

STUDENT TRAINING GUIDE

Configuring Responsive Forms

Copyright © 2024 by Aras Corporation. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.

Distribution of the work or derivative of the work in any standard (paper) book form for a commercial purpose is prohibited unless prior permission is obtained from the copyright holder.

Aras Innovator, Aras, and the Aras Corp "A" logo are registered trademarks of Aras Corporation in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

Microsoft, Office, SQL Server, IIS and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Notice of Liability

The information contained in this document is distributed on an "As Is" basis, without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement. Aras shall have no liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this document or by the software or hardware products described herein.

Revision March 2024



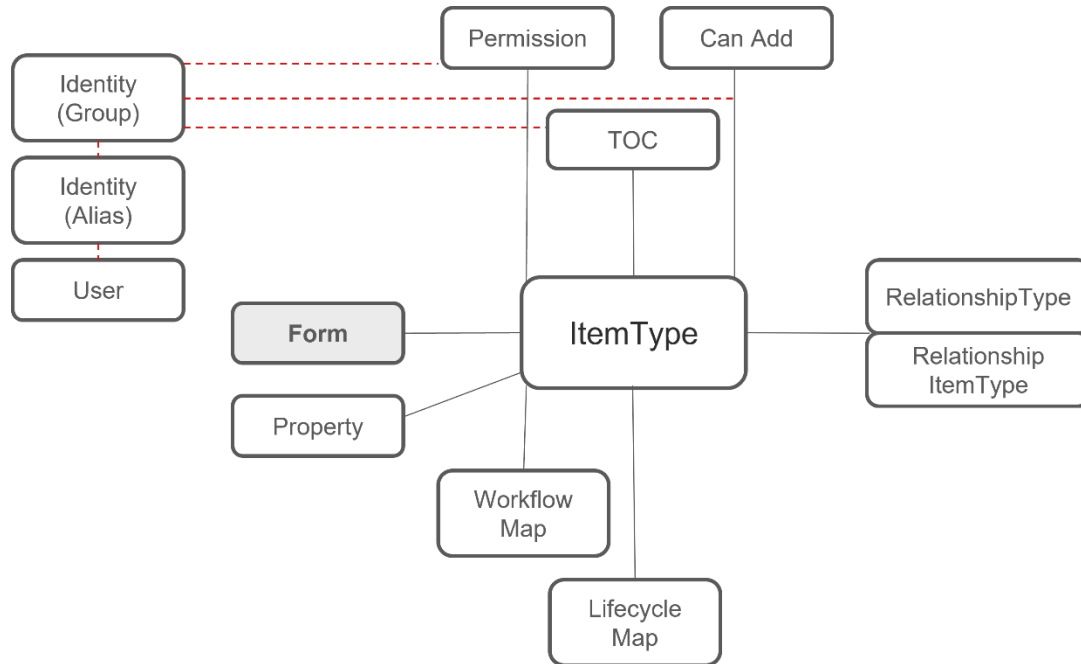
Configuring Responsive Forms

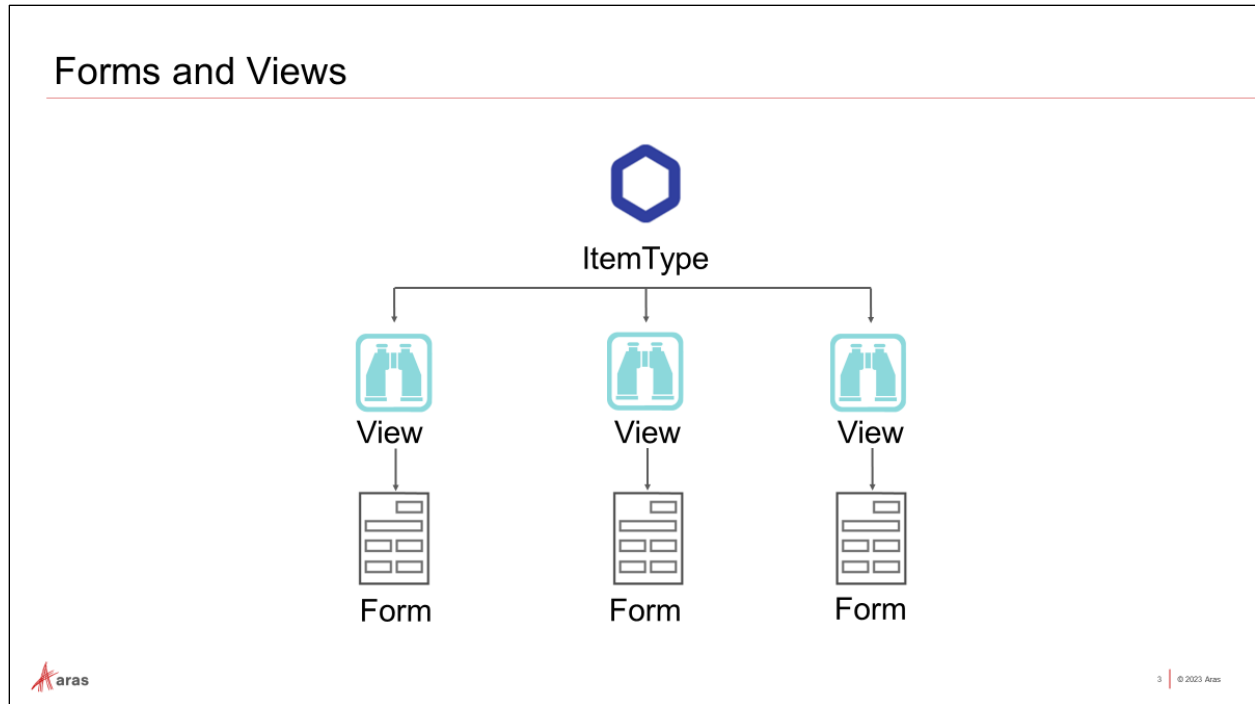
Overview

In this unit, you will create a user interface for a solution that allows a user to interact with web Forms.

Objectives

- Understand Forms and Views
- Review Field Settings
- Build an ItemType View
- Build a Responsive Form





Forms and Views

A Form is a User Interface for an ItemType to allow viewing and editing of an Item instance of that type, provided that the User accessing the Form is authorized to do so.

A View is the relationship between an ItemType and a Form. Forms are designed using the Form Editor, and then associated to an ItemType on the Views Relationship Tab.

The same Item can be displayed to different users in different ways by creating multiple Views and Forms. One Form could be used for editing, while another Form could be used for printing.

Document Forms

Default Form (Classic Form)

The screenshot shows a classic form titled 'Document'. It has a header with 'Document Number', 'Revision', and 'State' fields. Below that are 'Name' and 'Designated User' fields. The 'Type' field is a dropdown menu, followed by 'Authoring Tool' and 'Version' fields. There is an 'Effective Date' field with a calendar icon. A large text area for 'Description' is present, along with a 'Comment' field. At the bottom, there are three checkboxes: 'Files Attached', 'Changes Pending', and 'Template'. A 'From Template' label is at the very bottom.

Classification = Specification (Responsive Form)

The screenshot shows a responsive form titled 'Document'. It has a header with 'Document Number', 'Revision', 'State', and 'Name' fields. The 'Type' field is a dropdown menu set to 'Specification'. Below it are 'Created On' and 'Assigned Creator' fields. There is a 'Designated User' field and an 'Effective Date' field with a calendar icon. A large text area for 'Description' is at the bottom.



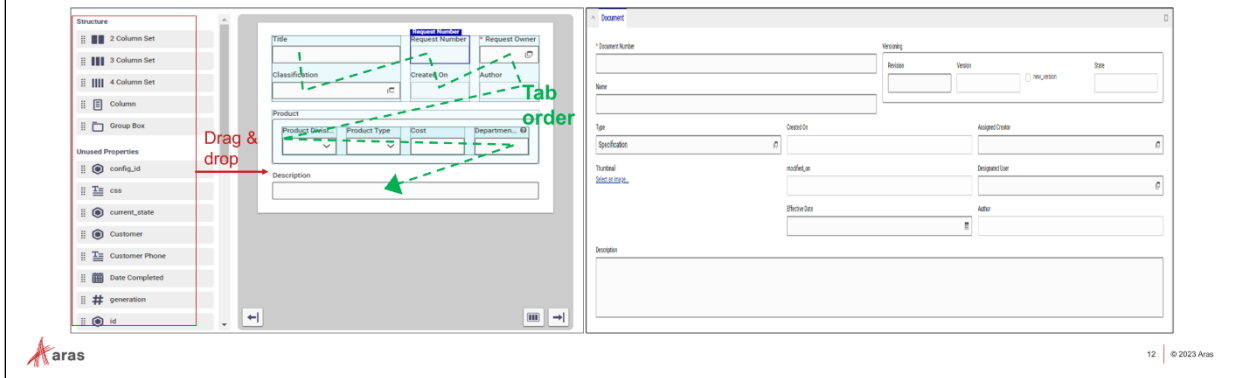
Document Forms

In our example, we will create an additional Form for the Document ItemType. One Form will display by default, and a second Form will display only when a Document Classification value is set to Specification. The new Form will have a combination of the original properties from the Classic Form and additional Document properties.

The default Form will be a classic Form with absolute positioning of the fields in X and Y, whereas the Specification Form will be a Responsive Form, i.e., the fields width and positioning will adapt to the user's screen size.

Responsive Form – Benefits / Differences with Classic Forms

- Adapts to user's screen size
- All properties in the same location
- Visual grouping of fields
- Embedded tab order
- Different data model
- View types



Responsive form – Benefits and Differences vs. Classic Forms

In the next example, we configure the second Form to display fields related to internal requests.

Here are below some differences and/or benefits of responsive Forms versus classic Forms.

Adaptation to user' screen size: This Form is responsive: responsive Design is a general design principle that describes a user interface that adapts its layout and dimensions to various devices or screen sizes. Responsive Forms use a container-based approach to define the layout of a Form. This means that content can reflow to use the browser window's space more effectively – minimizing the need for scroll bars and reducing unnecessary whitespace.

On the contrary, Classic Forms render content using a static layout with X & Y position of each field, which does not change based on the size of the user's browser window. Users will be presented with scroll bars when a Classic Form's static content does not fit in frame, or they will see additional whitespace when the static content occupies less space than the current frame.

All properties in same location: both hidden and unhidden properties appear in the same location for drag-and-drop, improving the user-friendliness.

Easier visual grouping of fields: the Column Sets and Group Boxes enable a clearer grouping of fields. The automatic alignment of those fields also makes the UI easier to ready.

Tab order: related to the navigation across fields using Tab key. For responsive Forms, this setting is embedded in the responsive Form, to reduce the efforts for the administrator.

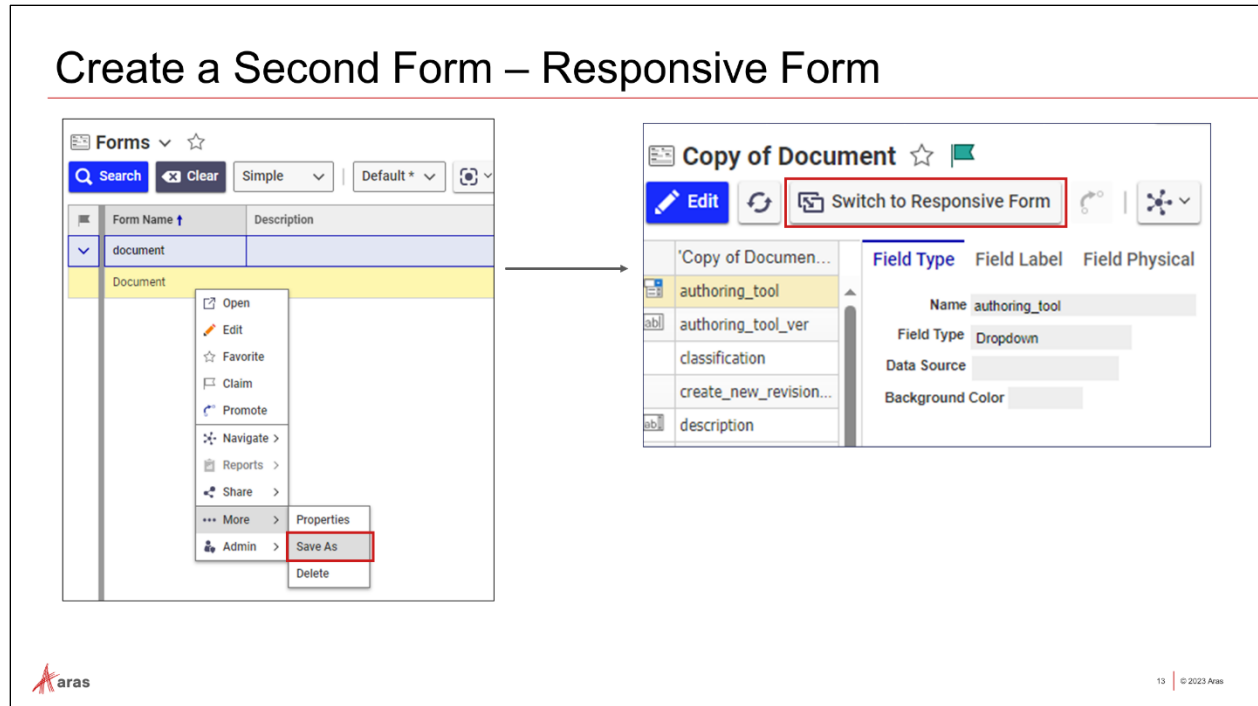
The tab order is setup automatically by the system according to the fields position on the Form:

- From top-left to bottom-right corner
- First on fields within one column
 - o Then, on columns within a column set/group box
 - Then, moving from one column set/group box to another (note: column sets/group boxes can only be stacked on top of each other, not side-by-side)

Data model: Both Classic and Responsive Forms use the Form ItemType to define Forms for the Aras Innovator client. However, each approach uses a different data structure under the Form item to describe the layout, behavior, and content of the rendered Form. This parallel architecture enables administrators to adopt Responsive Forms with minimal disruption to packaging and deployment. It also enables customers to continue using their existing Classic Forms.

Lists : for lists, “List no blank” setting does not exist on Responsive Forms. You can use the Required setting on the list property on related item type.

View types: for the moment, only Default view type is enabled for Responsive Forms. If you need to use other types (print, edit, preview, ...), use classic Forms instead.



Create a Second Form – Responsive Form

The second responsive Form will display only when the Document Classification value is set to Specification. We will copy our existing Form, rename it, and switch to Responsive Form to manipulate Responsive Forms (Forms adapting to user's screen size horizontally).

A single Form item might have both Classic and Responsive definitions present at the same time. However, Aras Innovator will only display one of them, according to the active definition. You can switch from one definition to the other at any time. Modifying the layout, behavior, or content of a classic definition does not affect the Responsive definition, and vice versa. For example, after form creation, if you do modifications in the classic form then switch to Responsive Form, the responsive definition will be empty (by design).

Try it ... Copy a Form

We will use the Forms search grid to copy a form.

1. Log into Aras Innovator with username: *admin* and password: *innovator*.
2. Navigate to Administration > Forms, open the Forms search grid, and search for the **Document** Form.
3. Right-click on the **Document** Form and choose More > Save As. Notice that the configuration of fields from source Form were copied in the new Form's classic definition.
4. Click Switch to Responsive Form.
5. Click Edit on the new Form. Notice that the properties do not appear yet on the bottom-left corner, because we have not linked the Form to an Item Type yet.
6. Change the Name to *Document Specification*.
7. Click Done.
8. Close the Form tab.

Creating a View

- Views relate an ItemType to a Form



- A View (and related Form) is automatically created the first time an ItemType is saved

Form Name ↑ 2	Type ↑ 1	Description	Identity [...]	Classification	Display Priority	Secure Social
Document	Default		World			<input checked="" type="checkbox"/>



14 | © 2023 Aras

Creating a View

Views are the relationship between an ItemType and a Form. For an end user to see the Form, you must associate it with an ItemType. Aras Innovator automatically creates a view to a Form with the same name as the ItemType the first time you save an ItemType to the database. Additional Forms can be associated with an ItemType using the Views relationship tab.

Configuring Views

Form Name ↑ 2	Type ↑ 1	Description	Identity [...]	Classification	Display Priority	Secure Social
Document	Default		World			<input checked="" type="checkbox"/>
Document Specification (R)	Edit		All Employees	Specification		<input type="checkbox"/>

Aras 15 | © 2023 Aras

Configuring Views

Multiple views can be configured for an ItemType to provide custom displays based on function, identity, and classification. There must always be one Form that is not classified, with its Type set to either Default or Add.

Here are the views settings:

- ① **Type**
 - Default** – Form used when adding, editing, viewing, or printing an Item.
 - Add** – Form used the first time an Item is added.
 - Edit** – Form used when the Item is locked for editing.
 - View** – Form used for viewing the Item.
 - Print** – Form used to print the Item.
 - Complete** – Form used for configurable InBasket dialog.
 - Preview** – Display when Preview > Form is enabled in the Search tab.
- ② **Identity** Form available for members of defined Identity.
- ③ **Classification** If Class Structure is implemented, the Form to display for a specific class.
- ④ **Display Priority** If more than one Form meets the selection criteria, the Form display order priority (1 = highest priority, 2, 3, etc.).

Reminder: currently, responsive Forms only support the Default view type. For view types like Add, Edit, View, Print, Complete, or Preview, users are advised to use classic Forms.

Try it ... Create and Classify a View

In this example, we add a Specification Classification View and allow only employees to view it.

1. Open the **Document** ItemType for Editing.
2. On the Views tab, click the Add Forms button.
3. Search for and select the **Document Specification** Form.
4. Set the Type to **Default**, the Identity to **All Employees**, and Classification to **Specification**.
5. Click Done.

Edit the Document Specification Form - Responsive Form

The screenshot displays a web form titled "Document" with the following fields and annotations:

- Document Number:** SPEC-0001
- Name:** Test Responsive Form Document
- Type:** Specification
- Thumbnail:** [Select an image...](#)
- Created On:** 2/20/2024 4:54:32 PM
- modified_on:** 2/20/2024 4:54:48 PM
- Effective Date:** (empty)
- Versioning:**
 - Revision: A
 - Version: 1
 - new_version
 - State: Preliminary
- Assigned Creator:** Component Engineering
- Designated User:** Component Engineering
- Author:** Innovator Admin
- Description:** (empty text area)

Annotations in red text:

- "Column Set with stacked fields" points to the "Document Number" and "Name" fields.
- "Field, no Column Set or Group Box" points to the "Description" field.
- "Group Box – 1 field per column" points to the "Versioning" section.

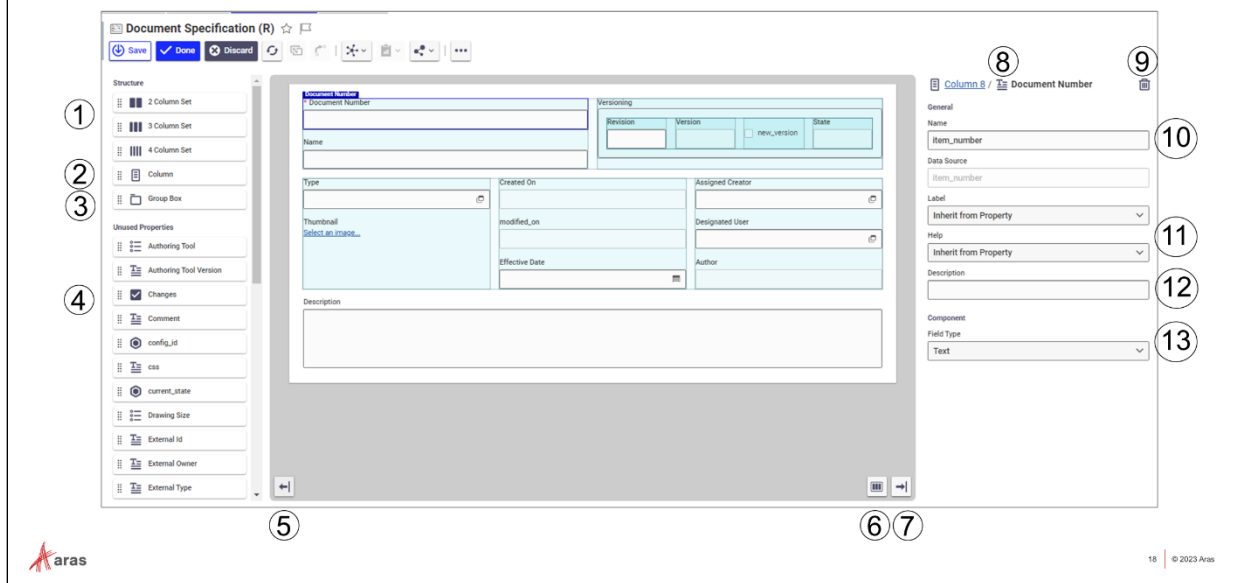
Aras logo is visible in the bottom left corner, and "17 | © 2023 Aras" is in the bottom right corner.

Edit the Document Specification Form – Responsive Form

Our responsive Form will contain different Aras responsive Form features:

- Column Set: container holding multiple columns, each column can contain one of several fields.
- Group Box: a column set with label and border displayed by default.
- Individual field dragged-and-dropped directly on the Form.
- column with double width compared to other columns.

Fields settings - Responsive Form



Fields settings – Responsive Form

Responsive Forms use an enhanced admin user interface allowing admins to define a container-based layout by dragging and dropping structural elements and fields into the Form. Administrators can also configure the layout behavior of the Form and preview how it will look and behave for end users.

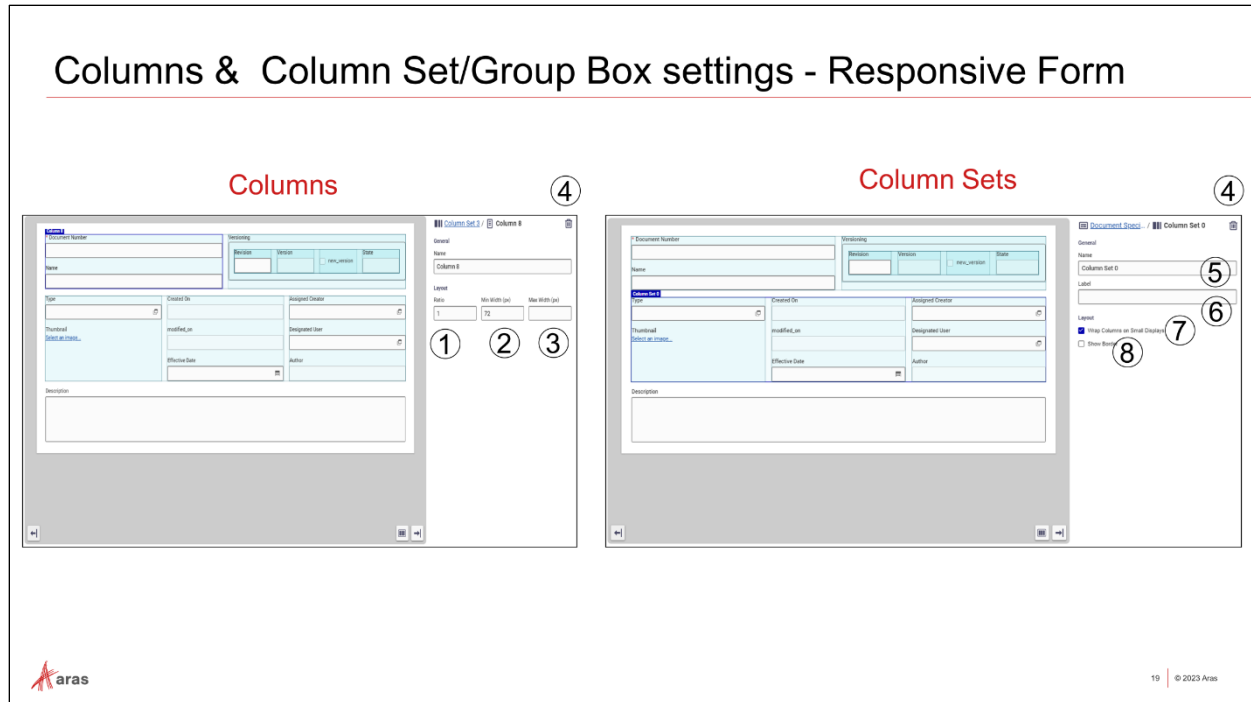
Note: drag and drop on the white part of the middle canvas, not the grey part.

- ① **Column Set** Container holding multiple columns. Each column can contain one of several fields. Column sets help organize and arrange Form fields or content side by side on wider screens. Users can drag-and-drop column sets with 2, 3 or 4 columns. Label and border are hidden by default.
- ② **Column** When dragging-and-dropping a column in a column set/group box, an additional column is added. Column widths get resized accordingly to keep the same total width. Aras Administrators can add as many columns as needed; the only limit is the minimum column width.
- ③ **Group Box** A column set with label and border displayed by default.
- ④ **Unused Properties** All properties from the related ItemType, both hidden and unhidden. Users drag and drop properties within columns or directly on the canvas. Reminder: if the responsive Form is not linked to any specific ItemType, then the Unused Properties will not appear.
- ⑤ **Close Panel** Closes the left-hand side Element panel.
- ⑥ **Hide Guides** Displays the Form's actual appearance as it will be shown to the user
- ⑦ **Close Panel** Closes the right-hand side Detailed panel.

- ⑧ **Hierarchy** Allows to navigate in the Form > Column sets > Columns > Fields hierarchy.
- ⑨ **Delete** Deletes the selected element.
- ⑩ **Name** Name of the field displayed on the designer. Inherited from ItemType property.
- ⑪ **Label & Help** Label stands for the Field label displayed to the end users. Help is related to tooltip. For both, the possible values:
 - Inherit from property: shows the label/help from the ItemType property
 - Hide: label/tooltip is hidden for users
- ⑫ **Description** A second configurable "label" beneath the input field, which administrators may use as descriptive helper text to assist end-users with understanding what data should be input to the field or how it relates to business processes.
- ⑬ **Field Type** Shows the field type based on the property data type (e.g. Text, Combo-box, Number, Item, Checkbox, Image).
Only Text property type has multiple field types as follows:
 - Text: A concise input field suitable for short entries.
 - Text Area: A larger input space ideal for longer pieces of text or paragraphs.

Try it ... Add Properties on the canvas

1. From the Views tab, open the **Document Specification** Form for editing.
2. Drag and drop a 2 Column Set on the white area of the canvas.
3. Drag and drop the following properties:
 - a. **Document Number** and **Name** in the first column.
 - b. Drag and drop a Group box in the second column. Notice that one column appears by default in the group box.
 - c. Drag and drop a 4 Column Set inside the Group box.
 - d. Drag and drop one property in each column, in this order: **Revision**, **generation**, **new_version**, **State**.
4. Drag and drop a 3 Column Set into the white area of the canvas, below the previous Column Set.
5. Drag and drop one property in each column, in this order: **Type** and **Thumbnail** in the first Column; **created_on**, **modified_on**, and **Effective Date** in the second; and **owned_by_id**, **managed_by_id**, and **created_by_id** in the third.
6. Drag and drop the **Description** property at the bottom of the canvas' white area. Notice that this field can have field type Text or Text Area.
7. Click Save.



Columns and Column Set/Group Box settings - Responsive Form

The Columns, Column Sets and Group Boxes can be set up to configure the labels, behavior of those elements when the screen gets reduced, etc.

You can use the top right corner to navigate within the hierarchy of Responsive Form > Column Set/Group Box > Column > Field.

Column settings

- ① **Ratio** Space each column occupies relative to others in terms of width. For instance, a column ratio of "2:1" between two columns means that one column is twice as wide as the others.
- ② **Min Width** Minimum width of the column in pixels, used when screen is reduced horizontally.
- ③ **Max Width** Maximum width of the column in pixels.
- ④ **Delete** Deletes the element (here, a column).

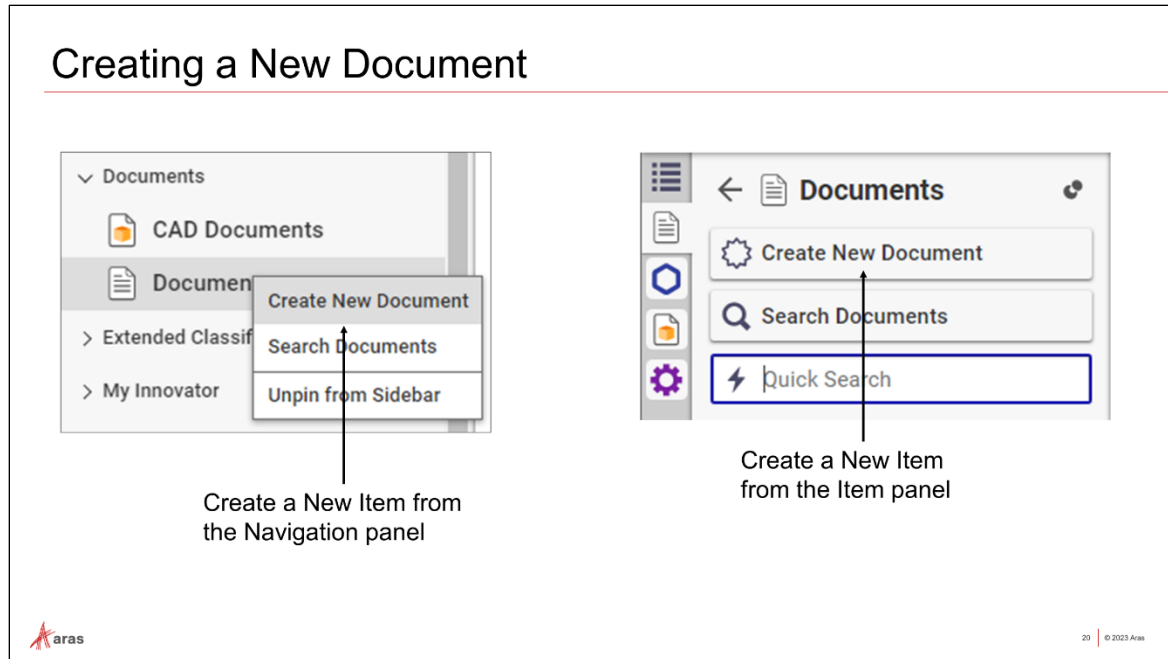
Column Set & Group Box settings

- ④ **Delete** Deletes the element (here, a column set / group box).
- ⑤ **Name** Provides name of the element.
- ⑥ **Label** Editable name of the element - is displayed to the user.

- ⑦ **Wrap Columns on Small Display** Adjusts columns to stack vertically (or "wrap") instead of aligning horizontally when viewed on devices with limited horizontal space. This ensures content remains legible and user-friendly regardless of the screen size.
- ⑧ **Show Border** Displays the border of the column set.

Try it ... Change the Columns and Group Box settings

1. Select the **Name** field of the first column.
2. On the Detailed panel (right-hand side), click on the Column link. This opens the column's detailed panel.
3. Notice the min and max width settings ; keep their default values.
4. On the Detailed panel (right-hand side), click on the Column Set link. This opens the Column Set's detailed panel. Notice the column set settings.
5. Click on the Group Box header, then change the Label field to **Versioning**.
6. At the bottom, click Close Panel on both left and right sides.
7. Click Hide Guides at the bottom.
8. Click Done.



Create a New Document

Now that the minimum requirements have been defined and a class structure established, create a new Document to see the result.

Create a new Document Item by either right-clicking on the Documents ItemType button in the TOC and selecting Create New Document, or by left-clicking on the Documents ItemType button in the TOC and then clicking the Create New Document button.

The **Document Form** should allow for the following behaviors:

- The Revision, Version, and State fields will populate after saving the Document.

Try it ... Create a Document

1. Navigate to Documents > Documents and create a new Document.
2. If necessary, maximize the Form to view all the fields.
3. Enter a value in the Document Number field.
4. Leave the **Type** field as is and enter values into the remaining editable fields to check the new Form.
5. Click the Done button to save your new Document and release the Claim.
6. Create another Document.
7. Enter *Specification* into the **Type** field and enter applicable values into the remaining editable fields to check the new Form.
8. Notice the content resizes with the changing screen size, tooltip, and tab order behaviors.
9. Click the Done button.

Summary

In this unit, you learned how to create an Item View using the Form Editor.

You should now be able to:

- Understand Forms and Views
- Review Field Settings
- Build an ItemType View
- Build a Responsive Form