures style mapping to map the labels to paragraph styles as follows:

SAP Ariba Label	Paragraph Style
Title Level 1	`My Title Level 1 Style` (linked to multilevel numbering level 1, with numbering format 1)
Content Level 1	`My Content Level 1 Style` (linked to multilevel numbering level 2, with numbering format 1.1)

Using these definitions, an assembled contract document will have numbering similar to the following:

1 Scope of Agreement (**Title Level 1**; this is the section title—the first paragraph in this section)

1.1 A Clause (**Content Level 1**; this is the first paragraph of the first clause in the section)

1.2 Another Clause text (**Content Level 1**; this is the first paragraph of the second clause in the section)

2 Period of Agreement (**Title Level 1**; this is the title of the second section)

2.1 Yet Another Clause (**Content Level 1**; this is the first paragraph of the first clause in the second section)

Migration of Previous Documents Containing Bookmarks

Templates and contract documents created using the previous workflow must be redesigned to work with the enhanced workflow.

SAP Ariba recommends the following strategy for migrating documents created using the previous workflow to the enhanced workflow:

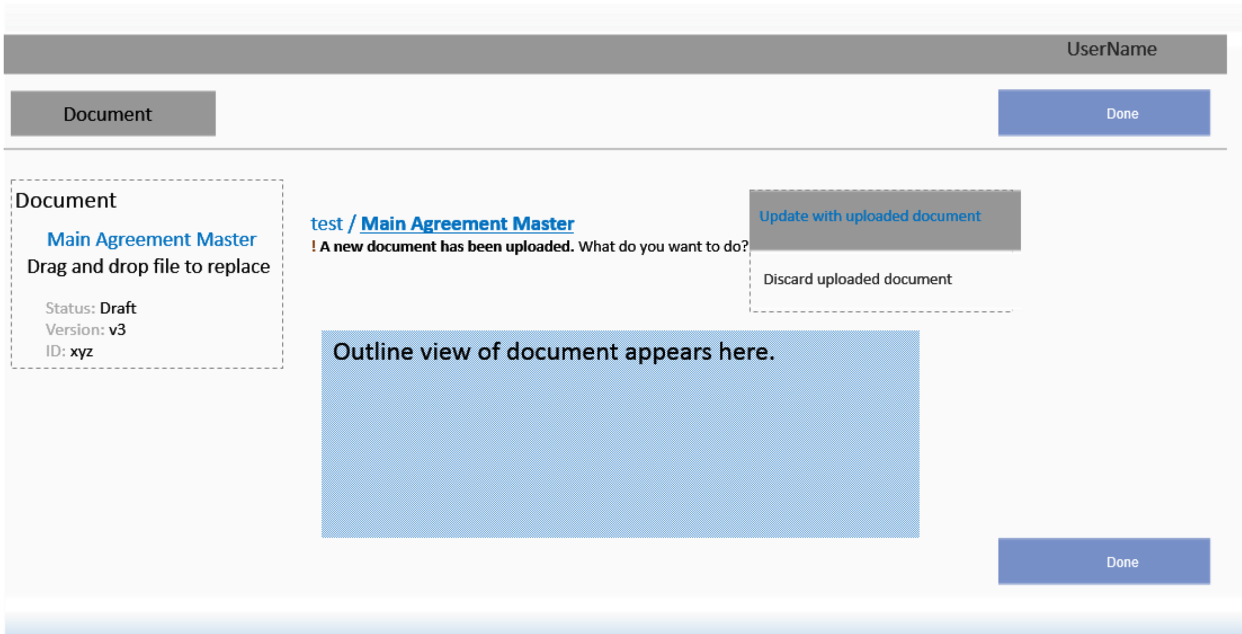
- Create new templates or edit your templates to conform to the new styling described in Assembled Contract Document Creation in the Enhanced Workflow [page 12].
- Create simply-styled versions of the contract workspace documents inheriting the templates, as described in Assembled Contract Document Creation in the Enhanced Workflow [page 12]. Although you can open your previous documents without this step, SAP Ariba recommends that you perform this step to make viewing documents in the **Outline View** or reviewing and modifying documents a consistent experience.
- After you have migrated your templates and contract documents, contact your designated SAP Ariba Customer Support representative to enable the enhanced contract authoring workflow.
- Verify that opening, updating, and reviewing documents in the enhanced workflow work correctly.

Document Upload Using the Drag and Drop Method

With enhanced contract authoring, you can directly drag and drop the documents into your contract workspace.

> ⓘ Note
>
> Enhanced contract authoring must be enabled to use this feature.

The following is a view of the contract workspace screen with drag-and-drop.



As shown in the above view, you can drag and drop your updated document to the top left corner of the screen into the area surrounded by the dashed lines. When you drag and drop the updated document, you are alerted that a new version of the document is uploaded. You can choose to update the exisitng document or to discard your uploaded document.

Document Review Using the Enhanced Redlining User Interface

A streamlined user interface is provided for comparing and merging documents.

SAP Ariba Contracts used the Microsoft Word document merge functionality to allow you to compare and merge assembled documents to manage various document versions during negotiations or reviews. Performing a document merge also required you to enable DFS.

With enhanced contract authoring, you can continuet to send a document to an offline reviewer (such as a supplier or customer) by making that reviewer an email reviewer. The reviewer changes the document and returns it as an attachment to you, the task owner. However, reviewing and merging changes uses a simplified redlining interface.

> ⓘ Note
>
> This feature requires the “ActiveX Removal” feature and the “Drag And Drop” feature. To enable these features, have your Designated Support Contact log a service request. An SAP Ariba Support representative will follow up to complete the request.

You are shown the number of changes and the suggested additions and deletions. To accept all the changes or reject all the changes, you choose **Process Changes** and **Reject Changes**, respectively. After you have accepted or rejected all the changes, you can save the document as a new version or you can replace the current version of the document.

Document merge also allows you to combine two assembled documents into one document. You can merge a document from within SAP Ariba Contracts or choose to merge a document from an external source.

When you upload a contract document such as a main agreement to your contract workspace, track changes is enabled automatically on the document even if the document was not created with track changes enabled. Subsequent updates to the document can thus be tracked automatically.

In addition, if you open the main agreement in Outline View and make changes to it, such as add or delete a section or a clause, these changes are displayed in the Outline View and are also visible when the document is downloaded or the document is compared with previous versions.

The clause title and folder navigation information are visible in the outline view, enabling you to easily identify which clauses are included in the assembled document. When you select the **Title** option from the **Table Options** menu, the Outline View displays the **Title** column. For clauses that are added from the clause library, the clause title is displayed in the **Title** column next to the clause. Hovering over the clause title or clicking the clause title displays the complete folder navigation information to enable you to identify which clauses are included in the assembled document.

To easily co-relate clauses with the section hierarchy when scrolling through or comparing versions of lengthy documents with several clauses under the corresponding sections or sub-sections, you can view the section hierarchy of a clause by a mouse-over (which displays a tool-tip) in the outline and comparison view of a document. You can also view the section hierarchy details at the section-name level through a breadcrumb that displays the full section path.

Assembled Contract Documents

About Working with Assembled Contract Documents

If you do not have an assembled contract document (Main Agreement or Contract Addendum) that is already loaded in a project, see Loading Assembled Contract Documents [page 44].

If you are working with an assembled contract document that is already loaded in a project, you can edit the document using Microsoft Word or any other OpenXML editor. However, in many cases, it is easier to use the **Outline View** tab to view, move, add, substitute, and deleted the components of the document—the clauses and sections.

A clause is one or more paragraphs treated as a single unit. Clauses are stored in and retrieved from the SAP Ariba Contracts Clause Library. Sections contain and organize clauses.

The **Outline View** tab of an assembled contract document displays an outline of the document, with representations of the sections and clauses. You can use menu options and drag-and-drop operations on the representations of the sections and clauses to move, replace, and delete sections and clauses. Using the outline view to manipulate sections and clauses minimizes problems with clause and section boundaries.

If you need to load an assembled contract document, see Loading Assembled Contract Documents [page 44].

Editing an Assembled Contract Document with the Outline View

Use the outline view to display a representation of an assembled contract (Main Agreement or Contract Addendum) and modify the components of the document—the sections and clauses.

Procedure

1. Open the project and navigate to the **Documents** tab.
2. Click the document title of an assembled document and choose **View Details**.
3. Click the **Outline View** tab.

   The outline view shows an outline of an assembled contract document, with representations of the document sections and clauses. Sections create a hierarchical organization in a contract document. Sections are indicated in bold type with an expand arrow on the left:

   ▼ **My Section**

   Sections exist only on a per contract document basis and do not exist in the Clause Library.

   SAP Ariba Contracts displays a snippet (the first 200 characters) for each clause in the outline view. You might see the following clause types in the Outline View:

   - **Template**: Clause was part of the original template.
   - **Library**: Clause that you substituted from the Clause Library.
   - **Nonstandard**: Clause that you modified by manually editing the text.
   - **Ad Hoc**: Clause you add from an external source.

   Each clause is stored in the Clause Library as a separate Microsoft Word document. Clauses are added to the Clause Library with titles to help members of the Clause Library team organize the clauses, but clause titles are not shown in the outline view.

   An **ALT** icon might be shown next to a clause. This indicates that the Clause Library contains one or more published clauses in the same folder that are intended as alternate clauses. For information about substituting a clause with an alternate clause, see Replacing (Substituting) Clauses in Assembled Contract Documents [page 23].
4. Complete one of the following actions:
   - Add or remove a clause [page 24]
   - Replace (substitute) a clause [page 23]
   - Delete a clause [page 25]

   You can also perform the actions listed in the tables below:
   - Actions You Can Perform on Clauses from the Outline View [page 22]
   - Actions You Can Perform on Sections from the Outline View [page 22]
   - Actions You Can Perform on a Document from the Outline View [page 23]

5. After you complete your changes, you can finalize [page 36] your document. This changes the document state so it cannot be changed either manually, or by automatic updates of document property fields when contract field values change.

Actions You Can Perform on Clauses from the Outline View

To Do This...	Choose...
Add a new clause or several new clauses to the document	**Actions** > **Add Clause(s)** to access the Clause Library. Expand the folders in the Clause Library and select the check boxes for the clauses you want to include.
Edit a clause	Download the document to your system, edit it using Microsoft Word or any other OpenXML editor, and upload it back or drag and drop it back into the drag-and-drop area in the top left corner of the Document screen.
Substitute a clause	the clause and choose **Actions** > **Substitute** to select a substitute clause from the Clause Library. When you substitute a clause, the document on your desk top and the Outline View update. See Replacing (Substituting) Clauses in Assembled Contract Documents [page 23] for additional information.
Move a clause	the clause and drag and drop it to the new location.
Delete a clause	the clause and choose **Actions** > **Delete** to delete a clause. SAP Ariba Contracts replaces this clause with the text “Clause intentionally deleted.” See Deleting Clauses [page 25] for additional information.
View versions of a clause	the clause and choose **Actions** > **View Versions** to view the versions of a clause. You can select the clause name and **Replace Document** to replace a one version of a clause with another.
Convert a clause into a section	the clause and choose **Actions** > **Promote to Section** to convert a clause into a section heading. See Promoting a Clause to a Section [page 27].
Filter clause display	the **Clause Type** menu and choose a clause type. This filters the clauses you see in the Outline View. You can choose **All** to show all clauses.

Actions You Can Perform on Sections from the Outline View

To Do This...	Choose...
Add a new section to the document	**Actions** > **Add New Section** to add a new section to the document. You can drag and drop the new section to position it, and drag clauses into the new section.
Expand collapsed sections, or collapse expanded sections	the table icon and choose **Outline Expansion** > **Expand All** or **Collapse All**.
Export a list of clauses to Microsoft Excel	the table icon and choose **Export to Excel** > **Export all Rows**.

To Do This...	Choose...
Add a subsection	the section name and choose **Add New Section** to create a subsection.
Add a clause to the section	the section name and choose **Add Clause(s)**.
Delete a section	the section name and choose **Delete**.
Edit a section	the section name and choose **Edit**.

Actions You Can Perform on a Document from the Outline View

Context

To Do This...	Choose...
Save a new version of the document	**Actions** > **Save as New Version** to save a new version of the document. You must assemble the document before you can save a new version.
Add fields that are updated automatically in your document	**Actions** > **Set Up Document Properties**. Document properties are fields that Microsoft Word inserts into your assembled document and updates when you update the contract workspace. These can include editable properties, such as an effective date, or non-editable properties, such as a suppler name.
Display or hide navigation	**Actions** > **Show/Hide** > **Outline/Document Navigation.**
Compare two versions of the document	**Actions** > **Show/Hide** > **Compare Versions Filter**. This filter allows you to compare two versions of a document. Select the versions to compare from the **Compare Versions Between** menus, or click **Compare in Microsoft Word**. Using this filter add the Change column, which lists the type of changes to the various clauses. You can also select **Actions** > **Edited Clause(s) Only** to filter the document further and display only the modified clauses.
Control the columns displayed in **Outline View**	the table icon and choose **Show/Hide Columns** > **Clause**, **Owner**, **Conditions**, **Type**, **Comments**, or **Action** to display or hide these columns. See Making Clauses Conditional [page 28] for details on making individual clauses conditional.

Replacing (Substituting) Clauses in Assembled Contract Documents

Procedure

1. Open the outline view for the assembled contract document, as described in Opening the Outline View of an Assembled Contract Document [page 21].
2. Go to the left of the clause and click **Actions** > **Substitute**. A pane showing the Clause Library opens. Expand the folders in the Clause Library.

Alternatively, if the clause you want to replace has an **ALT** icon displayed, the Clause Library contains one or more published clauses in the same folder that contains the current clause; these clauses are intended as alternate clauses. Click the **ALT** icon. A pane showing the folder in the Clause Library opens.

3. (Optional) Use the labels (if present) next to the clauses in the library to help you select the substitute clause. These labels are applied by a member of the Clause Library Team and can be used with the following meanings:
   - **Preferred**: A clause that is preferred for use at the start of supplier or customer negotiations.
   - **Alternate**: A clause with alternate, but equally correct language for a Preferred clause.
   - **Fallback**: A clause to use if a supplier or customer rejects the language in either a Preferred or Alternate clause.
4. Select the check box for the clause you want to use as a substitute. Click **OK**.

   If you are working on a document in a contract workspace and substitute a substitute a pre-approved clause from the Clause Library, the type changes to **Library**. If you are working on a document in a project template, the type of the substituted clause does not change.

   > → Tip
   >
   > Typically, a NonStandard clause substitution is completed in the second attempt. So, when you are substituting a NonStandard clause, if the clause substitution does not succeed in the first attempt, try again.

5. The outline view shows an information icon next to the substitute clause:

   i

   Clicking the information icon opens a text box with the clause description added by the clause author. You can replace or add to this text with information, such as an explanation for your substitution.

Adding and Moving Clauses in Assembled Contract Documents

Procedure

1. Open the outline view for the assembled contract document, as described in Opening the Outline View of an Assembled Contract Document [page 21].
2. To add clauses from the Clause Library to a specific section, click the section and choose **Add Clause(s)**.

   To add clauses to the end of the document, go to the upper right corner of the outline view and click **Actions** > **Add Clause(s)**.

   A panel showing the Clause Library opens. Expand the folders in the Clause Library and select the check boxes for the clauses you want to add to your document and click **OK**.

   > Note
   >
   > If you need to add more than a 1000 clauses, contact your Designated Support Contact to enable the parameter `Application.ACM.FixProxyErrorInCWWithLotsOfLibraryClauses`.

3. To move a clause, click the clause and drag and drop it to the new position.

Deleting Clauses

Context

Your default installation includes an **Empty Clause** in the Clause Library. The Empty Clause is a clause with the name and type Empty Clause. By default, when you delete a clause, SAP Ariba replaces it with the Empty Clause. This behavior is intended to preserve the numbering in the clause document and to prevent invalid references to particular section or clause numbering in the assembled document. If the site configuration option `Application.ACM.SoftClauseDeletion` is modified and set to `No`, deleting a clause removes the clause without replacing it. For more information, see Site Configuration Options for Contract Authoring [page 79].

The Empty Clause contains the following default text:

```
Clause intentionally deleted.
```

If your Clause Library does not contain a published Empty Clause, you might see unexpected behavior when you delete clauses. For example, you might appear to be unable to delete a clause from the Outline View, or see that clauses deleted through Microsoft Word do not delete in the Outline View.

Procedure

1. Verify that there is a published clause with the name and type **Empty Clause** in the Clause Library.
2. To delete a clause using the **Outline View** tab (SAP Ariba recommends that you use this method):
    1. Open the outline view for the assembled contract document, as described in Opening the Outline View of an Assembled Contract Document [page 21].
    2. Go to the left of the clause you want to delete and click **Actions** > **Delete**.
3. To delete a clause using Microsoft Word (SAP Ariba recommends that you do **not** use this method):
    1. Download the document name to your system.
    2. Open the document in Microsoft Word or any other OpenXML editor.
    3. Delete the clause.
    4. Save the document and upload it or drag-and-drop it into the application.

Uploading Changes from a Document File to the Outline View

Context

With enhanced contract authoring, you can drag and drop an updated document to the drag-and-drop area of the **Document** section in the top-left pane. After you drag and drop the updated document, a message is displayed to indicate that a new document has been uploaded. Click **What do you want to do?** to perform one of the following actions:

- Review the uploaded document by clicking **Review uploaded document** and then selecting either **Process All Changes** or **Reject All Changes**.
- Update the existing document with the uploaded document by clicking **Update with uploaded document**. You can choose to save the uploaded document as a new version or replace the current version of the existing document.
- Discard the uploaded document by clicking **Discard uploaded document**.

For more information, refer to Document Upload Using the Drag and Drop Method [page 17] and Document Review Using the Enhanced Redlining User Interface [page 18].

You can also upload a document by using the **Upload** button on the **Overview** tab. Click **Upload** and click **Choose File** to locate the document and select it to upload.

.

Editing Clause Text in Assembled Contract Documents

Procedure

1. Open the outline view for the assembled contract document, as described in Opening the Outline View of an Assembled Contract Document [page 21].
2. Download the file to your system. Edit the clause text in Microsoft Word or any other OpenXML editor.
3. Save the file and exit your editor.
4. Drag the modified file on your desktop into the drag-and-drop area in the top-left corner of the application (or upload the document after browsing to the file on your system) to update the document.

Editing an Assembled Contract Document in Microsoft Word

In most cases, it is easier to add, replace, move, and delete clauses in an assembled contract document (Main Agreement or Contract Addendum) from the **Outline View** tab. However, if you are not using the Clause Library and want to edit the contents directly, you can edit the document directly using Microsoft Word or any other OpenXML editor.

Procedure

1. Open your project and navigate to the **Documents** area.
2. Click the Microsoft Word icon next to the assembled contract document.
3. A Microsoft Word file for your document is downloaded. Edit the text in any OpenXML editor. SAP Ariba recommends that you use Microsoft Word. SAP Ariba recommends that you:
    - If you want to replace the text, delete the unwanted text first, and then add the new text.
4. Save the file, and exit your editor.
5. Drag the updated file to the drag-and-drop area on the top left area of the screen or upload it after browsing to the file on your system. An alert asks you whether you want to to review the document or discard the changes. If you review the document, you will be redirected to the new redlining view (if you have requested for enabling enhanced contract authoring features) where you can view the changes and accept or discard the changes.

Promoting a Clause to a Section

Context

Promoting a clause to a section enables you to change a clause to a section in the Outline View.

All ad hoc content that you add to a contract document in Microsoft Word during a negotiation is added as a clause. If new sections are incorrectly displayed as clauses in the Outline View of the contract document, you can use **Promote to Section** to change the clauses to sections. Bookmarking does not affect this promotion.

⚠ Limitations

The following points apply if you promote a clause to a section:

- After you promote a clause to a section in the Outline View, you cannot reverse the action. If you promote a clause in error, you must delete the section from the Outline View and add the text again as a clause.
- SAP Ariba Contracts uses the first paragraph of a section as the section title. See Actions You Can Perform on Sections from the Outline View [page 22] for section title restrictions.
- The character limit display in the Outline View for sections is 255 characters. If the clause you are promoting to a section is longer than 255 characters, not all of the characters appear in the Outline View. All of the characters display in the Microsoft Word document, however. Section titles do not wrap in the Outline View.
- The section you promote remains in the same place as the originating clause. You must use drag and drop to reposition it.
- If you need to promote multiple clauses to sections, you should promote them all and then drag and drop clauses into them. If you make significant changes to the document structure, you can promote no more clauses. At this point, you must regenerate the document, and then return to the Outline View and promote the rest of the clauses to sections.

- Each time you promote a clause to a section, you are changing the structure of the document. SAP Ariba Contracts recalculates the relationship between the sections and clauses for each section promotion. This process can be slow for lengthy documents.

Procedure

1. Navigate to the Outline View of a contract document in either a template or a contract workspace. Make sure the document is not published.
2. Choose **Actions** > **Promote Clause to Section** to the right of the clause you want to promote.
3. Drag and drop the section to the proper location in the Outline View, and drag appropriate clauses into it.

Making a Conditional Clause

When working on assembled contract documents (Main Agreement or Contract Addendum documents) in project templates, you can apply conditions to include or exclude individual clauses depending on condition values. For example, you could include a clause only if the **Regions** field is **APAC Asia Pacific**. To do this, you first define a condition that is true if the **Regions** field is **APAC Asia Pacific**. Then, you would apply the condition to the clause.

Prerequisites

You make clauses conditional in project templates. You must be a member of the Template Creator team or a member of the Project Owner team in a project template to edit project templates.

Procedure

1. Open the project template containing the assembled contract document. If the project template status is not Draft, create a new version of the template by navigating to the Properties pane and choosing **Action** > **Template** > **New Version**.
2. Define the condition for the clause on the **Conditions** tab. For details on defining conditions, see Defining Condition Expressions Using the Condition Editor.
3. Navigate to the **Documents** tab of the template.
4. Click the contract document name and choose **View Details**.
5. Click **Actions** and choose **Conditions** to display the Conditions column in the Outline View of the contract document.
6. In the Conditions column for the clause you want to make conditional, click **(none)** and select the condition you previously defined.

You can also make a section conditional. By making a section conditional, you automatically make all clauses in that section conditional. SAP Ariba Contracts indicates the clauses in the section as `inherited` in the Conditions column.

Applying Conditions in Bulk to an Assembled Contract Document

Prerequisites

You must be able to add documents to a project template. For example, members of the **Template Creators** group and members of the Templates team can do this. You must have Enhanced Contract Authoring enabled to use this feature.

Context

You can use a Microsoft Excel document to associate conditions with clauses or sections in a Main Agreement or Contract Addendum (called an assembled contract document). You can use conditions to include clauses or sections in assembled contract documents based on the value of project header fields. For example, you could include a clause with United States environmental regulations in a Main Agreement if the Regions project header is United States.

The first step is to export the conditions file from the assembled contract document. Make any updates to the file, including adding conditions, and then import the updated file.

Procedure

1. Open a Contract Workspace template.
2. Upload a Main Agreement or Contract Addendum document.
3. In the **Outline View** tab, on the **Actions** menu, under **Condition mapping**, click **Export**.
4. Examine the exported conditions file and prepare any changes.

   The conditions file is an Excel file that contains two sheets.

   The first sheet, **ConditionMapping**, maps clauses or sections to conditions. This sheet includes these columns:

   - **Internal Id**: Unique identifier for a section or clause in an assembled contract document. These internal IDs are generated when you export the conditions file. Clause internal IDs start with "Doc" and section internal IDs start with "WS".
     Conditions are applied based on the internal ID only; the order of the content in the file does not matter.

> ⓘ Note
>
> Internal IDs change from version to version of the same document. Before you import a conditions file into a document, make sure that you have applied conditions to a freshly exported conditions file from the same version of the document.

- **Section Title/Clause Text**: Either a section title or the text of a clause in an assembled contract document. You can apply conditions to either clauses or sections.
- **Compound Type**: Enter `Any of`, `All of`, `None of`, or `Never matches`, which apply to the conditions referenced in the **Conditions** column. For each of these compound types, you must use the appropriate number of conditions, as follows:
  - `Any of`: 0 or more conditions
  - `All of`: 2 or more conditions
  - `None of`: 1 or more conditions
  - `Never matches`: exactly 0 conditions

> ⓘ Note
>
> The compound type cannot be blank.

> ⓘ Note
>
> `Never matches` means to remove the section or clause from the document.

- **Conditions**: Entry in the **Unique Name** column in **Conditions**, the second sheet in the conditions file. You can separate multiple conditions by using the vertical bar (|).
- **Update Condition**: Enter `true` or `false`. True means to update the **Conditions** and/or **Compound Type** on that row; false means to leave them as is. When you initially export the conditions file, the **Update Condition** column is set to **false** for all rows.

The second sheet, **Conditions**, defines conditions and includes two columns:

- **Unique Name**: The unique name of the condition, which you enter into the **Conditions** column in the **ConditionMapping** sheet.
- **Expression**: The condition itself; for example `(OR ((Region)is equal to(China)))`.

> ⓘ Note
>
> The expressions on the **Conditions** sheet are for reference only. You cannot re-import them.

For more information on creating conditions, see Making a Conditional Clause [page 28], which describes how to associate conditions with one clause in a document. For details about the syntax of conditions, see Defining Condition Expressions Using the Condition Editor.

5. To import the conditions file, return to the **Outline View** tab, navigate to the **Condition mapping** area, and click **Actions** > **Import** . Browse to the location of the file you want to import.

   The import may not happen in real time. Click the **Refresh Status** button if it appears.
6. After the import, check the **Status** column, which will show one of these values:
   - **In Progress**: The import process is in progress.
   - **Completed**: The import process successfully completed processing all the records (rows) in the Excel file.
   - **Completed with errors**: The import process completed processing all the records (rows) in the file, but there are errors in the Error Log.

- **Failed**: The import process failed. It did not process the entire set of records in the file.

7. If necessary, click **View Details** in the **Error Log** column to see any errors.

   If you see only one row (the header row), there are no errors.

Bulk Upload of Conditions to a Contract Workspace Template

You can use a text-based syntax in Microsoft Excel to bulk-load conditions. Conditions can be either simple expressions or compound expressions composed of other compound expressions and/or simple expressions.

Prerequisites

You must be able to add documents to a project template. For example, members of the Template Creators group and members of the Templates team can do this. You must have Enhanced Contract Authoring enabled to use this feature.

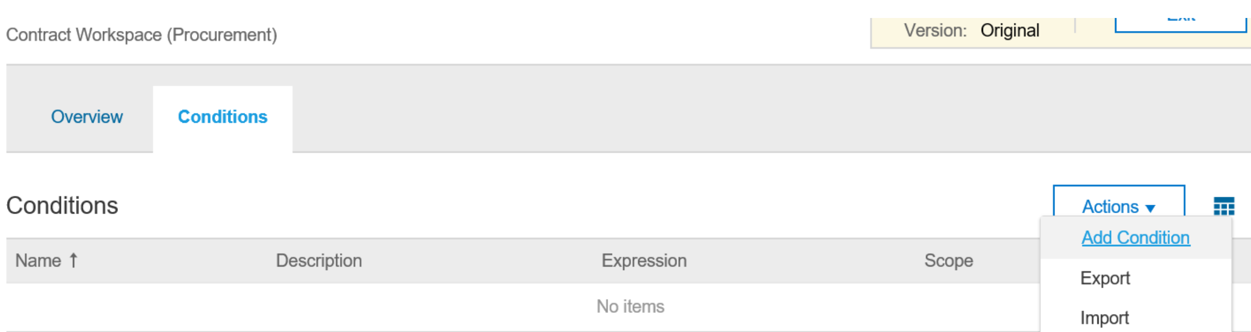
To bulk-load conditions, first export conditions from the **Conditions** tab of a Contract Workspace template. Then, in the exported spreadsheet, put the expressions in the **Conditions** spreadsheet in the **Expression** column:

1	Unique Name	Description	Expression	Update Condition
6	reg, comm, dept		(AND(((Region)is child of(EMEA))((Commodity)is in list(unspsc:5310|unspsc:5311))((Client)is equal to(300))))	false

Workflow for Bulk-Loading Conditions into a Contract Workspace Template

The workflow for bulk-loading conditions is separate from the workflow described in Applying Conditions in Bulk to an Assembled Contract Document [page 29]. The syntax for writing conditions is different from the syntax described in that topic. However, as is true for the other workflow, it is helpful to create conditions, export them, make changes, and then re-import the Excel file.

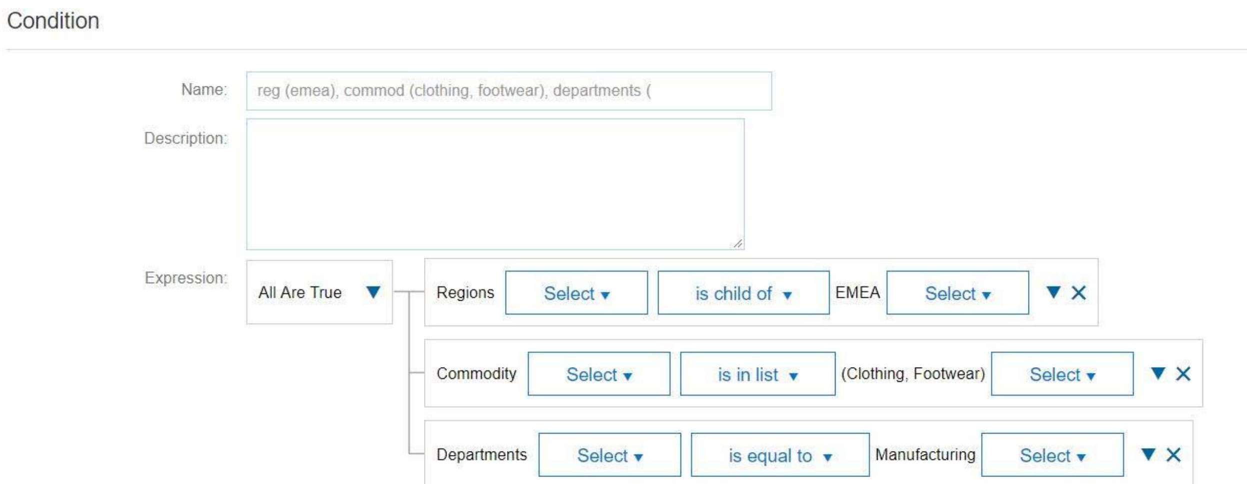
For example, if you want to create different conditions based on region and department, you can create one condition in the template, export the condition, cut and paste in Microsoft Excel to make the changes you want, and then re-import the conditions. When the conditions are imported, they appear on the **Conditions** tab of the Contract Workspace template.



In the **Update Condition** column, enter true or false. True means to update the condition on that row; false means to neither create nor update the condition on that row. When you initially export the conditions file, the **Update Condition** column is set to false for all rows.

Creating Text-Based Syntax for Conditions

In the user interface, you can create a compound expression like the following:



For details on how to create conditions in the user interface, see Defining Condition Expressions Using the Condition Editor.

Each of these three lines is a simple expression that evaluates to true or false. You can link these three lines by choosing **All Are True**, **Any Are True**, or **None are True**.

- **All Are True** is like using "and" between the simple expressions. The compound expression using **All Are True** evaluates to true only if all three simple expressions are true.
- **Any Are True** is like using "or" between the simple expressions. The compound expression using **Any Are True** evaluates to true if any one of the three simple expressions is true.
- **None Are True** is like using "not...and not...and not" between the simple expressions. The compound expression using **None Are True** evaluates to true only if none of the simple expressions is true.

Using **All Are True**, the compound expression shown above can be written as: **`(AND(((Region)is child of(EMEA))((Commodity)is in list(unspsc:5310|unspsc:5311))((Client)is equal to(300))))`**. This is the syntax for Excel import.

> **Note**
>
> Use the unique names of the fields instead of their localized user interface labels. For example, **Regions** from the UI becomes `Region`, and**Departments** becomes `Client`.

In properly indented format, the expression above looks like this:

```
(AND
        (
                ( (Region) is child of (EMEA) )
                ((Commodity)is in list(unspsc:5310|unspsc:5311))
                ((Client)is equal to(300))
        )
)
```

- The outer parentheses, shown in yellow, are the root parentheses of the compound expression. These parentheses enclose the entire expression.
- The next set of parentheses, shown in turquoise, are the compound expression parentheses. These parentheses enclose all child expressions, simple and compound.
- The next set, shown in green, are the simple expression parentheses.
- Finally, the innermost set, shown in gray, are the field name or field value parentheses. Both field names and field values must be enclosed in parentheses.

Parentheses must be in matched sets. Unmatched parentheses cause an error. Note that there are no separators between expressions.

To easily write this syntax, start with the basic format:

```
(AND
    (

    )
)
```

Write the simple expressions one at a time. First write:

```
(AND
    (
        ((Region) is child of (EMEA))
    )
)
```

Then write the Commodity line:

```
(AND
    (
        ((Region) is child of (EMEA))
        ((Commodity) is in list(unspsc:5310|unspsc:5311))
    )
)
```

Use unique lookup keys for field values. Use a colon in case of multiple lookup keys. For example, above, `unspsc:5310` represents the Clothing commodity in the UNSPSC domain. The vertical bar (|)separates multiple field values.

Fields can be of the following types: Boolean (true/false), Date, Decimal Number, Double (decimal), Group (unique name), Integer, Money, Multiline Text (String), Percentage, String, Text (String), Text Multiple Select (multiple strings), URL Link (String), User (unique name), and User Multiple Select.

> ⓘ Note
>
> For the date, use the concise date format that corresponds to your locale; for example, MM/DD/YYYY for the US locale. For another locale, you may need to use DD/MM/YYYY or whatever your standard date format is.

Predefined fields that can be used include `Commodity`, `Region`, `Client`, `Supplier`, and `Organization`. For a complete list of the predefined fields, and for information about custom fields, see Using Predefined and Custom Fields When You Bulk-Load Conditions in a Contract Workspace Template [page 34].

Some data field values must match specific values in the system, such as values for regions or commodity codes. For these field types, use an export task to find out what the specific values should be. If you can't access the necessary export task, contact your Customer Administrator.

Using Predefined and Custom Fields When You Bulk-Load Conditions in a Contract Workspace Template

When you use Microsoft Excel to bulk-load conditions into a Contract Workspace template, you can use both predefined and custom fields.

The table below shows the names of the predefined fields to use when you are bulk-loading conditions into a Contract Workspace. In the Excel file, use the unique name listed in the first column below, not the user interface name listed in the second column.

If your site uses custom fields and you want to generate unique names from the user interface names, add the prefix **cus_** and remove any spaces from the user interface name. For example, a custom field with the user interface name **Purchasing Order Amount** has the unique name **cus_PurchasingOrderAmount**. You can get the unique names of fields by first exporting conditions as described in Bulk Upload of Conditions to a Contract Workspace Template [page 31]. If a condition is defined on a field, you'll see the unique name of the field in the spreadsheet as part of the expression for that condition.

Unique name	User interface name
Active	Active
AdhocSpendUsers	Ad hoc Spend Users
AffectedParties	Affected Parties
AgreementDate	Agreement Date
AllowAdhocSpend	Allow Ad hoc Spend
AmendmentReason	Amendment Reason
Amount	Contract Amount
arb_CompanyCode	Company Code
arb_DocumentCategory	Document Category
arb_DocumentType	Document Type

Unique name	User interface name
arb_PaymentTerms	Payment Terms
arb_PurchasingGroup	Purchasing Group
arb_PurchasingOrganization	Purchasing Organization
AutoRenewalCount	Auto Renewal Count
AutoRenewalInterval	Renewal Interval (in months)
BaseLanguage	Base Language
BeginDate	Start Date
BusinessSystem	External System
Client	Departments
Commodity	Commodity
ComplexSpendAmount	Total Approved Amount
ComplexSpendAvailableAmount	Available Amount
ComplexSpendReleaseApprovers	Release Approvers
ComplexSpendReleaseCreators	Release Creators
ContractCreationSupplierClassification	Minimum supplier status for contract creation
ContractId	Contract ID
ContractIntegrationSupplierClassification	Minimum supplier status for contract integration with ERP
ContractPublishingSupplierClassification	Minimum supplier status for contract publish
ContractStatus	Contract Status
DisplayStatus	Status
EffectiveDate	Effective Date
EndDate	End Date
EngagementRequestId	Engagement Request Project
ExpirationDate	Expiration Date
ExpirationTermType	Term Type
ExpiringEmailRecipients	Expiration Email Recipients
ExpiringReminderBegin	Email Notification
ExpiringReminderFrequency	Expiring Reminder Frequency
ExternalPublishVersion	External Publish Version
HierarchicalType	Hierarchical Type
InternalId	ID
IsComplexSpendWorkspace	Combined Spend
IsEvergreen	Is Evergreen

Unique name	User interface name
IsInProjectTemplate	Is In Project Template
MaxAutoRenewalsAllowed	Maximum number of Renewals
NoticeDate	Notice Date
NoticeEmailRecipients	Notice Email Recipients
NoticePeriod	Notice Period (in days)
NoticeReminderBegin	Notice Email Notification
NoticeReminderFrequency	Notice Reminder Frequency
Origin	Origin
OriginalExpirationDate	Original Expiration Date
Owner	Owner
ParentAgreement	Parent Agreement
ProcessStatus	Process Status
ProposedAmount	Proposed Contract Amount
ProposedIncrementalAmount	Proposed Incremental Amount For Approval
Region	Regions
RelatedId	Related ID
SelectLineItemToSendExternalSystem	Enable Select Line Items To Send External System
Status	Project State
Supplier	Supplier
TimeCreated	Time Created
TimeUpdated	Time Updated
UserGroups	User Groups
UserPermissions	User Permissions
Workspace	Workspace

Finalizing Assembled Documents

You can specify that a published assembled document remain unchanged (“final”) in terms of the body of the document and the document attributes. You can use this feature if you are working on or amending a contract workspace and your assembled documents are in a state that you do not want to change, either manually, or by automatic updates of document property fields when contract field values change.

Procedure

1. In the **Documents** tab of your contract workspace, select the published assembled document name.
2. To automatically finalize the document when the workspace is published, choose **Action** > **Edit Attributes**. On the attributes page, select **Yes** for **Auto Finalize on Publish**. Click **Save**.

   You can also set the **Auto Finalize on Publish** option when you load an assembled contract document.
3. To finalize the document immediately, choose **Action** > **Finalize**. The status of the document changes to Finalized.

Results

You can later make the document modifiable by selecting **Action** > **Make Draft.** You can select **Make Draft** while you are still working in the contract workspace, or during an amendment. This returns the document to the Draft status. You can modify the document, assemble and publish it, and then finalize it again as needed.

Embedment of CLIDs as Tables in Assembled Contract Documents

Contract line item terms and ad hoc sourcing event questions can be added to a contract document (main agreement or contract addendum) that is associated with a contract workspace template or contract workspace in table format.

You can also create a custom table style in the Microsoft Word version of a contract document and have this table style used by default when contract line item terms and sourcing event questions are rendered in tables within a contract document. For example, you might want to use a table format with specific borders, cell coloring, and so forth. For more information, see Site configuration parameters below.

In a downloaded contract document, if the change tracking feature is turned on in Microsoft Word, the tracking feature highlights tables that were added or that were deleted from the SAP Ariba system. The name of the user that made the change within the system is associated with each change. For example, if a new table definition is added in SAP Ariba, that table is highlighted in the downloaded document, and the name of the user that made the change is associated with the change. You can accept or reject any changes in the downloaded document using the change tracking functionality.

> ⓘ Note
>
> The content of contract line item terms and sourcing event questions tables is added by the SAP Ariba system only. Any updates that are made to a table by a user when a contract document is downloaded are not retained if the document is re-uploaded.

For information about adding ad hoc sourcing event questions to a contract workspace, see Adding Sourcing Event Questions to a Contract Workspace.

For information about parameters in Intelligent Configuration Manager, see Intelligent Configuration Manager Parameters Reference.

Prerequisites

- SAP Ariba Strategic Sourcing Suite must be enabled for your site.
- The following site configuration parameter must be set to `Yes`: `Application.ACM.TableSupportInContractDocuments.Enabled`

  Your administrator can set this parameter in Ariba Administrator under **Intelligent Configuration Manager** › **Manage Configurations**.
- The feature UP-845 ActiveX Removal must be enabled for the site
- The following parameter must be enabled (set to Yes) to be able to create a contract line items document in a contract workspace: `Application.ACM.ContractLineItemsDocument.Enabled`

  Your administrator can set this parameter in Ariba Administrator under **Intelligent Configuration Manager** › **Manage Configurations**.

Restrictions

- For contract line items tables, this feature only applies to line items documents for which the Capacity type is Standard (which is a maximum of 2000 line items). It does not apply to line items documents with Capacity type Large.
- Attachment, Certificate, Product Questionnaire terms are not supported in tables.
- Cost groups terms and advanced pricing conditions are not supported.

Site Configuration Parameters

`Application.ACM.TableSupportInContractDocuments.TableStyleName`

Allows you to apply a table style that you create in Microsoft Word to tables that are generated for contract line item terms and sourcing event questions when they are added to contract documents (main agreement or contract addendum). You can create a table style in Microsoft Word and copy the style name to the Value field for this parameter.

Your administrator can set this parameter in Ariba Administrator under **Intelligent Configuration Manager** › **Manage Configurations**.

Related Information

Adding a Sourcing Event Questions Table Definition in a Contract Workspace Template [page 39]

Adding a Contract Line Items Table Definition in a Contract Workspace Template [page 40]
Adding Sourcing Event Questions in a Table Within an Assembled Contract Document [page 41]
Adding a Contract Line Items Table Within an Assembled Contract Document [page 42]

Adding a Sourcing Event Questions Table Definition in a Contract Workspace Template

You can add a sourcing event questions table definition to an assembled contract document (main agreement or contract addendum) that is associated with a contract workspace template. This means that for any contract workspace that is based on this template, when a user adds sourcing event questions to the workspace those questions are added to the relevant contract document in table format automatically.

Context

> ⓘ Note
>
> For information about enablement and prerequisites for this functionality, see Embedment of CLIDs as Tables in Assembled Contract Documents [page 37].

Procedure

1. Open the contract workspace template and navigate to the **Documents** tab.
2. Click the document title of an assembled document and choose **View Details**.
3. Click the **Outline View** tab.
4. Do one of the following:
    - To add a table in the main body of the document, select **Actions** > **Add Table Definition**.
    - To add a table in a section within the main body of the document, select the section name and choose **Add Table Definition**.
5. Enter a name for the table.
6. Select **Sourcing Event Questions Table** in the **Table Type** menu.
7. Click **OK**.
   The table definition is added to the document outline view, where it is listed as a **Sourcing Event Questions Table** in the **Type** column.

Adding a Contract Line Items Table Definition in a Contract Workspace Template

You can add a contract line items table definition to an assembled contract document (main agreement or contract addendum) that is associated with a contract workspace template. This means that for any contract workspace that is based on this template, selected contract line items terms that are associated with the assembled document are added to the relevant contract document in table format automatically.

Context

> ⓘ Note
>
> For information about enablement and prerequisites for this functionality, see Embedment of CLIDs as Tables in Assembled Contract Documents [page 37].

You can add any number of contract line items table definitions within a document, each with different combinations of contract line items terms, if required.

> ⓘ Note
>
> Contract line items tables can only be added to contract line items documents for which the capacity type is standard (which is a maximum of 2000 line items). It does not apply to line items documents with capacity type large.
>
> The Contract Line Items Table option is only available when there is a contract line items document definition in the workspace template.

When contract line item documents (CLIDs) are embedded as tables in assembled contract documents (main agreement or contract addendum), in the Item Name column of the table, items are prefixed with a number that indicates the position of the item in the table. For example, item 1 is listed as "1. Item", item 2 is listed as "2. Item", and so forth.

Procedure

1. Open the contract workspace template and navigate to the **Documents** tab.
2. Click the document title of an assembled document and choose **View Details**.
3. Click the **Outline View** tab.
4. Do one of the following:
   - To add a table in the main body of the document, select **Actions** > **Add Table Definition**.
   - To add a table in a section within the main body of the document, select the section name and choose **Add Table Definition**.
5. Enter a name for the table.
6. Select **Contract Line Items Table** in the **Table Type** menu.

7. Select **Yes** in the **Include Description Column** field if you want to include the line item description as a column in the contract line items table definition.
8. Select a term that you want to add as a table column from the **Select Column Name** menu and click on the plus button.

   You can repeat this step to add any number of terms as table columns. You can drag and drop terms to change the position of columns in the table within the assembled document.
9. Click **OK**.
   The table definition is added to the document outline view, where it is listed as a **Contract Line Items Table** in the **Type** column.

Adding Sourcing Event Questions in a Table Within an Assembled Contract Document

When you add ad hoc sourcing event questions to a contract workspace you can add these questions in table format to an assembled contract document (main agreement or contract addendum) that is associated with the contract workspace.

Context

> ⓘ Note
>
> For information about enablement and prerequisites for this functionality, see Embedment of CLIDs as Tables in Assembled Contract Documents [page 37].

For information about adding ad hoc sourcing event questions to a contract workspace, see Adding Sourcing Event Questions to a Contract Workspace.

Procedure

1. Open the contract workspace and navigate to the **Documents** tab.
2. Click the document title of an assembled document and choose **View Details**.
3. Click the **Outline View** tab.
4. Do one of the following:
   - To add a table in the main body of the document, select **Actions** > **Add Table Definition**.
   - To add a table in a section within the main body of the document, select the section name and choose **Add Table Definition**.
5. Enter a name for the table.
6. Select **Sourcing Event Questions Table** in the **Table Type** menu.
7. Click **OK**.

The table definition is added to the document outline view, where it is listed as a **Sourcing Event Questions Table** in the **Type** column.

Results

When you open the contract document in Microsoft Word, the sourcing event questions are displayed in a table.

Adding a Contract Line Items Table Within an Assembled Contract Document

You can add contract line item terms in table format to an assembled contract document (main agreement or contract addendum) that is associated with a contract workspace.

Context

> Note
>
> For information about enablement and prerequisites for this functionality, see Embedment of CLIDs as Tables in Assembled Contract Documents [page 37].

You can add any number of contract line items tables within a document, each with different combinations of contract line item terms, if required.

> Note
>
> Contract line items tables can only be added to contract line items documents for which the capacity type is standard (which is a maximum of 2000 line items). It does not apply to line items documents with capacity type large.
>
> The Contract Line Items Table option is only available when there is a contract line items document in the contract workspace.

Procedure

1. Open the contract workspace and navigate to the **Documents** tab.
2. Click the document title of an assembled document and choose **View Details**.
3. Click the **Outline View** tab.
4. Do one of the following:
    - To add a table in the main body of the document, select **Actions** > **Add Table Definition**.

- To add a table in a section within the main body of the document, select the section name and choose **Add Table Definition**.

5. Enter a name for the table.
6. Select **Contract Line Items Table** in the **Table Type** menu.
7. Select **Yes** in the **Include Description Column** field if you want to include the line item description as a column in the contract line items table.
8. Select a term that you want to add as a table column from the **Select Column Name** menu and click on the plus button.

   You can repeat this step to add any number of terms as table columns. You can drag and drop terms to change the position of columns in the table within the assembled document.
9. Click **OK**.
   The table definition is added to the document outline view, where it is listed as a **Contract Line Items Table** in the **Type** column.

Results

When you open the contract document in Microsoft Word, the selected contract terms are displayed in a table.

In the Item Name column of the table, items are prefixed with a number that indicates the position of the item in the table. For example, item 1 is listed as "1. Item", item 2 is listed as "2. Item", and so forth.

Loading Assembled Contract Documents

Context

If you are loading a large document, SAP Ariba recommends that you do this at a time when you do not need to use your desktop for any other Microsoft related processes.

Procedure

1. Choose.
2. Navigate to the **Documents** tab of the project template or contract workspace.
3. Click **Actions** and select **Create** > **Main Agreement** or **Create** > **Contract Addendum**.
4. Select the Microsoft Word (`.docx`) file you want to use as a basis for the Main Agreement or Contract Addendum (browse for the file, enter the file path, or drop a file icon in the drag-and-drop box).
5. Click **Create**.

Results

SAP Ariba creates a copy of the document and loads it into the project. If the document has tracked changes, SAP Ariba accepts all changes. SAP Ariba creates sections and clauses based on the applied styles.

Clause Library

The Clause Library is the central repository for clauses. You use clauses in assembled contract documents (Main Agreement and Contract Addendum documents).

About the Clause Library

The Clause Library is intended to help you manage and reuse contract clauses. Before authoring contract documents with SAP Ariba Contracts you must populate the Clause Library with clauses to use in your company's contract documents. You can organize the clauses in the Clause Library in folders.

The Clause Library is a knowledge repository, and as such, is located in the Knowledge Areas. To add clauses to the clause library (or to edit anything in the clause library) you must be a member of Clause Library team, which is listed on the Clause Library Team tab.

Clause Types in the Clause Library

When you load clauses into the Clause Library, SAP Ariba Contracts prompts you to choose a clause type. Clause types help identify the use for each clause in a contract document. The following clause types are useful in most typical contract negotiations:

- Preferred: Use for clauses you prefer to use at the start of your supplier or customer negotiations.
- Alternate: Use if there is alternate, but equally correct language for a Preferred clause.
- Fallback: Use if a supplier or customer rejects the language in either a Preferred or Alternate clause.

There is no functional difference between these types of clauses. Use the types to help you organize the clauses in your library. Types are only important when substituting clauses in an assembled contract document. For example, you might choose to substitute a clause with another clause from the Clause Library, and see the choice of an Alternate or Fallback clause. The clause type is not visible when viewing a contract document in the outline view or as a Microsoft Word document.

Empty Clause

Each Clause Library contains a clause with the type Empty Clause. This clause is installed by default in **Others** > **Empty Clause** > **Empty Clause**. By default, clauses deleted by contract authors are replaced with the Empty Clause.

The Empty Clause contains the following default text:

```
Clause intentionally deleted.
```

If your Clause Library does not contain a published Empty Clause, you migth see unexpected behavior when contract authors delete clauses. For example, authors might be unable to delete a clause from the Outline View, or see that clauses deleted through Microsoft Word do not delete in the Outline View.

For more information on deleting clauses see Deleting Clauses [page 25].

> ⚠ Limitations
>
> Do not delete the default Empty Clause. Do not create additional clauses with the type Empty Clause.

Accessing the Clause Library

Procedure

If you are a member of any group on the **Team** tab of the Clause Library workspace, click the **Clause Library** link in the Common Actions area of the Dashboard.

If you are not a member of a Clause Library team, use the following procedure to access the Clause Library:

1. In the Search area on the Dashboard, click **Search**.

2. For **Project Type**, select **All**.
3. In the search field, type **Clause Library**, and click **Search**.
4. In search results, click the Clause Library, and select **Open**.
5. On the **Clause Library** page, click the Clause Library you want, and select **Action** > **Open**.

Creating Folders in the Clause Library

SAP Ariba recommends that you create folders in the Clause Library and use the folders to organize your clauses. The Clause Library folder structure is also used to determine which clauses are identified as alternate or fallback clauses for a given clause.

Prerequisites

You must be a member of the Clause Library workspace team.

> **⚠ Limitations**
>
> - Folder names cannot contain the following characters:
>   `\ / : ? " < > | # + % &`
> - Do not use long folder names in the Clause Library. Microsoft Word has a file and folder path limit of 255 characters. You cannot open the clause if the folder names exceed this limit.
> - For optimal performance when viewing or using clauses from the library, try to limit the number of clauses or subfolders in a folder to 50 or fewer.

Procedure

1. Navigate to the Clause Library. Click the **Clauses** tab if it is not already selected.
2. Choose **Actions** > **Create** > **Folder**.
3. Enter a name for the folder and click **OK**.

Adding an Individual Clause to the Clause Library

The simplest way to add clauses to the Clause Library is to add them one at a time using the user interface. This process is useful for adding only one or two clauses to the Clause Library.

Prerequisites

You must be a member of the Clause Library team.

Context

- You do not need to load every clause your company uses. Load only clauses that:
  - Contract authors will use more than once
  - You want to report usage data for (a clause must be in the Clause Library to provide usage data to reporting)

  If you are loading a clause that will have alternate clauses, create a folder for that clause and its alternatives. If you do this, users working on project documents that include a clause from the folder will see an **ALT** icon next to the clause that they can click to select alternative clauses (other clauses in the same folder).
  For more information on folders that contain multiple published clauses, see Designating Fallback and Alternate Clauses [page 55].
  SAP Ariba recommends that you give unique, short, descriptive names for each clause help you easily identify clauses and use them properly.
- When loading individual clauses to the Clause Library, the document name (excluding the `.docx` file extension) is used as the clause name. For example, if you load the document `WarrantyUS.docx` as a clause, the clause is named `WarrantyUS`.
  Documents loaded as individual clauses to the Clause Library cannot have the following special characters in their names:
  `\ / : ? “ < > | # + % &`

You can also add multiple clauses to the clause library from a single document [page 53].

Procedure

1. Create a document containing text for a single clause. Do not format the clause in any way. You do not need to remove background information such as headers and footers if they exist. SAP Ariba Contracts ignores all bookmarks in clauses you manually load.
2. Navigate to the Clause Library. Click the **Clauses** tab if it is not already selected.
3. (Optional) Create a folder to contain your clause as described in Creating Folders in the Clause Library [page 47].
4. Open the location (folder) to which you want to add the clause. Click **Actions** > **Create** > **Clause**. A **Create New Clause** page opens.

5. Select the file you want to load (browse for the file, enter the file path, or drop a file icon in the drag-and-drop box).

   Enter a description for the clause. This description will be available to contract authors when displaying a document in the outline view or when substituting clauses. The clause description text is not viewable by Suppliers, customers, or external users.
6. Select the clause type: **Preferred**, **Alternate**, or **Fallback.** See Designating Fallback and Alternate Clauses [page 55] for details.

   New clauses are in Draft status and are not immediately available for use in templates or as a substitute for clauses in existing contract workspaces. Also notice the description of the clause.
7. Click **OK** to load the document as a clause. SAP Ariba Contracts starts Microsoft Word on your client and opens the document to summarize the contents. Close the document and exit Microsoft Word.
8. Publish the clause to make it available for contract authors. On the **Clause Library** page, click the name of clause you loaded and choose **Action** > **Publish**.

About Mass Upload of Clauses to the Clause Library

Admin users in SAP Ariba Contracts can bulk upload the clauses and the associated content into a clause library via a document containing the clauses and the set of associated rules/information in an Excel document. Users can download the clauses and associated information from SAP Ariba thus allowing users to make the changes to the clauses and rules outside the SAP Ariba solution and then add the updated information back into the system.

You may have a pre-existing library of clauses with associated attributes and related information that you want to upload to your clause library folder within SAP Ariba. You can create a folder structure containing the clauses and an Excel document containing information about the clauses, and generate a .zip file. You can then import this .zip file toSAP Ariba Contracts.

You may also want to export your existing clause library content from SAP Ariba for making changes to it and later update the library in SAP Ariba Contracts. You can select multiple clauses from your clause library in SAP Ariba Contracts and export them to a .zip file that contains the folders and clauses that you exported along with an Excel document containing information about the relative path, description, owner, access, control, and other attributes of the clauses in your clause library.

> ⓘ Note
>
> For an empty folder, the attributes are exported in the Excel document but the folder is not created in the exported .zip file.

You can also update a set of clauses in your clause library by first exporting the clauses from the clause library to a .zip file, updating the clauses in your local system, and then importing a .zip file containing the updated clauses to SAP Ariba Contracts.

Prerequisites

This feature requires the "ActiveX Removal" feature. To enable the "ActiveX Removal" feature, have your Designated Support Contact log a service request. An SAP Ariba Customer Support representative will follow up to complete the request.

Format of the Excel Document for Mass Upload of Clauses

In the Excel document, the folders are listed before the clauses. For example, if folder A contains a clause Clause A and also contains folder B, which contains clause B, the Excel sheet lists the paths as follows:

- /A
- /A/B
- /A/B/Clause B.docx
- /A/Clause A.docx

> ⓘ Note
>
> The names of the folders and the clauses are case-sensitive. The **View Details** > **Use Requires Approval By** field specifies the users from whom approval is required, so even if your uploaded .zip files contain clause documents that have the status as Published (not Draft), they will generate approval tasks for the approvers specified in the **Use Requires Approval By** field. When updating a set of clauses, the **Base Language** attribute cannot be modified, and the sequence of clauses is not reset after every import.

Listing the folders in the Excel document is not required, but is optional. If they are not listed in the Excel document, but only as part of the clause name, the folders are dynamically created. The following is a description of the available fields in the Excel document:

- File path: Text. The path of the target location in the clause library for the clause or folder. It must match an existing file within the folder, clauselibrary, in the .zip file. When used in conjunction with the **Internal Id** field, this field is ignored.
- Description: Free text. The description of the clause, as it exists today, in the overview of the clause.
- Owner: The owner of the clause, as it exists today, in the overview of the clause. Must match the ID (unique name) of an existing user or system group.
- Access Control: The access control permissions of the clause, as they exist today, in the overview of the clause. To obtain a list of valid values for the access control, see About the PermissionBitVec element in contract exports in Contract Administration Guide. Multiple values can be specified, separated by a vertical bar (|).
- Base Language: The base language of the clause, as it exists today, in the overview of the clause. It is case-sensitive. To obtain a list of valid values for the **BaseLanguage** field, use the Base Language (CSV) export data task.
- Editors: The editors of the clause, as they exist today, in the overview of the clause. Must match the ID (unique name) of an existing user or system group. Multiple values can be specified, separated by a vertical bar (|).
- Clause Type: The type of clause. Must be one of Preferred, Fallback, or Alternative.
- Change Requires Approval By, Use Requires Approval By, and Delete Requires Approval By: The users or group of users who must approve the changes/usage/deletion of a clause, as it exists today, in the overview of the clause. Must match the ID (unique name) of an existing user or system group. Multiple values can be specified, separated by a vertical bar (|).

- Internal Id: Leave this field empty upon creation of new clauses. In case of update from a previously exported file, leave the exported ID, which will be used to identify the clause to upload (instead of the field, File Path), increasing the speed with which the file is processed.
- Update File: To update the content of a specified clause, set this field to **true**, and leave as **false** not to modify the clause content. This is useful if you want to update only the content of a few clauses after a mass export operation. Only the content of the clauses set to **true** will be updated. If the value of this field is **false**, only the clause attributes (but not the clause content) will be updated.
- Status: Status of the clause after the load operation. Must be one of Published or Draft.

Importing Clauses to the Clause Library [page 51]

Updating Existing Clauses in the Clause Library [page 52]

Exporting Clauses from the Clause Library [page 52]

Importing Clauses to the Clause Library

Context

You can import a set of clauses to the clause library.

Procedure

1. Create a set of files in Microsoft Word .docx format corresponding to the clauses, and organize the files in a folder or set of nested folders.
2. Use the clause library **Export** operation to create a `clauselibrary.zip` file with a `ClauseLibrary.xlsx` Excel document to use as a template. Alternatively, create the `ClauseLibrary.xlsx` file, and add a row with the column heads listed in the previous section.
3. In the `ClauseLibrary.xlsx` document, create a row for each folder and clause, and specify the path and description for the folder or clause, and details such as the path, base language, editors, and clause type for clauses.
4. Create a .zip file named `clauselibrary.zip` that contains the folders, clauses, and your `ClauseLibrary.xlsx` document. Note: The Excel document and the content to upload must reside in a folder called `clauselibrary` (case sensitive). That folder must be directly in the root of the .zip file.
5. Navigate to your clause library in SAP Ariba Contracts and select **Import**. Browse or drag-and-drop the updated `clauselibrary.zip` file to SAP Ariba Contracts.

   On importing, the folders, clause files, and items attributes are created or updated as indicated in the .zip file.
6. Verify that the folders, clause files, and items attributes as indicated in the .zip file are created or updated in the clause library. You can open a clause to view its details and verify that the attributes match the ones you specified in the Excel document for that clause.

Updating Existing Clauses in the Clause Library

Context

You can update existing clauses in the clause library.

Procedure

1. From your clause library, select the folders that need to be updated and export the`clauselibrary.zip` file by selecting **Export**.
2. Extract the contents of the .zip file.
3. To update the content of the clause(s), update the Word file(s) containing the clause(s), and edit the rows corresponding to the clause(s) in the `ClauseLibrary.xlsx` file to indicate any changes to the attributes. If you update the content of a clause, you must provide the value True in the `Update File` column of the row corresponding to that clause. If the value of the `Update File` column is False, only the clause attributes (but not the clause content) will be updated.
4. Create a new `clauselibrary.zip` file by including the folders, updated clauses, and the updated Excel document.
5. Navigate to your clause library in SAP Ariba Contracts and select Import. Browse or drag-and-drop the updated `clauselibrary.zip` file to SAP Ariba Contracts.

   On importing, the folders, clause files, and items attributes are updated as indicated in the ZIP file.
6. Verify that the folders, clause files, and item attributes as indicated in the .zip file are updated in the clause library. You can click a clause to view its details and verify that the attributes match the ones you specified in the Excel document for that clause.

   In the updated clause library, updates and additions of new clauses appear after the existing clauses.

Exporting Clauses from the Clause Library

Context

You can export the entire clause library or a subset of the clause library.

Procedure

1. Navigate to your clause library, select the clause(s) to be exported, and select **Export**.

   On exporting the clauses, a .zip file titled `clauselibrary.zip` is generated that contains the hierarchy of the clauses and folders selected for export, as well as an Excel document titled `ClauseLibrary.xlsx` that contains the attributes of all the folders and files included in the export.

   The .zip file contains the folder structure and the underlying clause files (in Microsoft Word .docx format) and serves as a template for uploading the changes back into the clause library using the import option. The Excel document contains the item attribute information for the folders and the clauses.
2. Unzip the .zip file to a folder on your system, and verify that the clauses you exported are present in the folder.

   The clauses that you exported are available in the folder to which you exported them.

Adding Multiple Clauses to the Clause Library from a Project Template Document

You can load multiple clauses to the Clause Library from a single document by first loading the document into a project template.

Prerequisites

- To use this method, users must load the contract document into a project template before loading the clauses into the Clause Library. Therefore, users must have the ability to add documents to a project template (such as members of the Template Creators group and members of the Templates team) to use this method.
- You must have group access to the Clause Library.

Context

You do not need to load every clause your company uses. Load only clauses that:

- Contract authors will use more than once
- You want to report usage data for (a clause must be in the Clause Library to provide usage data to reporting)

If you are loading a clause that will have alternate clauses, create a folder for that clause and its alternatives. If you do this, users working on project documents that include a clause from the folder will see an **ALT** icon next to the clause that they can click to select alternative clauses (other clauses in the same folder).

For more information on folders that contain multiple published clauses, see Designating Fallback and Alternate Clauses [page 55].

SAP Ariba recommends that you give unique, short, descriptive names for each clause help you easily identify clauses and use them properly.

You can also add an individual clause to the clause library [page 48].

> ⓘ Note
>
> SAP Ariba recommends removing the duplicates and limiting the clause and section count to less than 10,000 in the template to achieve optimum performance.

Procedure

1. Prepare a document to be loaded as a Main Agreement or Contract Addendum.
2. Open a project template. If the template status is not `Draft`, create a new version by clicking **Actions** > **New Version** from the Properties area.

   Load the document as a Main Agreement or Contract Addendum in the template.
3. Click the **Outline View** tab to view the clauses.
4. To view the document, click the Microsoft Word icon for a clause in the outline view.
5. Open the document in the outline view. For each clause you want to publish to the Clause Library, choose **Actions** > **Publish to Clause Library**.

   A **New Clause** window opens.
6. Enter information about the clause.

   Change the default title **(New Clause)** and create a useful title for the clause. Keep the title brief and descriptive to help contract authors identify clauses they want to use. The title displays in reports and search results.

   Title names cannot contain the following characters:

   `\ / : ? " < > | # + % &`

   Enter a description for the clause. This description will be available to contract authors when displaying a document in the outline view or when substituting clauses. The clause description text is not viewable by Suppliers, customers, or external users.

   Choose the clause type: **Preferred**, **Alternate**, or **Fallback.** See Designating Fallback and Alternate Clauses [page 55] for details.

   (Optional) Set the approval levels to add users to document approval flows when a clause is used or modified. See About Using Clause Level Approvals [page 55] for details.

   Specify the owner for the document and any editors. Editors can view and edit the clause.

   Specify any access control attributes.
7. Click **Save**.
8. (Optional) Clauses published to the Clause Library from a project template are added to the root level (top) of the Clause Library. To move the clause to a folder, drag and drop the clause to the appropriate folder. If you want to create a new folder, use the procedure described in Creating Folders in the Clause Library [page 47].

Designating Fallback and Alternate Clauses

Prerequisites

To make a clause available for use as a fallback or alternate for a clause in a contract workspace, the clause must:

- reside in the same folder in the Clause Library as the clause it can replace
- be published

Context

If contract document includes a clause from a Clause Library folder that contains multiple published clauses, users displaying the document in the outline view will see an **ALT** icon to the right of the clause. Clicking the **ALT** icon takes the user to the folder in the Clause Library that contains the clause. The **ALT** icon appears regardless of whether the other clauses in the folder in the Clause Library are Alternate, Fallback, or Preferred.

Procedure

- If you are loading a clause to the Clause Library, select the clause type in the **Clause Type** field.
- If the clause is already loaded:
    1. Navigate to the clause in the Clause Library.
    2. Click the clause name and choose **Action** > **Edit Attributes**.
    3. Select the clause type in the **Clause Type** field.
    4. Click **Save**.

About Using Clause Level Approvals

In SAP Ariba Contracts, it is often desirable to have a clause inclusion to trigger a specific approval. For example, if your business case requires a particular member of the legal team to be added to the contract workspace to approve a clause substitution, you can implement this using clause level approvals.

SAP Ariba provides the following clause level approval fields:

- **Changes Require Approval By**
- **Use Requires Approval By**
- **Delete Requires Approval By**

Adding a user or group to a clause level approval field results in different actions depending on whether the clause is in the Clause Library or in a contract document created within a contract project.

Clause Changes and Usage in Workspaces

If a clause is changed, used, or deleted in a contract document in a workspace, all users (or groups) specified in the corresponding clause level approval field are added as approvers to any approval task that encompasses that clause. The users are added in parallel to the entire approval flow. An approval task that encompasses a clause in a project can be at the following levels (from low to high):

- document: an approval task on the contract document that contains the clause
- folder: an approval task on a folder containing the contract document that contains the clause
- project: an approval task on an entire contract workspace that contains the contract document that contains the clause

The users specified in the clause level approval field are added to an existing approval task for the clause at the document, folder, or project level; if there are multiple tasks for the clause at different levels, the users are added to a task at the lowest level. If the existing approval task is in the Not Started state, the approval flow is modified. If the existing approval task is in the In Approval state, the task is withdrawn and a new approval round is started with the modified approval flow. If an approval task does not exist at the document, folder, or project level, an approval task is created for the document.

Approvers added to tasks in workspaces because of clause level approvals cannot be removed.

Related Information

About the Changes Require Approval by Field [page 56]
About the Use Requires Approval By Field [page 57]
About the Delete Requires Approval By Field [page 58]

About the Changes Require Approval by Field

Adding a user or group to the **Changes Require Approval By** field means different things depending on whether the clause is in the Clause Library or in a contract workspace created with a contract document that includes the clause.

Changes Require Approval by Field and Actions in the Clause Library and in Contract Workspaces

Adding a user or group to the **Changes Require Approval By** field means that every new version of that clause in the Clause Library must have approval before you publish the clause in the Clause Library. Approval is required if a user edits a clause in Microsoft Word, or edits it manually through the user interface.

Deletion of a clause does not trigger approval of the clause in the Clause Library.

Clauses are approved by either adding the users specified in the **Changes Require Approval By** field to an existing clause level approval task or by creating a new approval task for the clause if one does not exist.

Keep in mind the following when using the **Changes Require Approval By** field for clauses modified from the Clause Library:

- The first time a clause is edited, if there is not already an approval task defined that includes that clause, a new approval task is created on the **Tasks** tab of the Clause Library.
- If a user is specified in the **Changes Require Approval By** field for a clause, the clause cannot be published for use until the approval task on the **Tasks** tab of the Clause Library workspace has been completed.
- Any approval task created for clause level approval in the Clause Library is required.
- There is no limit to the number of users, either directly or through groups, that can be added to the **Changes Require Approval By** field on a clause.
- A user is never added more than once to an approval task no matter how many clauses that user is associated with.
  For contract workspaces, users or groups specified in the **Changes Require Approval By** field must approve the following actions for that clause in the workspace:

- The clause is manually edited by any user in Microsoft Word
- The clause is manually edited through the Outline View

Users and groups specified in the **Changes Require Approval By** field are added to an existing approval task that encompasses the clause or added to a new approval task, as described in Clause Changes and Usage in Workspaces [page 56]. A user is never added more than once to an approval task no matter how many clauses that user is associated with.

About the Use Requires Approval By Field

Adding a user or group to the **Use Requires Approval By** field has different implications depending on whether the clause is in the Clause Library or in a contract workspace created with a contract document that includes the clause.

Use Requires Approval By Field and Actions in the Clause Library and in the Contract Workspace Documents

Adding a user or groups to the **Use Requires Approval** By field on a clause has no effect when working on a clause in the Clause Library.

Adding users or groups to the **Use Requires Approval By** field on a clause has the same effect in a contract workspace as adding a user or group to the **Changes Require Approval By** field for a clause except the approval is triggered by a substitution instead of a change.

Users or groups in the **Use Requires Approval By** field must approve the following actions for the clause in a workspace:

- The clause is added as a new clause to a contract document in a workspace

- The clause is added as a substitute for another clause in a contract document in a workspace

The **Use Requires Approval By** field has no effect if a clause is included in a contract document within a contract workspace by default (the clause is inherited from the template document).

Users and groups specified in the **Use Requires Approval By** field are added to an existing approval task that encompasses the clause or added to a new approval task as described in Clause Changes and Usage in Workspaces [page 56].

About the Delete Requires Approval By Field

Adding a user or group to the **Delete Requires Approval By** field has different implications depending on whether the clause is in the Clause Library or in a contract workspace created with a contract document that includes the clause.

Delete Requires Approval By Field and Actions in the Clause Library and in the Contract Workspaces

Adding a user or groups to the **Delete Requires Approval By** field on a clause has no effect on clauses deleted from the Clause Library.

Users or groups in the **Delete Requires Approval By** field must approve the deletion of the clause in a workspace document. Users and groups specified in the **Delete Requires Approval By** field are added to an existing approval task that encompasses the clause or added to a new approval task as described in Clause Changes and Usage in Workspaces [page 56].

Updating Clauses in the Clause Library

You can update the text of a clause in the Clause Library. If the clause is included in a contract document in a project template, new projects created from the template will include the updated clause text. The change is not propagated to documents in existing projects.

Procedure

1. Locate the clause in the Clause Library.
2. Click the clause name and choose **Actions** > **Download**.

   Open the document on your system using Microsoft Word or any other OpenXML editor.
3. Edit the text. Save the document and exit your editor.

4. Drag and drop the updated document into the drag-and-drop area in the top left corner of the Documents screen or upload it into the application.

Results

- If you update a clause in the Clause Library and publish it, all project templates that contain assembled contract documents with that clause automatically use the new version for all future contract workspaces created with that template.
- When you update clauses in the Clause Library, the system does not update clauses in existing workspaces, including clauses inherited from a template document. When a workspace is created, the system creates a copy of the clause text from the Clause Library. The system never updates this copied text after the workspace is created.
- When you view the project template, the clause shown on the **Outline View** tab of the contract document remains as the old version (despite the fact that the newer clause appears when you create a contract workspace) until you create a new version of the template. The type will be `Library (Deprecated)` to indicate that the clause text shown is not the latest version of the clause. To view the newest clause version in the template and to change the type to `Library`, you can publish a new version of the project template.

Deleting and Inactivating Clauses in the Clause Library

If you inactivate or delete a clause from the clause library, the clause remains in project template and workspace documents that have already been created, but users cannot add the clause to new documents.

Context

The differences between inactivating and deleting a clause from the clause library are:

- A clause must be published to inactivate it.
- You can reactivate an inactive clause, but you cannot restore a deleted clause.
- If you delete a clause from the clause library, the system no longer lists the clause in the library. If you inactivate a clause, the system continues to list the clause in the library with the type `Template (Inactive)`.

Procedure

1. Locate the clause in the Clause Library.
2. Click the clause name and choose **Actions** > **Delete** or **Actions** > **Inactivate**.

Results

If you inactivate or delete a clause from the clause library:

- Users cannot add the clause to a document in a template or workspace or use the clause as a substitute in a document.
- The clause remains in template and workspace documents that have already been created.
- The system will include the clause in documents for new projects created from a published template that already includes the inactive or deleted clause. If you want to exclude the inactive or deleted clause from future workspaces created from the template, create a new version of the template, remove the clause from the document in the template, and publish a new template version.
- The outline view of documents in projects that use the inactive or deleted clause have the type `Template (Deprecated)` to indicate that the clause has been inactivated or deleted from the clause library. In templates, the type will be `Library (Deprecated)`.

Searching for Clauses

Searching for clauses works similarly to searches for other documents. You can search within folders, in contract workspaces, or just in the Clause Library. Search results allow you to see which templates and documents use specific clauses.

Prerequisites

You must be a member of the Clause Library team.

Procedure

1. Navigate to the Clause Library.
2. Under the **Clauses** tab, click **Actions** > **Search**.

   A **Search** page opens.
3. Enter search criteria, such as title words or keywords.
4. Click **Search**.

Results

The search results list the clauses that matched the search criteria with the following columns:

- Title: The clause title.
- Location: The folder path to the clause in the Clause Library.
- Project: Location of the clause (the Clause Library).
- Clause Type: The clause type (Preferred, Alternate, Fallback).
- Summary: The first 200 characters of the clause text to help in identification.

Finding Where a Clause Is Used

You can search for clauses where they have been used. This enables you to see which templates and contract workspaces use a specific clause, including the section where it is used.

Prerequisites

You must be a member of the Clause Library team.

Procedure

1. Navigate to the Clause Library.
2. Click the clause name and choose **Action** > **Search Where Used** on the **Clauses** tab.

   A **Search** page opens.
3. Specify search options. By default, the search displays only projects where the clause is used, but you can widen the search to include templates by selecting **Templates** for the **Display only** option.

   You can also enter project title words or keywords to narrow the search results.
4. Click **Search**.

About Exporting and Importing the Clause Library

If you are the Clause Library Project Owner, or if you are on the Clause Library team and a member of the Contract Administrator group, you can export the Clause Library to a ZIP file. Clause Library export and import allows you to use a Clause Library that has been developed in another environment without having to manually re-create it. For example, you can develop your Clause Library in a test environment, export it, and then import it into a production environment when it is complete. You can also export the Clause Library from your production environment and import it into your test environment to test changes.

You can perform a full export or import of all of the clauses in the Clause Library, or a partial export or import of selected folders. You can only export or import clauses in folders; you cannot export or import individual clauses, although you can move them manually.

See Maintaining Links Between Templates and Clauses [page 62] for details on maintaining the relationship between clauses and templates during Clause Library import and export.

When you export and import the Clause Library, keep in mind the following:

- Folders and clause documents are exported and imported in the order in which they appear in the source Clause Library. If you re-order content in the destination Clause Library, and then re-import the source Clause Library, you will lose that new ordering.
- SAP Ariba Contracts does not export or import the translations of folders and clauses during Clause Library export and import.
- Be sure to save all clause document changes before attempting to export or import clauses. During Clause Library export and import, you will not be notified if you have unsaved changes to clause documents, and those unsaved changes will be lost. Clause Library export and import is intended to help you populate your destination Clause Library; it is not designed for managing changes to individual clauses.

Link Maintenance Between Templates and Clauses

Since contract workspace templates use clauses, there is a relationship between the template and the clause in the Clause Library. You can maintain this relationship during Clause Library export and import by following these guidelines:

- Export the Clause Library from the source environment and import it to the destination environment before you export and import the associated templates. Otherwise, links between templates and clauses are lost.
- Perform a full export from the source environment and a full import into the destination environment the first time you use Clause Library export and import.
  Performing a full export and import of the Clause Library ensures that the entire clause folder structure is imported into the new environment. This is important because if you perform a partial export and import and miss one or more folders, then import the applicable template, any links between the template and clauses in the missing folder or folders will be lost. If those links are lost, you must manually reconstitute them by removing each clause from the template and then adding each clause back to the template. Exporting and importing the entire clause folder structure prevents the unintentional loss of links between templates and clauses.
- Perform partial exports and imports for subsequent updates to your destination Clause Library.
- Import clauses with Published status. Clauses must have this status to be used in a template. Links between templates and clauses imported in Draft status are lost.
- Make sure that all in-process clause changes in the source environment are completed before Clause Library export. In-process clause updates that have not been completed are not exported.

Exporting the Clause Library

Context

You can perform a full library export, or a partial export of selected top-level folders. You export only the latest version of the clauses; any previous versions that exist in history are not exported.

When you export all or part of a Clause Library, individual clauses are copied into XML files in a ZIP file. The ZIP file preserves the folder hierarchy that exists in the source Clause Library.

> ⚠ Limitations
>
> Your Clause Library might include ZIP files that contain, for example, supporting documents. Since Clause Library export creates a ZIP file that contains the entire Clause Library, it cannot contain nested ZIP files of Clause Library content. Attempting to export and import a Clause Library that contains ZIP files will result in empty ZIP files. To preserve the contents of ZIP files in the Clause Library, temporarily move the files to your local disk. You can later manually upload the ZIP files after you have completed the Clause Library import to the destination environment.

Procedure

1. Navigate to the **Clauses** tab on the Clause Library page.
2. Do one of the following to begin export of either the full Clause Library or selected folders:
   - To export the full Clause Library, click **Actions** and choose **Export** > **Clause Library**.
   - To export a top-level folder and its contents, click the folder and choose **Export** > **This folder and content**. The folder itself and all of its contents are exported in a ZIP file.
3. Click the **Include Supporting Documents** check box to include supporting documents in the export.
4. Click **Export**.
5. Click **Save** and save the exported ZIP file to the location of your choice.

Results

By default, an exported ZIP file for the entire Clause Library is named `ClauseLibrary.zip`, while a ZIP file for a folder export is named for the folder, for example, `ServiceAgreements .zip`. You can rename these ZIP files as needed.

Importing the Clause Library

Context

You can perform a full library import, or a partial import of selected folders. The import ZIP file retains the folder structure that existed when the Clause Library content was exported from the source Clause Library, and imports clauses in that structure. Supporting documents retain their original status when you import them into the Clause Library.

You might be importing clauses into an empty Clause Library. However, if you are importing clauses into a Clause Library with existing content, it is important to understand how imported clauses interact with existing clauses in the destination library:

- Clause Library import replaces any clause that has been locked by its owner.
- Clause Library import replaces any clause that has an Approval task in a pending state, and issues a warning for every attempt to replace a clause with a non-approved Approval task during import.
- Existing clauses that are replaced during Clause Library import are updated with a new version. This new version indicates that it was created from a bulk import, has the status specified at import (Published, Draft, or the status exported from the source Clause Library), and maintains any existing links to templates and supporting documents.
- If you import a new version of an existing folder, you must import it to the same location in the folder structure. For example, if you are importing a new version of the Agreement folder, which is currently located in Clause Library > Service Agreements, you cannot import it into the Clause Library folder. You must import it into the same folder, Clause Library > Service Agreements.

Procedure

1. Navigate to the **Clauses** tab on the Clause Library page.
2. Do one of the following to begin import of either the full Clause Library or selected folders:
    - To import the full Clause Library, click **Actions** and choose **Import** > **Clause Library content**.
    - To import a folder and its contents, click the folder into which you want to import it and choose **Import** > **Clause Library content**.
3. Click the appropriate radio button to import the clauses in Published status, Draft status, or the status in which they were exported.

    If the imported clauses are used in templates, import them in Published status so that they retain their links to templates. If you do not import them in Published status, you will need to reconstitute the links between clauses and templates by manually removing the clauses from the template and re-adding them.
4. Click **Browse** and navigate to the location of the Clause Library ZIP file you want to import.
5. Click **Import**.

Clause Library

The Clause Library is modeled like a contract workspace project, and contains **Overview**, **Clauses**, **Tasks**, and **Team** tabs. The **Tasks** and **Teams** tabs function similarly to these tabs in contract workspaces. See the "Creating and Managing Projects" guide for details on these tabs.

Overview Tab in the Clause Library

The **Overview** tab displays information that applies to the entire Clause Library. This tab is divided into the following areas:

- **Overview**
- **Process**
- **Quick Links**
- **Announcements**
- **Overview** area: In this area you see the project state, version, description, and other information about the project. The following table lists the tasks available from the **Actions** menu in this area.

To Do This...	Choose...
Edit overview information	**Actions** > **Edit Overview** to edit the overview information.
View details for the Clause Library	**Actions** > **View Details**. This information displays on three tabs: Overview, Documents, and Version History. Click the **Overview** tab to view the project states, status, and other details. You can change this information by clicking **Actions** > **Edit Attribute**s. You can also lock and unlock documents, or save a new version. Click the **Documents** tab to view the clauses in the Clause Library. Click the **Version History** tab to view the version history for the Clause Library.
Display the Clause Library in compact or full view	**Actions** > **Full View** or **Compact View**. Full View displays the Clauses, Tasks, and Team tabs for the Clause Library. Compact View displays the Clause Library without tabs.
Add the Clause Library to a list of projects that might need your attention	**Actions** > **Add to Watched Projects**. Lists the Clause Library on the Watched Projects content area on the dashboard.

- **Process** area: This area lists task phases, follow-on or predecessor projects, and milestones to provide visibility for these items. Click **All Tasks** to access the **Tasks** tab.
- **Quick Links**: This area lists documents that you want to access quickly. On the **Documents** tab, create a Quick Link by clicking a document name and choosing **Open** > **Add to Quick Links**.
- **Announcements**: This area displays the announcements pertaining to the Clause Library. These are intended for the entire contract workspace team. Click **New** to add a new announcement, or **Details** to view, edit, or delete announcement details.

Clauses Tab in the Clause Library

The **Clauses** tab lists all of the clauses in the Clause Library. You can expand clause folders to access the individual clauses. Since all clauses in a folder are automatically designated as alternates of each other, the way you organize your clauses in folders affects how they are used. See Accessing the Clause Library [page 46] for more details.

The following table lists the actions you can perform from the **Clauses** tab. The actions you see might vary depending on the status of the clause folders and clauses.

To Do This...	Choose...
View details for a clause folder	The folder name and choose **Action** > **Open** > **View Details**. The details are displayed in the Overview, Documents (lists the clauses in the folder), and Version History tabs.
Search for clauses in a folder	The folder name and choose **Action** > **Search** to search for clauses in a folder.
Edit folder attributes	The folder name and choose **Action** > **Edit Attributes.** This allows you to edit the title, description, owner, and access control and choose to save the folder to a new version, or retain the original version. If it is important for you to restrict access between users who can create sales contract workspaces and procurement contract workspaces, then you can apply these access controls to folders containing the clauses or to the specific clauses: • **Private to Procurement Contract Users** • **Private to Sales Contract Users**
Move a clause folder	The folder name and choose **Action** > **Move** to move a clause folder.
Add a clause folder to Quick Links	The folder name and choose **Action** > **Add to Quick Links.**
Lock or unlock a clause folder	The folder name and choose **Action** > **Lock Documents** or **Unlock Documents.** This prevents others from editing the clause folder and the clauses in it until you unlock it.
Save a clause folder to a new version	The folder name and choose **Action** > **Save as New Version.**
Create tasks for a clause folder	The folder name and choose **Action** > **Create New Task** > **To Do**, **Review**, **Approval**, or **Negotiation.** You can also create these tasks for individual clauses.
Upload a supporting document to a clause folder	The folder name and choose **Action** > **Upload** > **Supporting Document.** This allows you to upload a non-clause document pertaining to the folder. A supporting document in the Clause Library can help explain the clauses around it, provide helpful information or notes to clause authors, or provide process documents for clause approval. Supporting documents are only accessible by and visible to users with access to the Clause Library.
Export clauses to a Microsoft Excel file.	The folder name and **Action** > **Export**.
Import clauses from a Microsoft Excel file.	The folder name and **Action** > **Import**.
Create a subfolder	The folder name and choose **Action** > **Create Folder**. This allows you to create a subfolder in a clause folder.

To Do This...	Choose...
Create an analytical report on your clauses	The folder name and choose **Action** > **Analytical Report**. See Running Analytical Reports for detail on reporting.
Provide a Microsoft Excel template	the folder name and choose **Action** > **Create** > **Reporting Excel Template.**
Create a compound report on your clauses	The folder name and choose **Action** > **Compound Report**. See Creating Compound Reports for detail on reporting.
Create a new clause	The folder name and choose **Action** > **Create** > **Clause.** See Accessing the Clause Library [page 46] for details.
Open a draft of a clause	The clause name and choose **Action** > **Open Draft**.
Publish a clause	The clause name and choose **Action** > **Publish.** After you publish a clause, contract authors can use it in assembles documents in contract workspaces.
Open the last published version of a clause	The clause name and choose **Action** > **Download Last Published** to open the last published version of the clause in Microsoft Word.
View details of a clause	The clause name and choose **Action** > **View Details**. The details are displayed in the Overview and Version History tabs.
Search for instances of a specific clause in documents	The clause name and choose **Action** > **Search Where Used**, which allows you to find where the clause is used. This search returns a list of documents that contain the clause, including the section in the document, its title, the project associated with the document, the clause type, and a summary of the contents of the clause. See About Using Clause Level Approvals [page 55] for details on searching.
Edit clause attributes	The clause name and choose **Action** > **Edit Attributes.** This allows you to edit the title, description, owner, and access control, approvers and choose to save the clause to a new version, or retain the original version. If it is important for you to restrict access between users who can create sales contract workspaces and those who create procurement contract workspaces, then you can apply these access controls to specific clauses: • Private to Procurement Contract Users • Private to Sales Contract Users See About Using Clause Level Approvals [page 55] for details on clause approval.
Copy a clause	The clause name and choose **Action** > **Copy.** Enter a new name for the copy of the clause.
Move a clause	The clause name and choose **Action** > **Move.**
Add a clause to Quick Links	The clause name and choose **Action** > **Add to Quick Links.**
Lock your clause so no one else can change it	The clause name and choose **Action** > **Lock**. You can later unlock the clause.
Remove a clause from use in the clause library	The clause name and choose **Action** > **Inactivate**. The clause library shows the status as Inactive. You can later choose **Action** > **Reactivate** to use the clause again. If the clause is currently in use in any template or workspace documents, the clause remains in the document. The outline view of the document in workspaces shows the clause type as `Template (Deprecated)`; in templates, the type will be `Library (Deprecated)`.

To Do This...	Choose...
Delete a clause	The clause name and choose **Action** > **Delete.** The clause will no longer be listed in the clause library. If the clause is currently in use in any template or workspace documents, the clause remains in the document. The outline view of the document in workspaces shows the clause type as `Template (Deprecated)`; in templates, the type will be `Library (Deprecated)`.
Delete an old version of a clause	The clause name and choose **Action** > **Delete Old Versions.**
Replace a clause	The clause name and choose **Action** > **Replace Document.**
Merge a clause	The clause name and choose **Action** > **Merge Document.**

Document Properties

About Document Properties

This chapter describes how to use contract attributes as document properties in assembled contract documents (Main Agreement and Contract Addendum documents). Document properties contain data that can be used as fields to provide dynamic text in a Microsoft Word document.

For example, you can configure the Supplier Name contract attribute as a document property. Next, you insert this document property as a field (a document property field) in the Main Agreement document. When the Supplier Name attribute is set or changed in the contract workspace, the system updates the Main Agreement with the new value for the Supplier Name when it next generates the document.

You can use document properties in a Main Agreement or Contract Addendum contained in a template or in a contract workspace. In most cases, template authors set up document properties in template documents, rather than having contract authors or contract project owners set document properties in workspace documents. However, in cases where users add contract addenda in a workspace or supplier paper is used, users might want to configure document properties for workspace documents.

You can create document versions with literal text in place of document property fields. This is useful when sending a document to external users for review and can be set up as an automatic process when creating a review task in a template. You can also manually create a clean version by selecting **Actions** > **Generate Clean Version** from the Outline View tab for a document.

SAP Ariba supports two types of document properties for contract attributes:

- Read-only properties
  The system only reads the contract attribute values to set the document property values. The contract attribute values are not affected by any document changes.
- Editable properties

The system can read and modify the contract attribute values using the document property values. Changing the field value in the document also changes the corresponding contract attribute value.

Read Only Document Properties

By default, you can configure the following contract attributes as read-only document properties:

- Contract ID
- Contract Status
- Contract Title
- Commodity
- Region
- Client
- Buyer Name
- Buyer Address - Lines (the street address lines)
- Buyer Address - City
- Buyer Address - State
- Buyer Address - Postal Code
- Buyer Address - Country/Region Name
- Supplier Name
- Supplier Address - Lines (the street address lines)
- Supplier Address - City
- Supplier Address - State
- Supplier Address - Postal Code
- Supplier Address - Country/Region Name

You can insert a read-only document property field multiple times in the same document. For example, you can insert the document property field for the Supplier Name multiple times in a Main Agreement.

Editable Document Properties

By default, you can configure the following contract attributes as editable document properties:

- Agreement Date
- Effective Date
- Expiration Date

Editable document properties are also known as **Form Fields** in Microsoft Word. Editable document properties also have the following characteristics:

- If an editable document property (form field) is defined for a particular document, there must be one and only one instance of it in the document.
- If you define an editable document property for a particular contract document and the field has not already been inserted into the document, the field is inserted automatically at the end of the document, immediately

after the sectionGlobalContract bookmark. You must manually move the field to the appropriate location, as described in Viewing Editable Document Property Fields [page 77].

Document Properties Workflow

1. Set up the document properties. There are two ways you can do this:
   - By using the SAP Ariba Contracts interface to select the contract attribute fields. No typing is required, but you cannot set an initial or placeholder value to appear in the document before the attribute value is set. For more information, see Setting Up Document Properties Using the SAP Ariba Contracts Interface [page 71].
   - By using the Microsoft Word Properties pane. You must manually type the correct document property name. This method enables you to set an initial or placeholder value to appear in the document before the attribute value is set. This feature is useful when adding document properties to a template document to help you identify document property fields before a workspace is created (attribute values are typically set in workspaces but not in templates).
     For more information, see Setting Up Document Properties Using Microsoft Word [page 72].

     > ⚠ Limitations
     >
     > SAP Ariba recommends that you set up the document properties using either Microsoft Word or the Contracts Solution interface, but not both. Either method is acceptable and you might find one easier than the other, depending on your personal preferences.
2. Insert a field for the document property in the Main Agreement or Contract Addendum. For more information, see Inserting Fields for Document Properties in Text [page 74].
3. If you are not using enhanced contract authoring, generate a new version of the Main Agreement or Contract Addendum.
4. (Optional) "Freeze" the document property text.
   By default, changing the contract attribute value for the document property causes the system to update the document with the new value when it next generates the document. If you no longer want the system to automatically update a document with new document property values (you want to “freeze” the text), you can finalize the document. For information about finalizing documents, see Making Assembled Documents Final [page 36].

Setting Up Document Properties Using the SAP Ariba Contracts Interface

Context

SAP Ariba recommends that you set up the document properties using either Microsoft Word or the SAP Ariba Contracts interface, but not both. Either method is acceptable, and you might find one easier than the other, depending on your personal preferences.

After you set up document properties, you insert the fields for the properties in a Main Agreement or Contract Addendum. See Inserting Fields for Document Properties in Text [page 74].

Procedure

1. In the contract template or workspace, click View Details for the contract document for which you want to set up document properties.
2. Assemble the document if the state is Not Assembled. The state must be `Assembled` to make the **Set up Document Properties** option available in the **Actions** menu.
3. On the Outline View tab for the contract document, click **Actions** > **Set up Document Properties**.
4. On the **Set Up Document Properties** page, specify the contract attribute fields you want to use as document properties.

   The screen is split into **Read Only** and **Editable** sections. Select the fields you want to use in the document and use the arrow keys to move them under the **Available for use in Microsoft Word** area.

Results

The **Available for use in Microsoft Word** area lists each selected field name followed by the corresponding Microsoft Word property name in parentheses. For example, `Supplier Name (caSupplier)`. In this example, `caSupplier` is the Microsoft Word property name.

Setting Up Document Properties Using Microsoft Word

Context

SAP Ariba recommends that you set up the document properties using either Microsoft Word or the SAP Ariba Contracts interface, but not both. Either method is acceptable, and you might find one easier than the other, depending on your personal preferences.

After you set up document properties, you insert the fields for the properties in a Main Agreement or Contract Addendum. See Inserting Fields for Document Properties in Text [page 74].

Procedure

1. Open the Main Agreement or Contract Addendum using Microsoft Word. One way to do this is by navigating to the document in the Contracts Solution interface and clicking the Microsoft Word icon.

2. Make sure the document has been marked Unprotected in Microsoft Word.

   When a document is marked unprotected, the **Tools** menu displays the option **Protect Document**. If a document is marked protected, mark it unprotected by clicking **Tools** > **Unprotected Document**.
3. Click **File** > **Properties**.
4. In the **Properties** window that appears, navigate to the **Custom** tab.
5. In the **Name** field, enter the document property name that corresponds to the contract attribute (CA) you want to use in the Microsoft Word file.

   Read-only document property names have the following format:

   `ca`*`fieldname`*

   Editable document property names have the following format:

   `_ca`*`fieldname`*

   Where *`fieldname`* is the field name.

   The following tables list the contract attributes available by default and the corresponding read-only or editable document property names:

Table 1: Read-Only Document Properties

To Use This Contract Attribute...	Enter This Name for the Read-Only Document Property...
Contract Status	`caStatus`
Contract Title	`caTitle`
Commodity	`caCommodity`
Region	`caRegion`
Client	`caClient`
Buyer Name	`caBuyer`
Buyer Address - Lines	`caBuyer.CorporateAddress.PostalAddress.Lines`
Buyer Address - City	`caBuyer.CorporateAddress.PostalAddress.City`
Buyer Address - State	`caBuyer.CorporateAddress.PostalAddress.State`
Buyer Address - Postal Code	`caBuyer.CorporateAddress.PostalAddress.PostalCode`
Buyer Address - Country/Region Name	`caBuyer.CorporateAddress.PostalAddress.Country.Name`
Supplier Name	`caSupplier`
Supplier Address - Lines	`caSupplier.CorporateAddress.PostalAddress.Lines`
Supplier Address - City	`caSupplier.CorporateAddress.PostalAddress.City`
Supplier Address - State	`caSupplier.CorporateAddress.PostalAddress.State`
Supplier Address - Postal Code	`caSupplier.CorporateAddress.PostalAddress.PostalCode`
Supplier Address - Country/Region Name	`caSupplier.CorporateAddress.PostalAddress.Country.Name`

Table 2: Editable Document Properties

To Use This Contract Attribute	Enter This Name for the Editable Document Property
Agreement Date	`_caAgreementDate`
Effective Date	`_caEffectiveDate`
Expiration Date	`_caExpirationDate`

6. Enter a starting value in the **Value** field. This value appears in the document if the corresponding contract attribute value is not set. If you do not have a starting value, type spaces or other placeholder characters in this field.
7. Click **Add** to add the document property to the **Properties** list.
8. Click **OK** to close the **Properties** window.
9. Save the Microsoft Word file.

Inserting Fields for Document Properties in Text

Prerequisites

Before you can insert a field for a document property in the text of an assembled contract document (Main Agreement or Contract Addendum), you must set up a document properties for a contract attributes as described in Setting Up Document Properties Using Ariba Contracts Solution [page 71] or Setting Up Document Properties Using Microsoft Word [page 72].

After you have inserted fields for document properties, you may need to generate or regenerate the document if you are not using enhanced contract authoring. If you are not using enhanced contract authoring, generate a new version of the Main Agreement or Contract Addendum.

> ⚠ Limitations
>
> - You can insert an editable document property only one time in a given document.
> - You insert a read-only document property as many times as you like inside a given document. For example, to use the Supplier name field in multiple clauses throughout the main agreement, as well as the header, insert the field in multiple places.

Procedure

1. Open the Main Agreement or Contract Addendum using Microsoft Word. One way to do this is by navigating to the document in the Contracts Solution interface and clicking the Microsoft Word icon.
2. (Optional) Set your Microsoft Word options to display fields as shaded text. By default, Microsoft Word does not highlight or shade read-only document property fields. You might want to set your options to always show fields with shading to flag these fields.

Click ▶ **Tools** ▶ **Options** ▶ and select the **View** tab. Select **Always** in the **Field shading** menu. Click **OK** to close the **Options** pane.

3. Position the cursor at the place in the Microsoft Word contract document where you want to insert the field for the document property.
4. Choose ▶ **Insert** ▶ **Field** ▶.
5. In the **Field** window, set **Categories** to **(All)**.

   Highlight **DocProperty** in the **Field Names** chooser.

   Select the property you want to insert into the contract document, such as `caSupplier` for the `Supplier Name` field. Click **OK** to insert the property.

Adding a Contract Term as a Document Property Field in Microsoft Word

Context

If your site has the SAP Ariba Strategic Sourcing Suite, it is possible to create a contract terms library by associating sourcing event questions and contract terms with categories in a category attribute hierarchy and uploading the hierarchy file to SAP Ariba. For information about setting up questions as contract terms in a category attribute hierarchy, see Contract Terms Support in Category Attribute Hierarchy.

When these questions and terms are tagged as contract terms in the category attribute hierarchy XML, the contract terms can be added as document properties to contract master agreements and contract addendum documents. For example, you might have a question like "When does your insurance certificate expire?" in your category attribute hierarchy. If tagged as a contract term, the question, along with the supplier response from the sourcing event, is copied to the contract workspace when one is created. In the contract workspace or in the contract workspace template, you can add this contract term to the contract master agreement or a contract addendum document as a document property. You can then open the MS Word file for the contract document from within a contract workspace or contract workspace template and add this document property as a variable field the Microsoft Word file. So, in the Word file you might have a clause like "The contract will be fulfilled before your insurance certificate expires on ...." In this example, you would add the variable field after "on" in the clause and the expiry date provided by the supplier would be added to the Word document automatically.

In the above example, if you edit the expiry date in the **Contract Terms** area on the contract workspace, the value is updated automatically in the MS Word file.

Procedure

1. In the contract workspace or contract workspace template, click the document name and select **View Details**.
2. On the **Outline View** tab for the contract document, click ▶ **Actions** ▶ **Set up Document Properties** ▶.
3. On the Set Up Document Properties page, in the **Contract Term Document Properties for use in Microsoft Word contract documents area**, select the contract terms you want to use in the document and use the arrow keys to move them to the **Available for use in Microsoft Word** area. The selected contract

term short label is listed with the corresponding Microsoft Word property name. Contract term document properties have a "ct_" prefix and the external ID of the contract term is included as the suffix, for example, "ct_DeliveryTemperature_011".

4. Open the contract document in Microsoft Word. You can do this by double clicking the Microsoft Word icon next to any clause.
5. Position the cursor at the place in the Microsoft Word contract document you want to insert the contract term field for the document property.
6. In Microsoft Word, choose **Insert** > **Quick Parts** > **Field**.
7. In the **Field** window, highlight **DocProperty** in the Field names list.
8. Select the field you want to insert into the contract document. Properties that use contract attribute fields have the prefix ct, such as ct_DeliveryTemperature_011.
9. Click **OK** to insert the property.
10. Generate or regenerate the document.

Example: Adding the Supplier as a Document Property Field

1. Click the document name and select **View Details**.
2. On the Outline View tab for the contract document, click **Actions** > **Set up Document Properties**.
3. On the **Set Up Document Properties** page, select the fields you want to use in the document and use the arrow keys to move them to the **Available for use in Microsoft Word** area. For example, the user selected the read-only field Supplier Name. The selected field name is listed with the corresponding Microsoft Word property name (`caSupplier` in this example).
4. Open the contract document in Microsoft Word. You can download the contract document in Microsoft Word in one of the following ways:
   - Click the Microsoft Word icon or the document link in the **Document** pane at the upper-left corner of the page.
   - Click **Done** to go back to the **Documents** tab and then click the Microsoft Word icon next to the document or select the **Download Draft** option from the context menu of the document.
5. In Microsoft Word, choose **File** > **Properties** . Navigate to the **Custom** tab.
6. Verify that property appears in the list of properties and click OK to close the **Properties** window in Microsoft Word.
7. Position the cursor at the place in the Microsoft Word contract document you want to insert the field for the document property.
8. In Microsoft Word, choose **Insert Field**.
9. In the Field window, highlight DocProperty in the Field Names chooser.
   Select the property you want to insert into the contract document. Properties that use contract attribute fields have the prefix `ca` or `_ca`, such as `caSupplier`. Click **OK** to insert the property.
10. Generate or regenerate the document.

Viewing Editable Document Property Fields

Prerequisites

You must set up a document properties for a contract attributes as described in Setting Up Document Properties Using Ariba Contracts Solution [page 71] or Setting Up Document Properties Using Microsoft Word [page 72].

Context

Microsoft Word displays and handles fields for editable document properties as bookmarks.

Procedure

Double-click an editable document property in a Microsoft Word document to open a window with text form field options. This window also displays the bookmark name for the document property field.

> ⚠ Limitations
>
> Microsoft Word uses bookmarks for editable properties, and as such, any editable document properties you define in your contract document appear in the bookmark list. The SAP Ariba Contracts solution ignores bookmarks used for editable document properties when processing clauses.

Moving Editable Document Property Fields

Prerequisites

You must set up a document properties for a contract attributes as described in Setting Up Document Properties Using Ariba Contracts Solution [page 71] or Setting Up Document Properties Using Microsoft Word [page 72].

Context

If the SAP Ariba Contracts solution inserts an editable document property at the end of a contract document, you must manually move it to the appropriate location.

Procedure

1. Record (write down or copy) the bookmark name.
2. Copy and paste the editable document property to the new location in the document.
3. Double-click the moved field and enter the bookmark name you recorded.

> ⚠ Limitations
>
> Dragging and dropping, or simply cutting and pasting an editable document property without re-entering the original bookmark name results in an unusable document property.

Modifying an Editable Document Property in Microsoft Word

Prerequisites

You must set up a document properties for a contract attributes as described in Setting Up Document Properties Using Ariba Contracts Solution [page 71] or Setting Up Document Properties Using Microsoft Word [page 72].

Procedure

1. Double-click the field in the Microsoft Word document.
2. Microsoft Word opens a text form field options window.
3. Enter the new value in the default text field. For dates, use the format appropriate for your locale (`mm / dd / yyyy` or `dd / mm / yyyy`).

Site Configuration Options for Contract Authoring

Site configuration options are set for you by SAP Ariba.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.ariba.com

THE BEST RUN