



Antelope 6 Workspace Form Designer

Created By

Antelope International Limited

Document No. A624005

Trademarks

The Antelope logo is a registered trademark of Antelope International Limited.

Antelope is a registered trademark of Antelope International Limited.

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

Apple, iOS, VisionOS, WatchOS, MacOS, Safari are registered trademarks of Apple Inc.

Android and Chrome is registered trademarks of Google.

PostScript is a registered trademark and PostScript 3 is a trademark of Adobe Systems Incorporated.

Each company whose software title is mentioned in this manual has a Software License Agreement specific to its proprietary programs.

All other brand and product names mentioned in this User's Guide are registered trademarks of their respective companies.

© 2024 Antelope International Limited. All rights reserved.

Table of Contents

A.	Concept Behind	5
I.	Between eForm and Document Profile	5
B.	eForm Designer	7
I.	Create file from Template	7
II.	Create the first Page	8
C.	Section	10
I.	Create new Section	10
II.	Configure the Section	11
D.	More eForm Objects	14
I.	Label	14
II.	Textbox	15
III.	Different Textbox Type	17
IV.	Databinding	18
V.	Label Textbox	20
VI.	Design, Preview and Mobile View	24
VII.	Pull-Down List & Label Pull-Down List	25
VIII.	Label Checkbox	32
I.	Grid Data	37
II.	Signature	43
III.	Image	45
IV.	Index Lookup	47
V.	Folder Selector	51
VI.	File Uploader	52
VII.	Rich Text Editor	54
E.	Script Library	55
I.	Background	55
II.	Create New Library	55
III.	Image.QRCode	57
IV.	Maths.MultiplyValue	59
V.	Maths.SumValue	61
VI.	UI.ComboboxControl	63
VII.	UI.CheckboxControl	65

F.	Script Editor	68
I.	Load Script Editor.....	68
G.	Object Functions and Events.....	69
I.	Label	69
II.	Textbox.....	70
III.	Combobox (Pull-Down List)	72
IV.	AutoComplete (Pull-Down List)	74
V.	Signature.....	76
VI.	Image	77
VII.	Data Grid	78
H.	Global Variable.....	81
I.	All Variables	81
I.	Language Translation	82
J.	eForm Advanced Setting	83
I.	Comment Color	83
II.	Generate Input Border for PDF	83
III.	Generate Signature Border for PDF.....	83
IV.	Page Rendering Mode	84
V.	Canvas Background Color	84
VI.	Variable	85
K.	Validation Test	86
I.	Validation Icon	86
L.	Advanced Setting of Data Grid	86
I.	Sum Total	86
II.	Total Data Binding	86
III.	Hidden.....	86
IV.	Rowspan	86
V.	Combo Related Column Index	86
VI.	Formular.....	86
VII.	Numeric Mask	86

A. Concept Behind

I. Between eForm and Document Profile

1. Before designing an eForm, it's important to understand the relationship between an eForm and a Document Profile.

An eForm is an XML file containing the form layout, objects like textboxes, pull-down lists, images, data grids, and the values filled in by users.

A Document Profile, on the other hand, is a database table that stores all the information, which can be retrieved through advanced searches or exported to other systems for integration and queries.

An eForm may include hundreds of fields, such as in a loan application form, a long survey, or a customer profile. However, not all information needs to be stored in the database for search purposes.

So, before designing an eForm, ask yourself: "What information in the eForm do I need for searching or integration? What information is just for display and reading purposes?"

2. Here's an example of a customer survey form. The eForm includes fields such as Your Name, Gender, Email, Mobile, Your Comments, and Signature. However, after interviewing end-users, it turns out they will only search these records by Your Name, Gender, Email, Mobile, Feeling and Comments. (Signature is not required to store in database).



Customer Survey Form

Your Name:

Gender:

Email:

Mobile:

Product Interest

☒ Product A ☒ Product B ☒ Product C ☒ Product D

Tell us your Feeling

Product Name	Desire of Purchase (0-5)
Product A	5
Product B	3
Product C	0
Product D	1

Your Comments:

Sign Here: 



Document Profile
(Database Table)

Field	Field Type						
Your Name	String (50)						
Gender	String (1)						
Email	String (255)						
Mobile	String (20)						
Tell Us Your Feeling	Details Line Item						
	<table> <tr> <th>Field</th><th>Field Type</th></tr> <tr> <td>Product Name</td><td>String (10)</td></tr> <tr> <td>Desire of Purchase</td><td>Integer</td></tr> </table>	Field	Field Type	Product Name	String (10)	Desire of Purchase	Integer
Field	Field Type						
Product Name	String (10)						
Desire of Purchase	Integer						
Product Interest	String (255)						
Comments	String (255)						

3. Here is a summary of your “research” after interviewing end-users on their using habits, and you start need to plan for your design.

Field in eForm	Field in Document Profile	Data Type in Document Profile	Sample
Your Name	Your Name	String (50)	Jennifer Tam
Gender	Gender	String (1)	F
Email	Email	String (255)	jennifer@demo.com
Mobile	Mobile	String (20)	+651234567
Product Interest	Product Interest	String (10)	A,B,C,D
Your Comments	Comments	String (255)	Here is my comment...
Tell us your Feeling	Tell us your Feeling	Details Line Item i. Product Name (String) ii. Desire (Integer)	Product A: 5 Product B: 3 Product C: 0 Product D: 1
Signature	-	-	-

4. In conclusion, the steps of implementing an eForm follows the below steps:
- Create the Document Profile. For how to setup a document profile, please refer to “A624002 – Antelope 6 Administration Manual, Section G”.

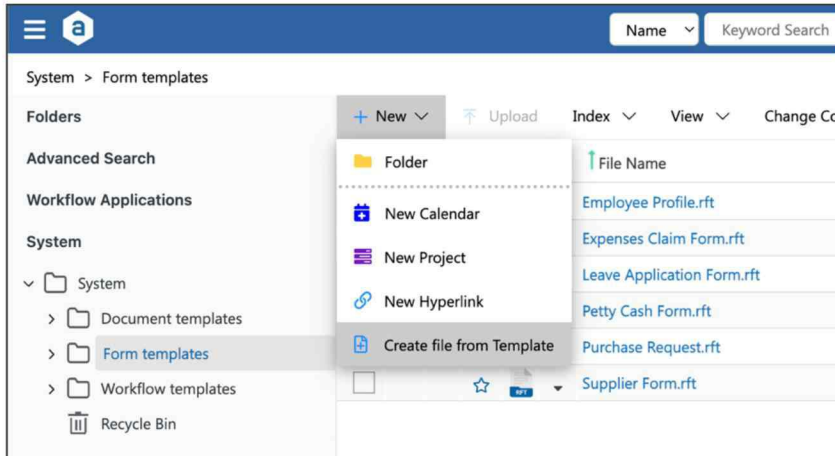
Document Profile Fields of Customer Survey Form							
					Type	Name	Length
					String	Your Name	50
					String	Gender	1
					String	Email	255
					String	Mobile	50
					String	Product Interest	10
					String	Comments	255
					Detail	Tell Us Your Feeling	8

- Create the eForm
- Map eForm to Document Profile
- Create a folder to host the eForm records.

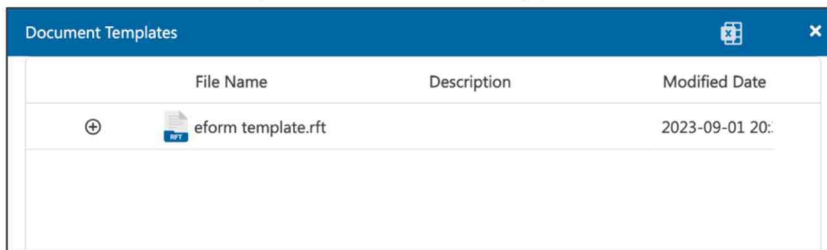
B. eForm Designer

I. Create file from Template

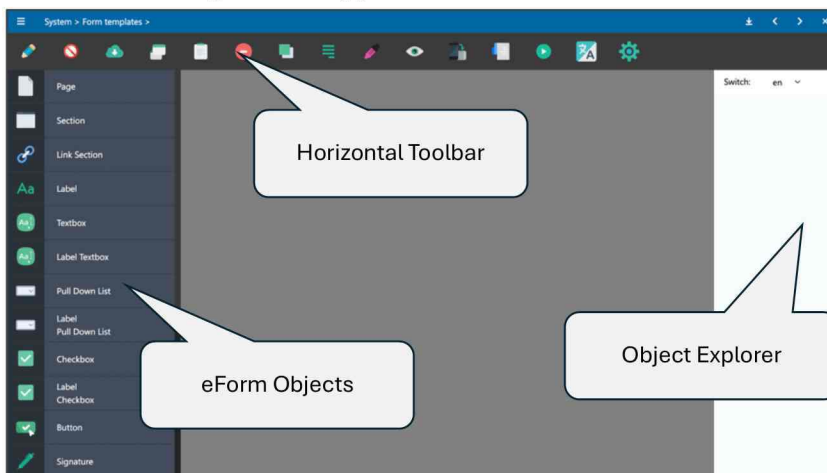
1. To create an empty eForm, select “System > Form Templates”, from the document list toolbar, select “New > Create file from Template”.



2. The Document Templates window will appear. Click the ⊕ icon.

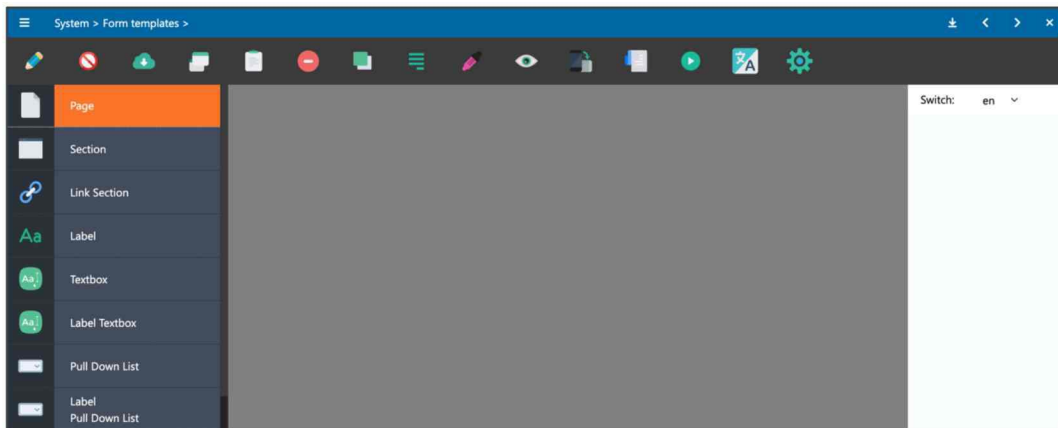


3. The eForm Designer will appear.

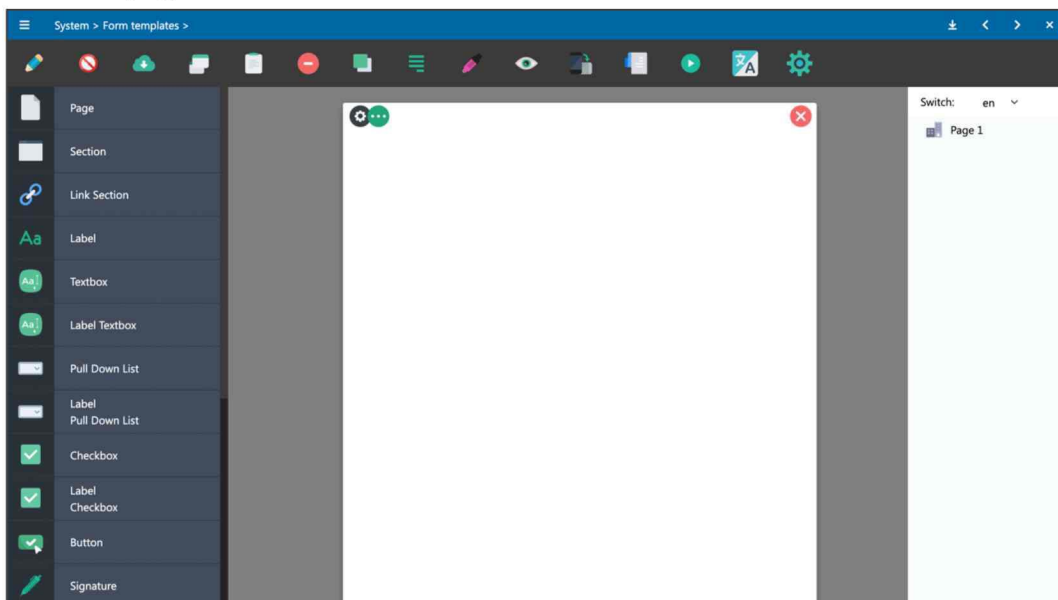


II. Create the first Page

1. Drag and drop the “Page” object to the central blank area.



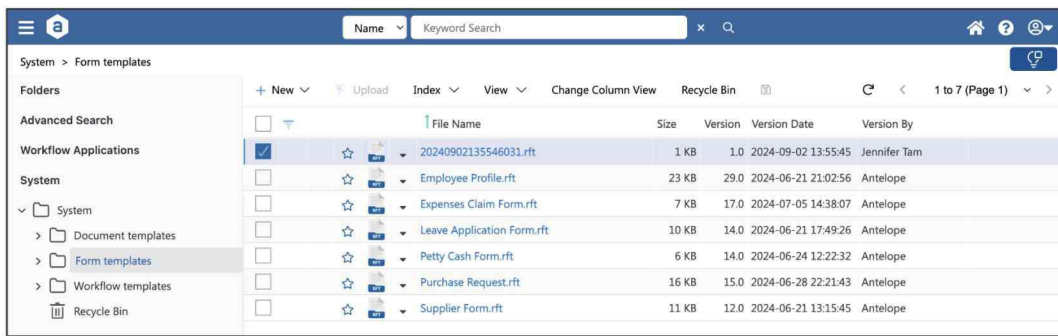
2. Your first page is created.



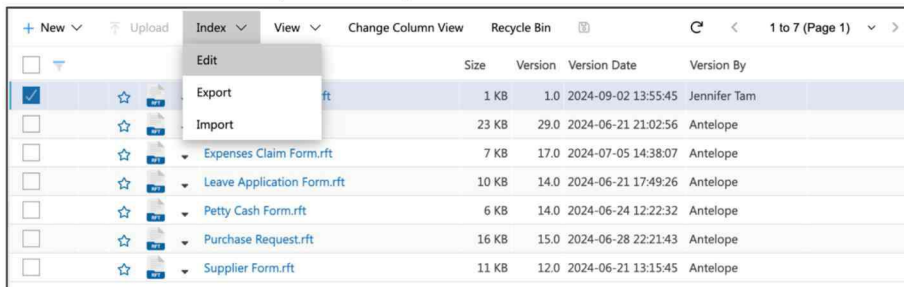
3. Before making any further configuration, it is recommending to save your new eForm first by clicking the pencil icon.



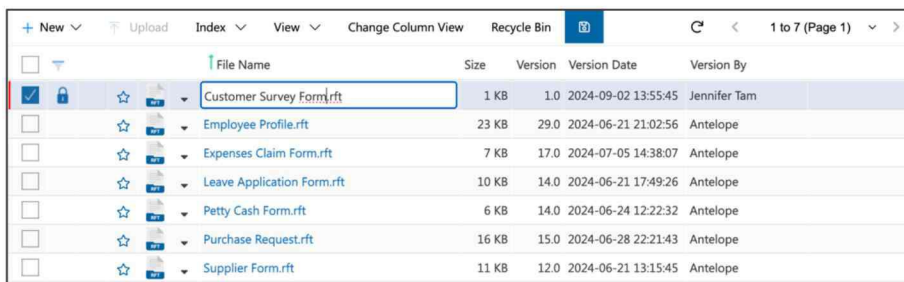
4. A new eForm is created.



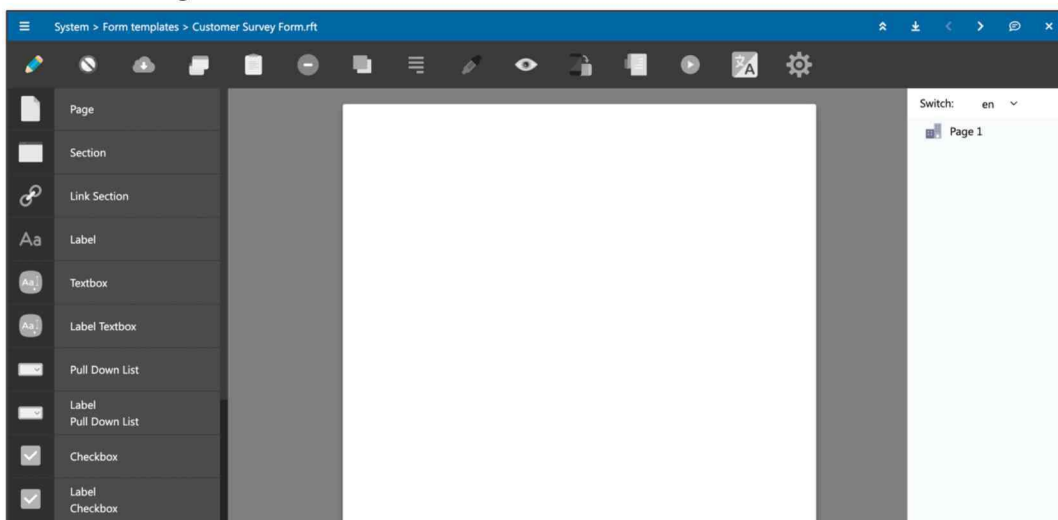
5. Rename the eForm by selecting “Index > Edit” from the document list toolbar.



6. Click Save button to rename.



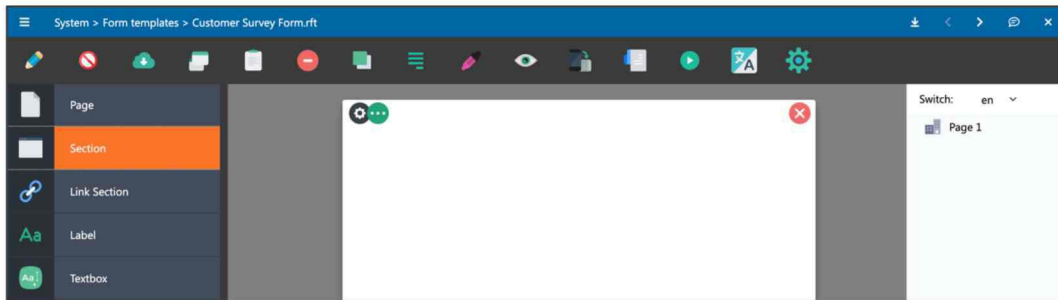
7. After renaming, open the eForm template again, and click the pencil icon to check-out for editing.



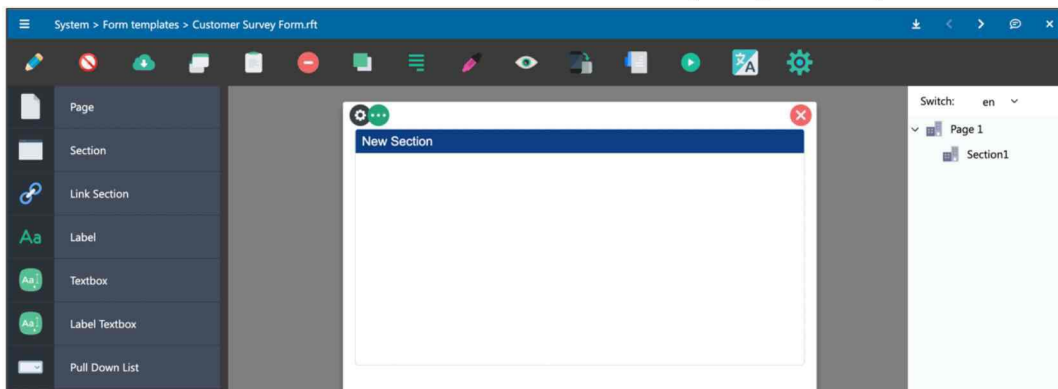
C. Section

I. Create new Section

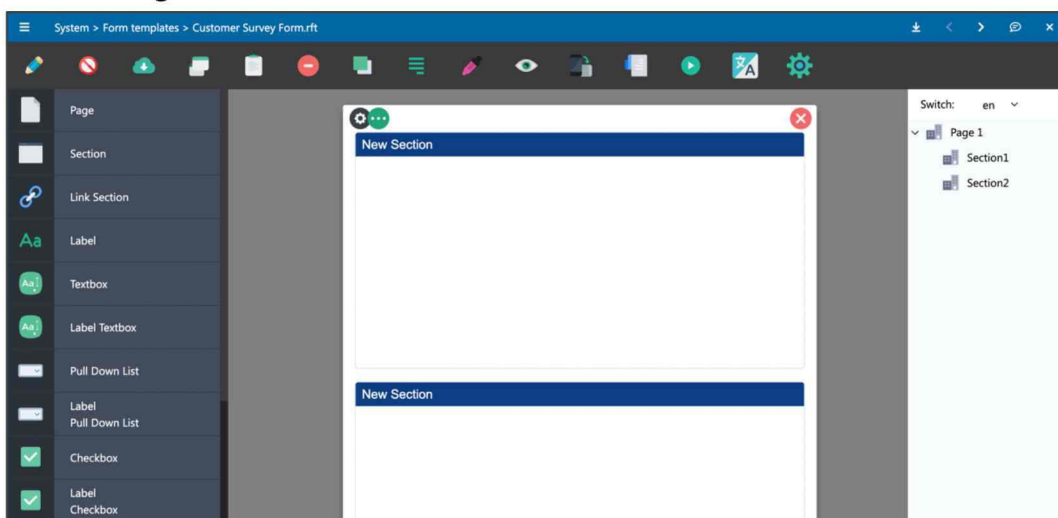
1. You can build several Sections in an eForm. The function of Section is to group different objects into areas and allows designer to control the access permission of sections to different user accounts / user groups.
2. Drag and drop the Section object onto your Page.



3. A new section is created. You can move the section by drag the deep blue section bar.



4. You can drag several sections on an eForm.

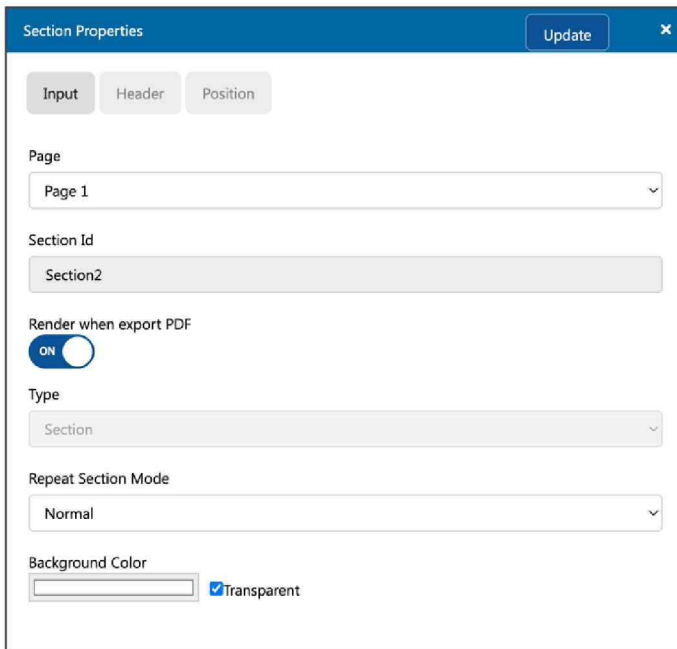


II. Configure the Section

1. To configure a section, highlight the blue section bar, and then click the  icon.



2. The Section configuration window will appear.

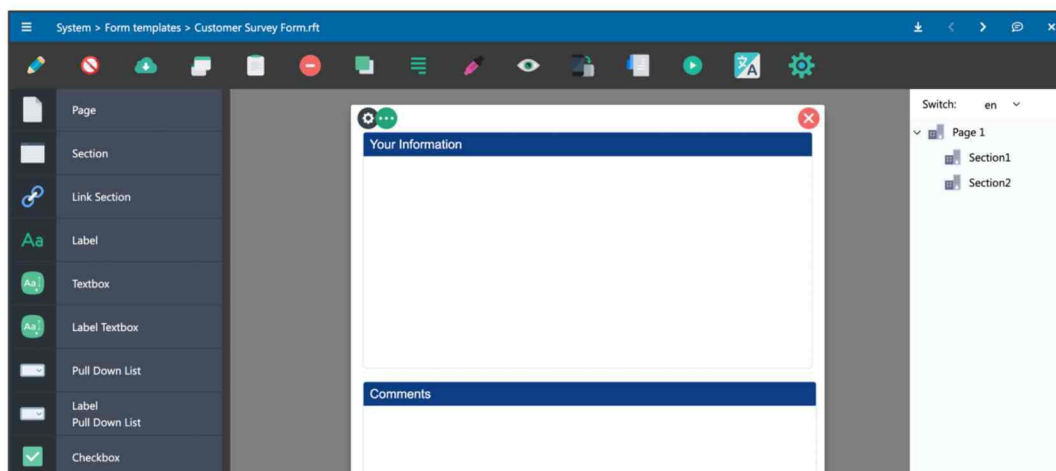


Field in eForm	Field in Document Profile
Page	Page number.
Section Id	The section object ID, which can be retrieved from the right-hand side column.
Render when export PDF	When the eForm exports as PDF file, whether the section border will be included in the PDF file.
Type	By default, is Section.
Repeat Section Mode	This is used for "Link Section", when an eForm contains more than one page, they can share the same Section.
Background Color	Section background color, by default, is transparent.

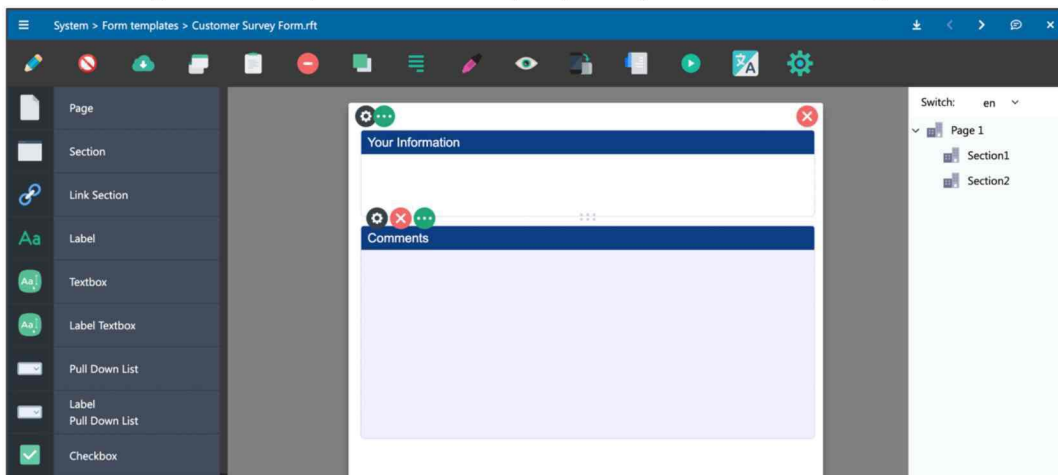
3. More setting is available in the next tab.

Field in eForm	Field in Document Profile
Header Text	Section title
Font Family	Section header font type
Font Size	Section header font size
Color	Section header font color.
Background Color	Section header background color.
Bold, Italic, Underline	Section header text.

4. You can rename both Sections header title.



5. eForm Designer allows you to resize any object by its width and height.



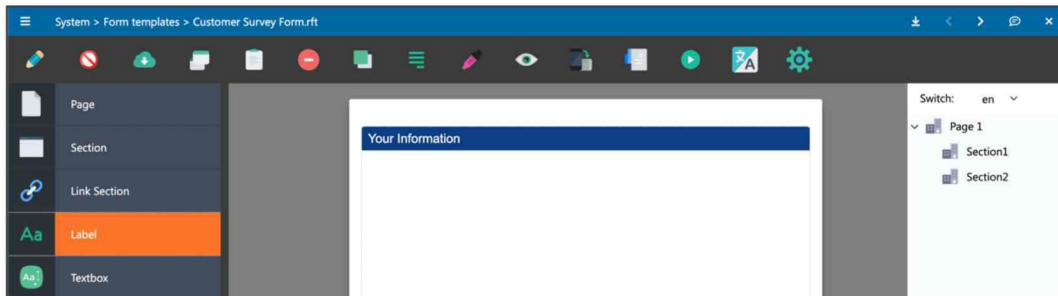
6. You can also adjust the Left, Top, Width and Height by their values in the Section configuration more precisely.

The screenshot shows the 'Section Properties' dialog box. It has a title bar with 'Section Properties' and an 'Update' button. Below the title bar are three tabs: 'Input', 'Header', and 'Position'. The 'Position' tab is selected. It contains four input fields for configuring the section's position and size: 'Left' (value 15), 'Top' (value 34), 'Width' (value 575), and 'Height' (value 110).

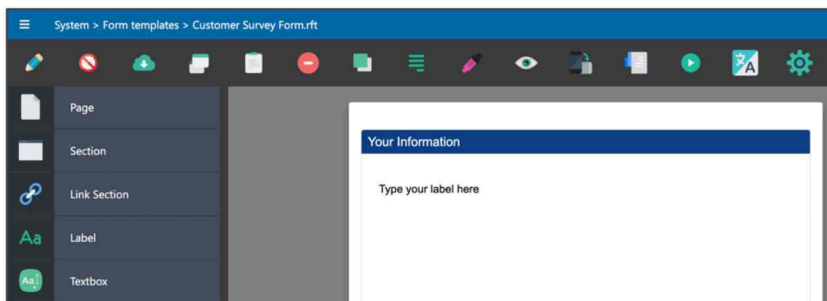
D. More eForm Objects

I. Label

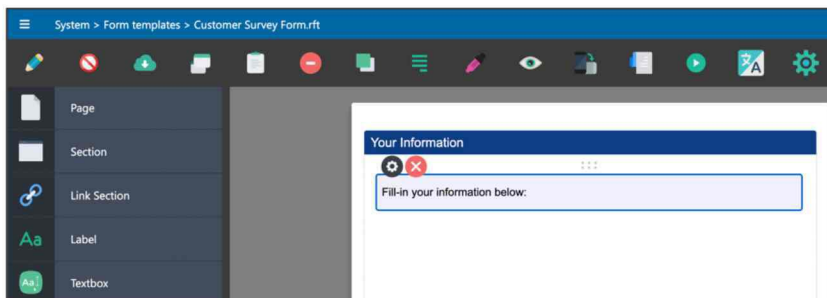
1. Drag and drop the label object onto the first Section.



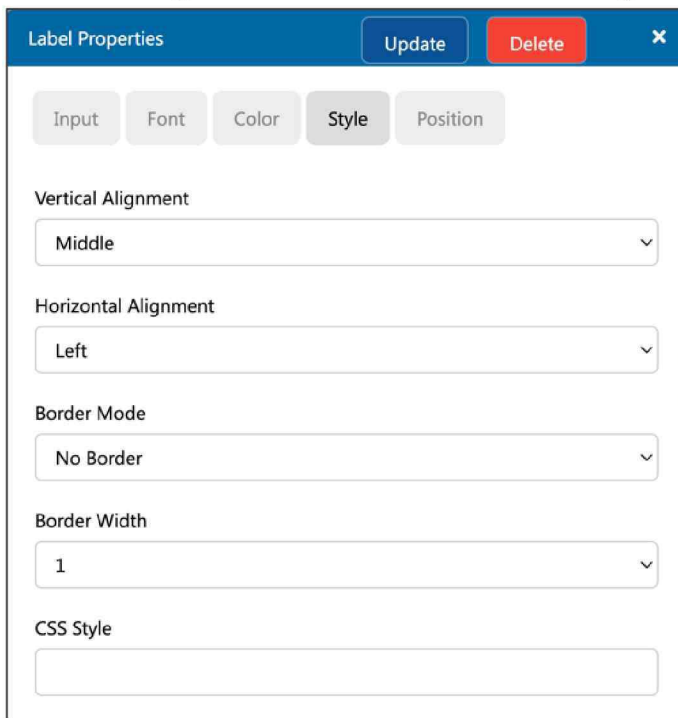
2. A new label is added.



3. Resize the label object and input your message.



4. You can configure the label with different setting from alignment, font properties and css.

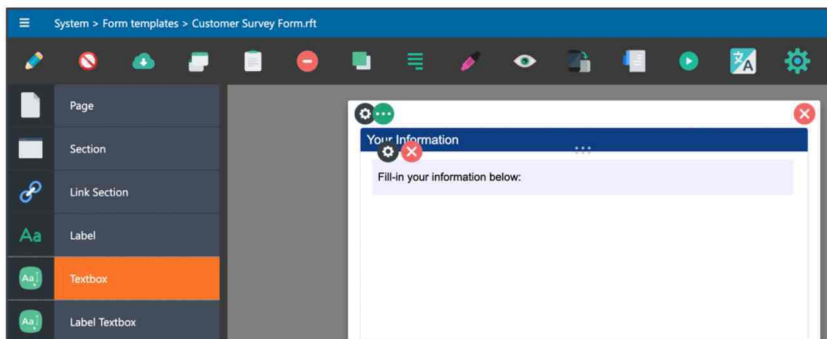


The 'Label Properties' dialog box is shown with a blue header bar containing 'Update' and 'Delete' buttons. Below the header are five tabs: 'Input', 'Font', 'Color', 'Style', and 'Position'. The 'Style' tab is selected. The settings are as follows:

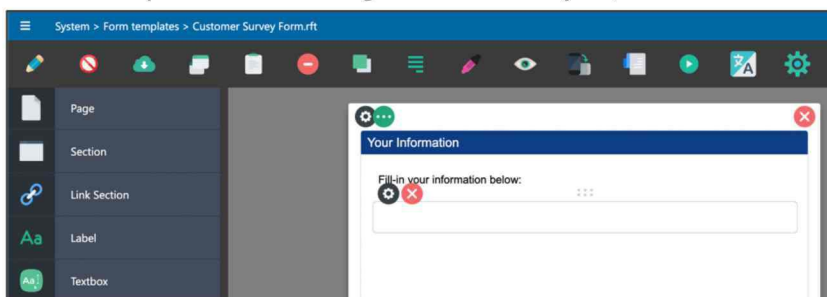
- Vertical Alignment: Middle (dropdown)
- Horizontal Alignment: Left (dropdown)
- Border Mode: No Border (dropdown)
- Border Width: 1 (dropdown)
- CSS Style: (empty text field)

II. Textbox

1. Drag and drop the textbox object onto the first Section.



2. There are quite a few settings in textbox object, which will be explained in below:



3. Input

Field Name	Description
Section	Section ID
Object ID	Textbox Object ID
Type	Textbox, Multiple Line Textbox, Password, Date, Datetime, User Selector, User and Group Selector, Currency
Default Value	Username User Description User Group's Name User OU's Name (On-Premises Version) User Department (On-Premises Version) User Company (On-Premises Version) Today's Date
Mandatory	The field is set as mandatory
Max Length	Maximum length of textbox in which user can input

The screenshot shows the 'Textbox Properties' dialog box with the 'Input' tab selected. The dialog has a title bar with 'Update' and 'Delete' buttons. The tabs are Input, Data bind, Font, Appearance, Event, and Position. The 'Input' tab contains the following fields:

- Section:** A dropdown menu showing 'Section1'.
- Object ID:** A text input field containing 'Textbox1'.
- Type:** A dropdown menu showing 'Textbox'.
- Default Value:** A dropdown menu showing 'No Default'.
- Mandatory if section is editable:** A toggle switch currently set to 'Off'.
- Max Length:** A text input field containing '200'.

4. Data Bind

Field Name	Description
Data bind	Bind to index field of document profile. For more information, please refer to next section "Databinding".
Data Export Format	Format of date export
Use date picker for Date Type	When textbox type is Date or Datetime, allow to use date picker.
Single User for User Type	When textbox type is User or User Group, only allows to select a single user each time
Validation	Check the data type validation: Numeric, Date, Time, Email, HKID

The screenshot shows the 'Textbox Properties' dialog box with the 'Data bind' tab selected. The dialog has a title bar with 'Update' and 'Delete' buttons. The tabs are Input, Data bind, Font, Appearance, Event, and Position. The 'Data bind' tab contains the following fields:

- Data bind:** A dropdown menu.
- Date Export Format:** A dropdown menu showing 'Default'.
- Use datepicker for Date Type:** A toggle switch currently set to 'Off'.
- Single User for User Type:** A toggle switch currently set to 'Off'.
- Validation:** A dropdown menu showing 'None'.

5. Font

Field Name	Description
Font Family	Font Type
Font Size	Font Size

The screenshot shows the 'Textbox Properties' dialog box with the 'Font' tab selected. The dialog has a title bar with 'Update' and 'Delete' buttons. The tabs are Input, Data bind, Font, Appearance, Event, and Position. The 'Font' tab contains the following fields:

- Font Family:** A dropdown menu showing 'Arial'.
- Font size:** A dropdown menu showing '14'.

6. Appearance

Field Name	Description
Text Align	Left, Right or Center text alignment within the textbox
Generate Border in PDF	When the eForm export as PDF, whether the textbox contains border
Place Holder	Textbox place holder and tool tips
Display Border Mode	Whether the textbox display on eForm in style of box or underline.

Textbox Properties

Update Delete

Input Data bind Font Appearance Event Position

Text Align
Left

Generate Border in PDF

Place Holder

Display Border Mode
Box

7. Event

This opens the eForm script editor. Please refer to eForm scripting sections.

Textbox Properties

Update Delete

Input Data bind Font Appearance Event Position

[Open Script Editor](#)

III. Different Textbox Type

The following are different textbox types as they appear in Preview mode:

Textbox Type	Default Value	Preview
General Textbox		This is a test
General Textbox	Username	jennifer
General Textbox	User Description	Jennifer Tam
General Textbox	User Group's Name	Management
General Textbox	Today	2024-09-02
Multiple Line Textbox		This is line 1 This is line 2 This is line 3 This is line 4
Password		

Textbox Type	Other Setting	Preview
General Textbox	With Placeholder	e.g. Peter
General Textbox	Box Border Style	This is a test

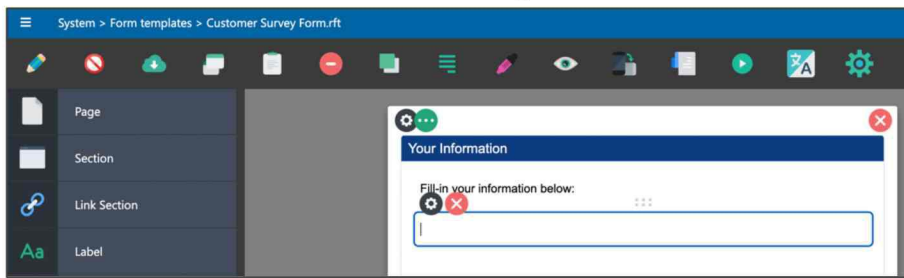
General Textbox	Underline Border Style	e.g. Peter
Date	Without Date Picker	年 / 月 / 日
Date	With Date Picker	2024-09-04
Datetime	Without Date Picker	2024/09/04 上午 12:00:00
Datetime	With Date Picker	2024-09-05 22:11
User Selector		Jennifer Tam
User and Group Selector		Accounting Department
Currency		\$216.00

IV. Databinding

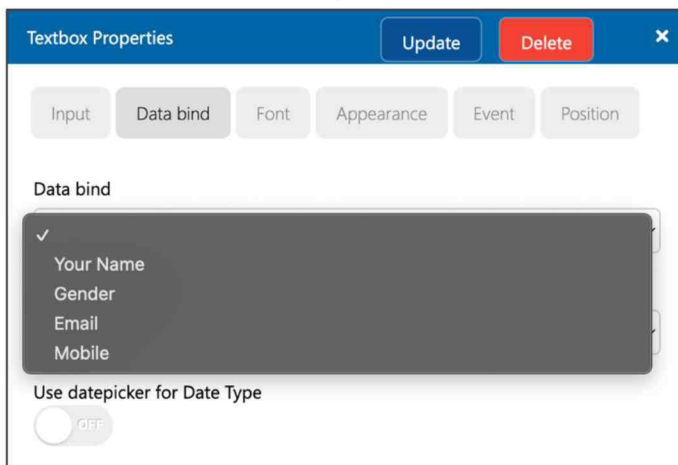
1. As discuss in Section A, you need to bind your eForm to Document Profile (Database Table) through data binding.
2. From the eForm Designer toolbar, click the gear icon. 

3. From Bind to Document Profile field, choose the “Customer Survey Form” from the drop-down list.

- Click the Update button to save your changes.
- Go to the Textbox1 and click the configuration icon.



- From the Data bind field, choose "Your Name".

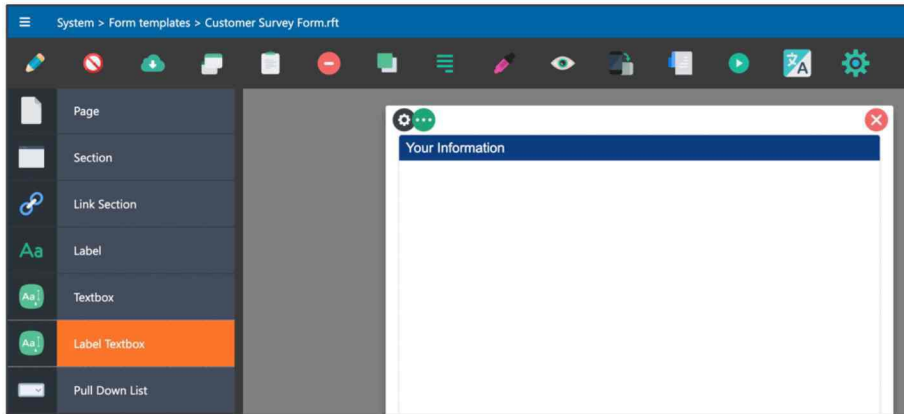


- Click the Update button to save your changes. When the user fills in the eForm, the text box value will be written to the document profile field "Your Name."

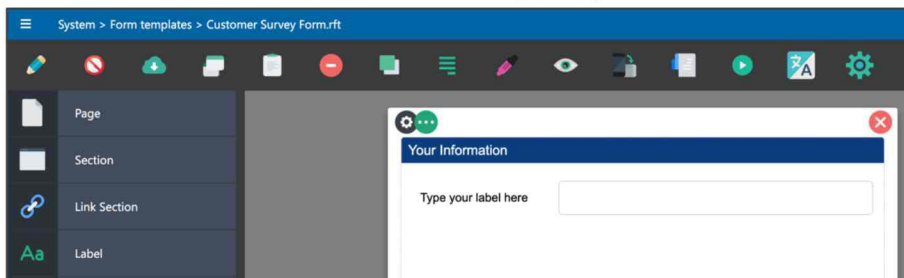


V. Label Textbox

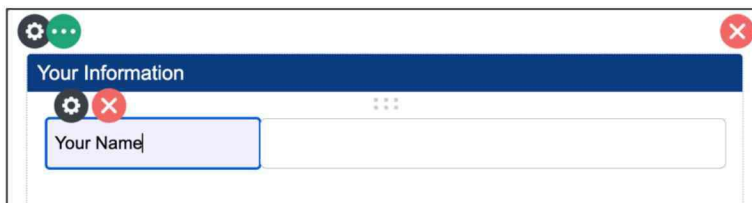
1. To create a text box with a label, you don't need to draw label and text box separately. The Label Textbox object provides a quick shortcut to build the entire set seamlessly.
2. Drag and drop the Label Textbox object onto the Section.



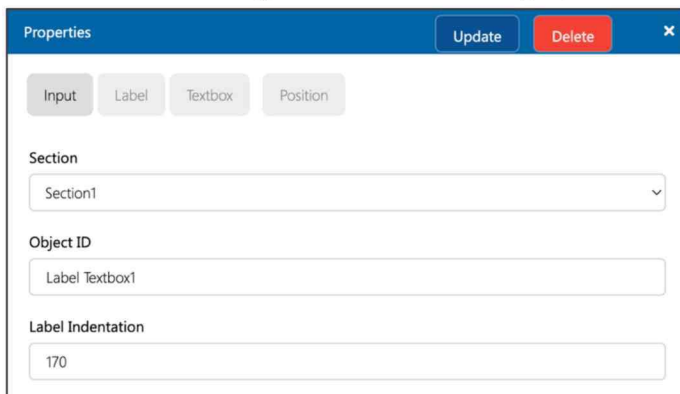
3. Both label and textbox are built as a single object.



4. Rename the label.

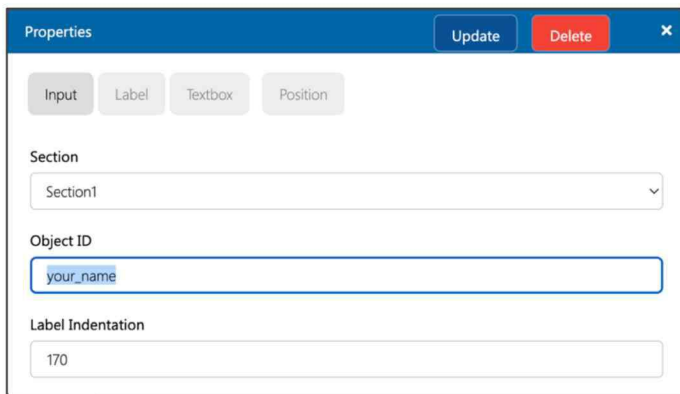


5. Then click the configuration icon to bring up the Label Textbox Properties window.



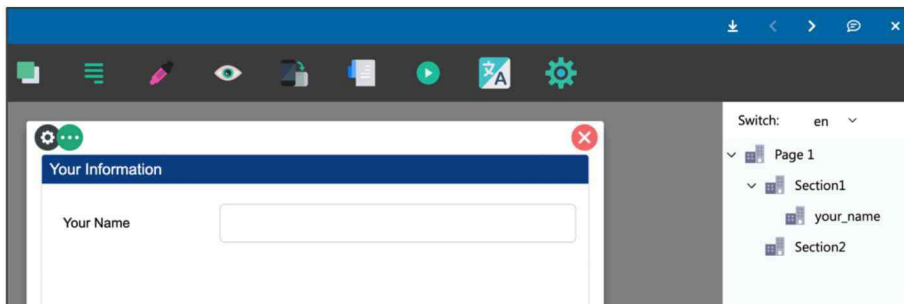
The 'Properties' window for a Label Textbox. It has a blue header with 'Update' and 'Delete' buttons. Below the header are four tabs: 'Input', 'Label', 'Textbox', and 'Position'. The 'Label' tab is selected. The 'Section' dropdown is set to 'Section1'. The 'Object ID' text field contains 'Label Textbox1'. The 'Label Indentation' text field contains '170'.

6. For easy managing, change the Object ID, then click the Update button to save your changes.



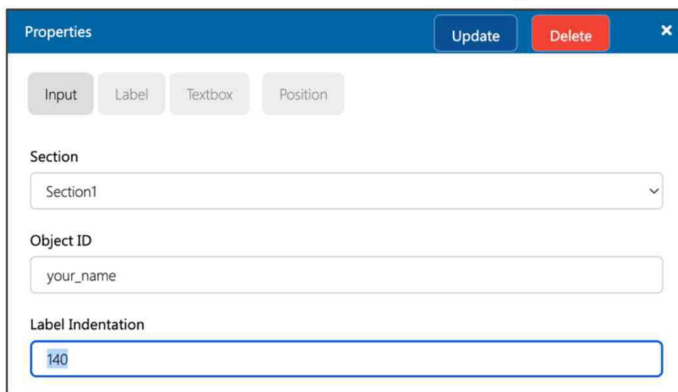
The 'Properties' window with the 'Object ID' field now containing 'your_name'. The 'Section' dropdown remains 'Section1' and 'Label Indentation' remains '170'.

7. The Object Explorer on the right-hand side will reflect the name changing.



The main workspace of the Antelope 6 Workspace Form Designer. On the left, a preview of a form titled 'Your Information' shows a 'Your Name' label and a text input field. On the right, the 'Object Explorer' panel shows a tree structure: 'Page 1' > 'Section1' > 'your_name' and 'Section2'. A 'Switch: en' dropdown is at the top of the Object Explorer.

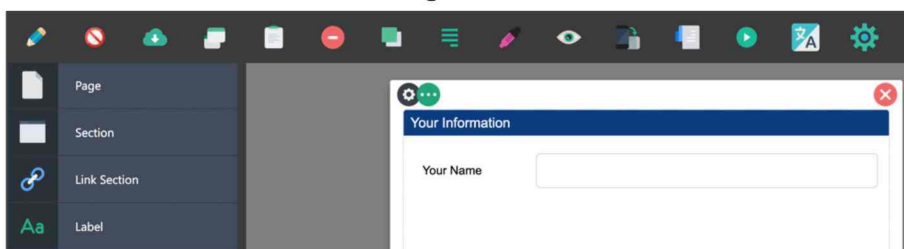
8. To reduce the label indentation, change the value.



The screenshot shows the 'Properties' panel for a 'Label' object. The panel has a blue header with 'Update' and 'Delete' buttons. Below the header are four tabs: 'Input', 'Label', 'Textbox', and 'Position'. The 'Label' tab is selected. The panel contains the following fields:

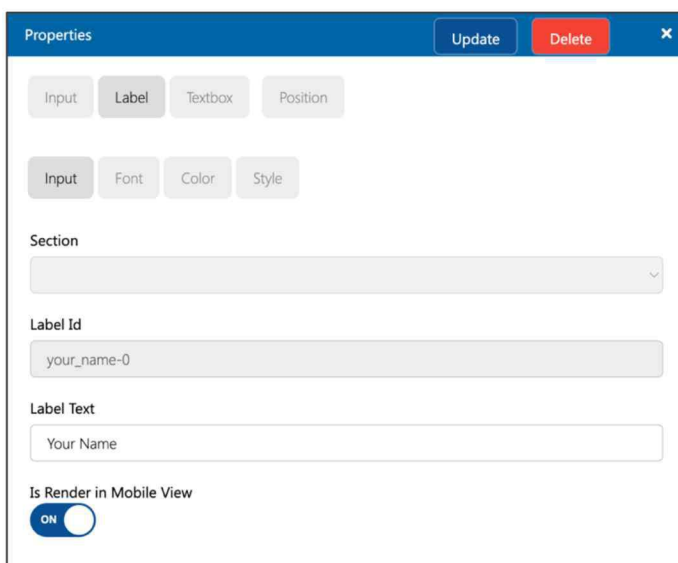
- Section:** A dropdown menu showing 'Section1'.
- Object ID:** A text field containing 'your_name'.
- Label Indentation:** A text field containing '140', which is highlighted with a blue selection box.

9. The label indentation will change.



The screenshot shows the workspace of the Antelope 6 Workspace Form Designer. On the left is a sidebar with a tree view containing 'Page', 'Section', 'Link Section', and 'Label'. The 'Label' item is highlighted. The main workspace area shows a form titled 'Your Information' with a blue header. Below the header is a label 'Your Name' followed by a text input field. The form is displayed within a window that has a toolbar at the top with various icons for editing and viewing.

10. The Label panel replicates the label object. Please refer to Section D.I – Label for details.



The screenshot shows the 'Properties' panel for a 'Label' object, with the 'Label' tab selected. The panel has a blue header with 'Update' and 'Delete' buttons. Below the header are two rows of tabs: the first row contains 'Input', 'Label', 'Textbox', and 'Position'; the second row contains 'Input', 'Font', 'Color', and 'Style'. The 'Label' tab is selected. The panel contains the following fields:

- Section:** A dropdown menu.
- Label Id:** A text field containing 'your_name-0'.
- Label Text:** A text field containing 'Your Name'.
- Is Render in Mobile View:** A toggle switch that is currently turned 'ON'.

11. While the Textbox panel replicates the textbox object. Please refer to Section D.II – Textbox for details.

The screenshot shows the 'Properties' panel for a 'Textbox' object. The panel has a blue header with 'Update' and 'Delete' buttons. Below the header are two rows of tabs: 'Input', 'Label', 'Textbox', and 'Position' in the first row; 'Input', 'Data bind', 'Font', 'Appearance', and 'Event' in the second row. The 'Textbox' tab is selected. The panel contains the following fields:

- Section:** A dropdown menu.
- Object ID:** A text field containing 'your_name-1'.
- Type:** A dropdown menu with 'Textbox' selected.
- Default Value:** A dropdown menu with 'No Default' selected.
- Mandatory if section is editable:** A toggle switch set to 'OFF'.
- Max Length:** A text field containing '200'.

12. One thing to keep in mind about object IDs: since the "Label Textbox" is a complex structure containing two bound objects (Label and Textbox), they have separate object IDs.

Object	Other Setting
Label Textbox	your_name
Label	your_name-0
Textbox	your_name-1

If you need to write something into the text box using the eForm script (Section F), your target object ID should be "your_name-1" instead of "your_name."

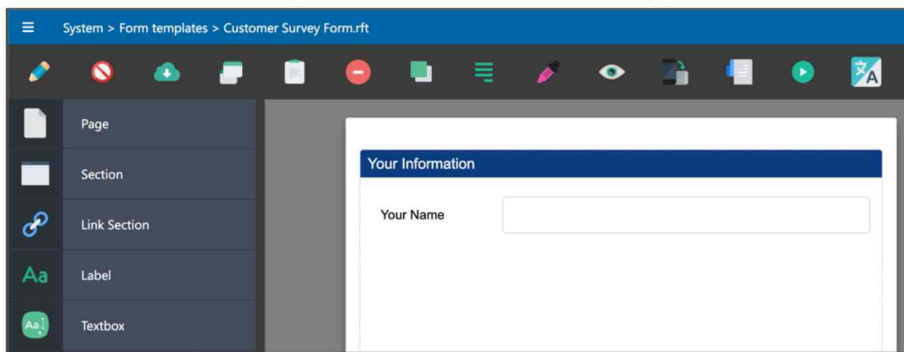
13. Other complex eForm objects, such as Label Pull-Down List and Label Combo Box, work on the same principle.

VI. Design, Preview and Mobile View

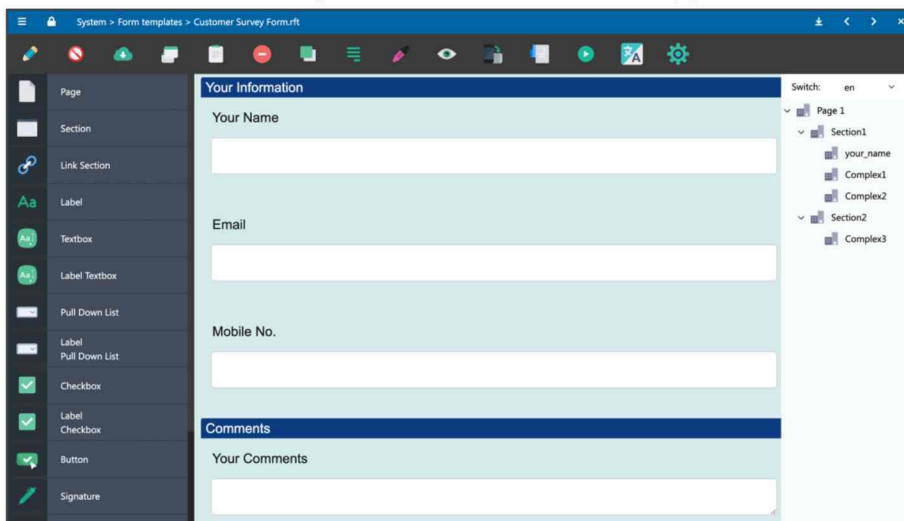
1. You can preview the eForm on both web and mobile platform. From the horizontal toolbar, click the design, preview and mobile icon.



2. Preview mode allows you to see the eForm as it appears in the user's workspace.

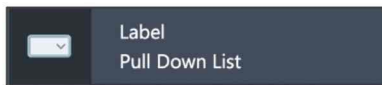


3. The mobile view allows you to see the eForm as it appears in mobile apps.

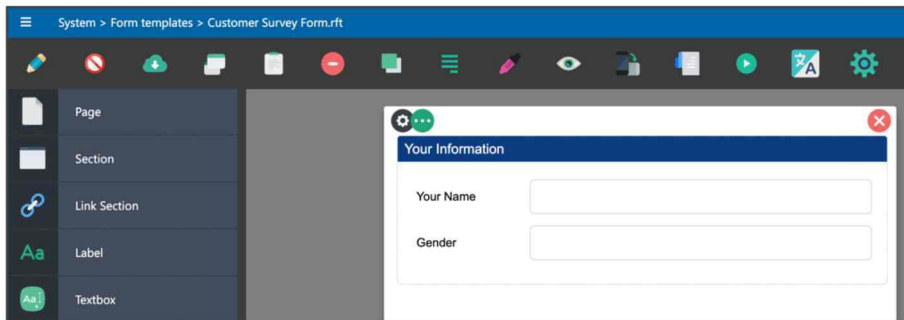


VII. Pull-Down List & Label Pull-Down List

1. Drag and drop the Label Pull-Down List object onto a Section.



2. The Pull-Down List object allows user to select from a list of option.



The pull-down list options can be defined by either one of the following methods:

- a. Create an option list manually
 - b. Bind to Data Source (Third party database)
 - c. Bind to Lookup List
3. To configure the Pull-Down List, click the configuration icon.



4. Input

 A screenshot of the Antelope 6 Workspace Form Designer interface showing the configuration window for the pull-down list. The window has a title bar 'Properties' with 'Update' and 'Delete' buttons. Below the title bar, there are tabs for 'Input', 'Label', 'Combo', and 'Position'. The 'Label' tab is selected. The configuration fields are:

- Section: A dropdown menu with 'Section1' selected.
- Object ID: A text field with 'gender' entered.
- Label Width: A text field with '140' entered.

Field Name	Description
Section	Section ID
Object ID	Object ID of the Label Pull-Down List
Label Width	The label indentation width

5. Label: A standard label control. For details, please refer to Section D.I: Label.

The screenshot shows the 'Properties' panel for a 'Label' control. The panel has a blue header with 'Properties', 'Update', 'Delete', and a close button. Below the header are tabs for 'Input', 'Label', 'Combo', and 'Position'. The 'Label' tab is selected. Underneath are sub-tabs for 'Input', 'Font', 'Color', and 'Style'. The 'Section' dropdown is empty. The 'Label Id' text field contains 'gender-0'. The 'Label Text' text field contains 'Gender'. At the bottom, the 'Is Render in Mobile View' toggle is turned 'ON'.

6. Combo: Input

The screenshot shows the 'Properties' panel for a 'Combo: Input' control. The panel has a blue header with 'Properties', 'Update', 'Delete', and a close button. Below the header are tabs for 'Input', 'Label', 'Combo', and 'Position'. The 'Combo' tab is selected. Underneath are sub-tabs for 'Input', 'Data bind', 'Selection List', 'Font', and 'Event'. The 'Section' dropdown is empty. The 'Combobox Id' text field contains 'gender-1'. Below this are three toggle switches: 'Mandatory if section is editable' (OFF), 'Default first item' (OFF), and 'Multiple Select' (OFF).

Field Name	Description
Section	For Label Pull-Down List object, the Section is disabled.
Combobox Id	Object ID of the Pull-Down List, pay attention to the object ID. If you need to deploy any script on the pull-down list, the object ID refer should be "-1". For example, in this case "gender-1".
Mandatory	Mandatory field when user has the rights to edit this section.
Default first item	By default, the first item of the pull-down list is selected.
Multiple Select	Allow multiple select of pull-down list options.

7. Combo: Data Bind

The screenshot shows the 'Properties' dialog for a 'Combo' field. The 'Data bind' tab is selected. It features two dropdown menus: 'Data bind' with 'Gender' selected, and 'Key Databind' which is currently empty. The dialog has 'Update' and 'Delete' buttons at the top right.

Field Name	Description
Data bind	Bind the object value to Document Profile, when you save the eForm, the value of this object will write to the index field of the Document Profile.
Key Data bind	When combo key and combo value are different, can bind to different index field.

8. Combo: Selection List (Part 1)

The screenshot shows the 'Properties' dialog for a 'Combo' field. The 'Selection List' tab is selected. It features a 'Data Source' dropdown menu which is empty, and an 'Input options' text area containing 'Male' and 'Female'. The dialog has 'Update' and 'Delete' buttons at the top right.

Field Name	Description
Input options	Define the pull-down list manually by inputting the options.
Data Source	Instead of building the pull-down list manually, you can connect the pull-down list to data source. The data source can be either by third party database connection or through Lookup list. For details, please refer to Antelope 6 Administration Manual.

9. For convenience, it will be easier to create a pull-down list for user selection manually. However, when recalling the system and database design in Section A.1:



Customer Survey Form

Your Name

Gender

Email

Mobile

Product Interest

☒ Product A ☒ Product B ☒ Product C

☒ Product D

Tell us your Feeling

Product Name	Desire of Purchase (0-5)
Product A	5
Product B	3
Product C	0
Product D	1

Your Comments

Sign Here 



Document Profile (Database Table)

Field	Field Type
Your Name	String (50)
Gender	String (1)
Email	String (255)
Mobile	String (20)
Feeling	Details Line Item
Field	Field Type
Product Name	String (10)
Desire of Purchase	Integer
Product Interest	String (255)
Comments	String (255)

The Gender pull-down list is “Male / Female”, however the document profile (database table) field which map to Gender is **String (1)**, which means only 1 character is allowed to store (M/F).

10. In this case, we need to create an Index Lookup Table (mapping table) for the eForm pull-down list to connect, which involves several steps:



Document Profile

Company	Document Profile	Level	Description
	Gender		

Company Lookup: Gender Index Lookup Index Field Table (Gender)

Step 1: Create an Index Lookup Document Profile as the “skeleton”.



Document Profile Fields of Lookup: Gender

Type	Name	Length	Format	Default Value	Rule
String	Full Name	10			
String	Prefix	1			

Step 2: Define the fields Full Name (Male/Female) and Prefix (M/F)



Index Lookup

Company	Name	Type	Document Profile	Sync With Group
	gender			

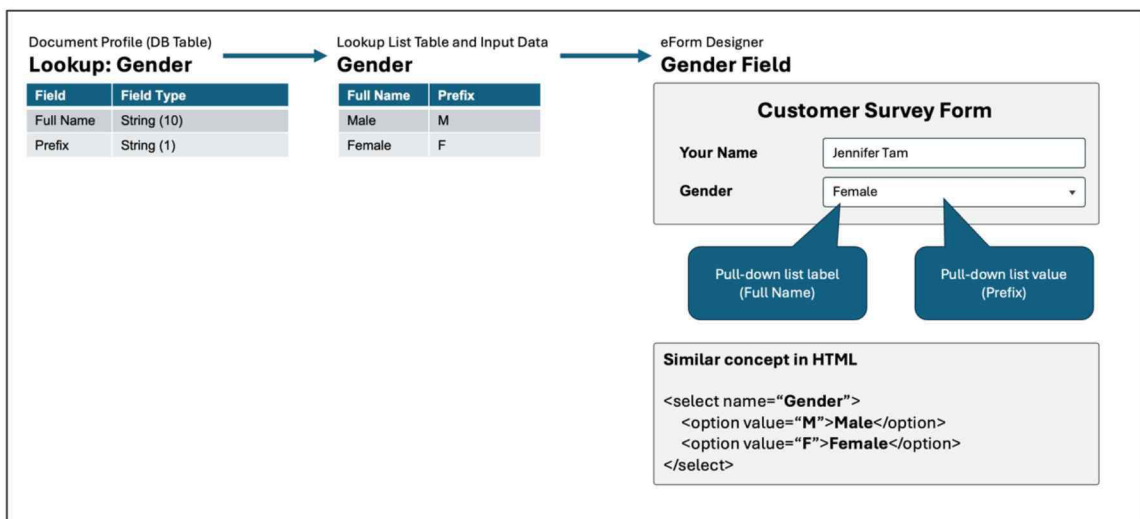
Company Gender Table Lookup: Gender

Step 3: Create an Index Lookup Table which use the Document Profile (Lookup: Gender) as skeleton.

Items						
	Full Name	Prefix	Owner	Modified Date	Created Date	
 						
  	Male	M	jennifer	2024-09-03 13:33:15	2024-09-03 13:33:15	
  	Female	F	jennifer	2024-09-03 13:33:19	2024-09-03 13:33:19	

Step 4: Input all data inside the Index Lookup Table Items

The following diagram explains the relationship between Document Profile, Lookup List and eForm Object:



11. Combo: Selection List (Part 2)

Properties

Update Delete

×

Input Label Combo Position

Input Data bind Selection List Font Event

Data Source

Gender

Input options

Key field name

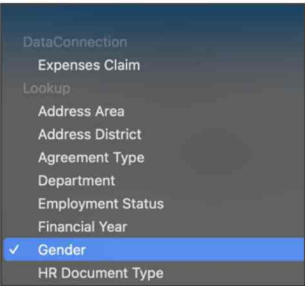
Prefix

Display field name

Full Name

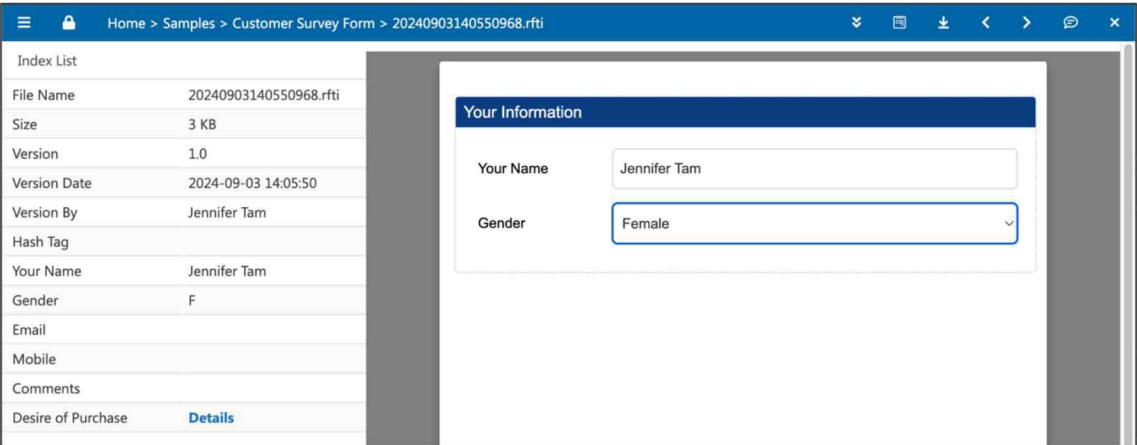
Back to the eForm Designer, instead of manually input Male / Female in Input

Options, you can now connect the pull-down list to Data Source: Gender.

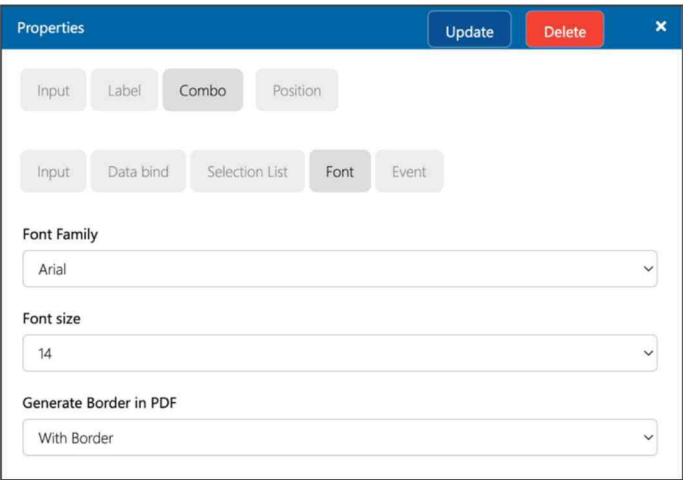


Field Name	Connect to	Result
Display field name	Full Name	Male, Female
Key field name	Prefix	M, F

After configuring the Selection List, when you fill-in the Customer Survey Form and save, the index field value (left-hand side panel) of Gender is saved as “F” instead of “Female”.



12. Combo: Font



Field Name	Description
Font Family	Define the font type
Font Size	Define the font size
Generate Border in PDF	Include pull-down list border when export the eForm as PDF file.

13. Combo: Event

Properties

UpdateDelete×

InputLabelComboPosition

InputData bindSelection ListFontEvent

[Open Script Editor](#)

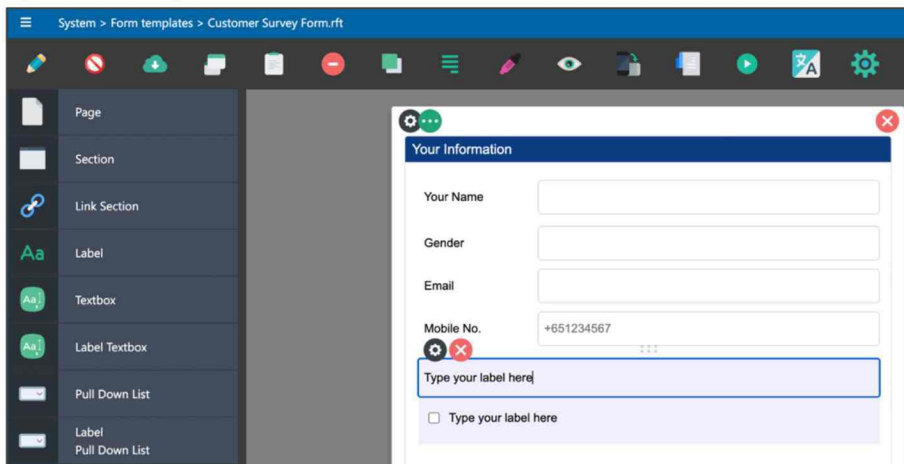
Field Name	Description
Open Script Editor	This triggers the script editor, for details, please refer to Section F.

VIII. Label Checkbox

1. Drag and drop the Label Checkbox object onto a Section.



2. The Label Checkbox object is added to the eForm section. Click the configuration icon to open the Properties window.



3. Click the configuration icon to bring up the Properties window.
4. Checkbox: Input

 A screenshot of the 'Properties' window in the Antelope 6 Workspace Form Designer. The window has a title bar with 'Properties' and buttons for 'Update', 'Delete', and a close icon. Below the title bar, there are four tabs: 'Input', 'Label', 'Complex', and 'Position'. The 'Input' tab is selected. The 'Section' dropdown menu is set to 'Section1'. The 'Object ID' text field contains the value 'product_request'.

Field Name	Description
Section	Section ID
Object ID	Object ID of the Label Checkbox

5. Label: A standard label control. For details, please refer to Section D.I: Label.

Properties [Update] [Delete] X

Input Label Complex Position

Input Font Color Style

Section

Label Id
product_request-0

Label Text

Is Render in Mobile View
ON

Let's change the label to "Product Interest" first.

Product Interest

☐ Type your label here

6. Complex: Input (Part 1)

Properties [Update] [Delete] X

Input Label Complex Position

Input Label Style Position

Section

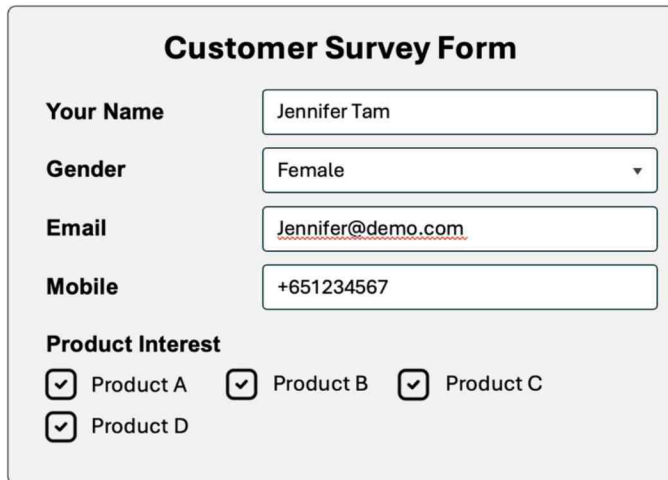
Checkbox Id

Data Binding
Product Interest

Field Name	Description
Section & Checkbox ID	These fields are not using for Label Checkbox complex object.
Data bind	Bind the object value to Document Profile, when you save the eForm, the value of this object will write to the index field of the Document Profile.

According to the form design in Section A, the Product Interest is a multiple-choice question which contains four checkboxes in two rows. We expect users can click

multiple checkboxes at the same time and the document profile will save the choices selection.



Customer Survey Form

Your Name

Gender

Email

Mobile

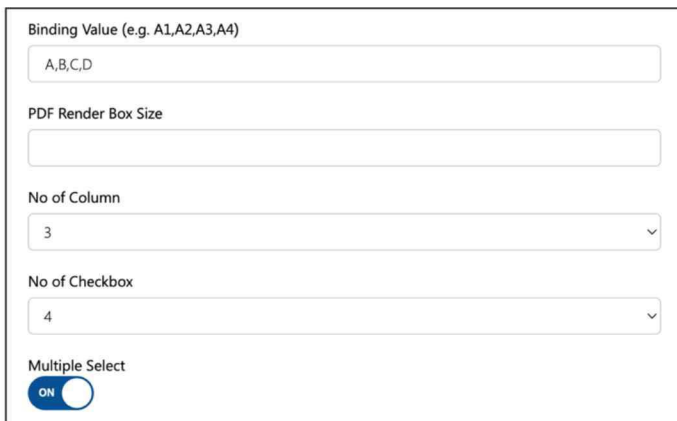
Product Interest

☒ Product A ☒ Product B ☒ Product C

☒ Product D

There in the Properties window, we need to define the binding value of the four choices (i.e. A,B,C,D). It depends on what answer you want to store in the profile.

As seen from the form design above, there are three columns and four checkboxes, therefore the “No. of Column = 3” and “No. of Checkbox = 4”.



Binding Value (e.g. A1,A2,A3,A4)

PDF Render Box Size

No of Column

No of Checkbox

Multiple Select

☒ ON

Once the setting is completed, input the label on the Label Checkbox.

Product Interest

☐ Product A ☐ Product B ☐ Product C

☐

Field Name	Description
Binding Value	The value of checkbox to save in Document Profile after user select.
PDF Render Box Size	The size of checkbox when render in PDF.
No. of Column	Number of column checkbox to arrange.
No. of Checkbox	Total number of checkboxes to display.
Multiple Select	Allow the user to select multiple checkboxes.

7. Complex: Label

Properties Update Delete ×

Input Label **Complex** Position

Input Label **Style** Position

Font Family

Font size

Text Spacing Ratio

Field Name	Description
Binding Value	The value of checkbox to save in Document Profile after user select.

8. Complex: Style

Properties

UpdateDelete✕

InputLabelComplexPosition

InputLabelStylePosition

Color

Transparent

Background Color

Transparent

Bold

OFF

Italic

OFF

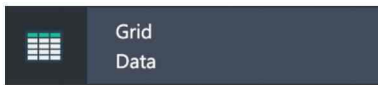
Underline

OFF

Field Name	Description
Color	Label color
Background Color	Label background color Product A Product B
Bold, Italic, Underline	Label font style Product A Product B

I. Grid Data

1. Drag and drop the Data Grid object onto a Section.



2. A new Grid object is added to the eForm.

3. To enable Grid Data object, your Document Profile must have a Detail field type.

	Type	Name	Length
⊕			
🗑️ ⬆️ ⬇️	String	Your Name	50
🗑️ ⬆️ ⬇️	String	Gender	1
🗑️ ⬆️ ⬇️	String	Email	255
🗑️ ⬆️ ⬇️	String	Mobile	50
🗑️ ⬆️ ⬇️	String	Product Interest	10
🗑️ ⬆️ ⬇️	String	Comments	255
🗑️ ⬆️ ⬇️	Detail	Tell Us Your Feeling	8

4. In this example, the grid (Tell us your Feeling) consists of a table of two columns: Product Name, Desire of Purchase.

Customer Survey Form

Your Name

Jennifer Tam

Gender

Female

Email

Jennifer@demo.com

Mobile

+651234567

Product Interest

☒ Product A

☐ Product B

☒ Product C

☒ Product D

Tell us your Feeling

Product Name	Desire of Purchase (0-5)
Product A	5
Product B	3
Product C	0
Product D	1

Your Comments

I feel great on your show, here are some of my comments ...

Sign Here



Therefore the Document Profile Detail should include the following fields:

Document Profile Fields of Customer Survey Form Line Items									
					Type	Name	Length		
							</		

- Click the configuration icon  to show the Properties Window. Map the Grid to the Document Profile detail field “Tell us your Feeling”.

Grid Properties

UpdateDelete

✕

InputPosition

Section

Section1

Id

Grid

Row

5

Enable Dynamic Row

OFF

Enable Row Count Column

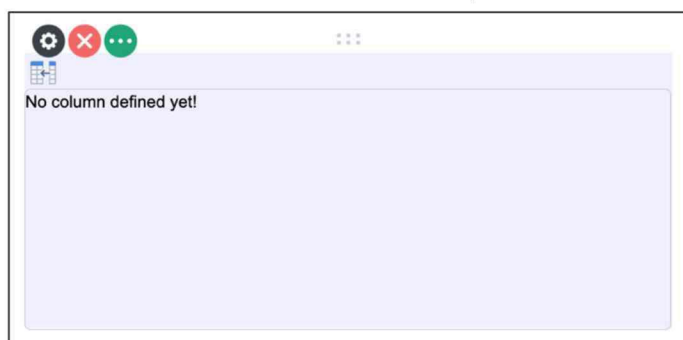
OFF

Grid Data Binding Detail Field

Tell Us Your Feeling

Field Name	Description
Section	The eForm Section
Id	Grid Data Object Id
Row	Number of rows by default, the minimum is 5 rows. 
Enable Dynamic Row	User can increase or decrease the no. of rows by clicking the   icons. 
Enable Row Count	Each row will have a counter when this option is enabled.  <i>Enable row count</i>  <i>Disable row count</i>
Grid Data Binding Detail Field	Data bind to Document Profile field.

6. To add the first column in the Grid, click the  icon.

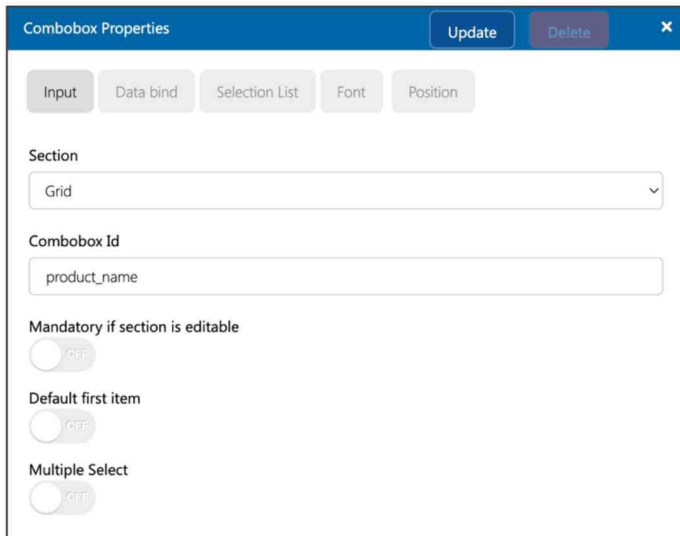


- The Create New Column window will appear. There are four object types you can add to the Grid Data object: Label, Textbox, Combo and Checkbox. In this example, we choose Combo. Click OK button to save your changes.



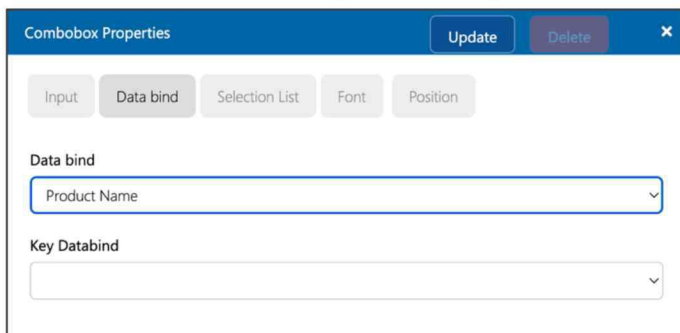
The 'Create New Column' dialog box has a blue header with the title 'Create New Column' and buttons for 'OK' and a close 'X'. Below the header, there is a label 'Select which Column type' followed by a dropdown menu. The dropdown menu is currently set to 'Combo'.

- The Combobox Properties window will appear. Set the object id as “product_name”.



The 'Combobox Properties' dialog box has a blue header with the title 'Combobox Properties' and buttons for 'Update', 'Delete', and a close 'X'. Below the header, there are five tabs: 'Input', 'Data bind', 'Selection List', 'Font', and 'Position'. The 'Input' tab is selected. Under the 'Section' label, there is a dropdown menu set to 'Grid'. Under the 'Combobox Id' label, there is a text input field containing 'product_name'. Below this, there are three toggle switches: 'Mandatory if section is editable' (set to OFF), 'Default first item' (set to OFF), and 'Multiple Select' (set to OFF).

Set the data bind to “Project Name”, which is the Document Profile Detail field.



The 'Combobox Properties' dialog box is shown with the 'Data bind' tab selected. The 'Data bind' dropdown menu is set to 'Product Name'. Below it, the 'Key Databind' dropdown menu is empty.

Manually input Product A ... Product D in the Selection List.

The screenshot shows the 'Combobox Properties' dialog box with the 'Selection List' tab selected. The 'Data Source' dropdown is empty. The 'Input options' list contains the following items:

- Product A
- Product B
- Product C
- Product D

9. Click Update button to save your changes. The first column is now created.

The screenshot shows the workspace with a grid component. The grid is labeled 'Grid.product_name' and has a settings icon and a close icon. The grid has one column and five rows.

10. Repeat Step 6 – Step 9 with the following settings:

Setting Name	Value
Object Type	Combo
Object ID	desire_of_purchase
Data Bind	Desire of Purchase
Selection List: Input options	0 1 2 3 4 5

11. The remaining column is also created.

12. The column title and width are not yet defined. Click the  icon to open the Grid Detail window.

Grid Detail										
	ID	Label	Width	Sum total	Total Data Binding	Hidden	Rowspan	Combo Relate Column Index	Formula	Nun
				product_name	200					
				desire_of_purchase	200					

13. Input the label and adjust the width, the save your changes.

Grid Detail										
	ID	Label	Width	Sum total	Total Data Binding	Hidden	Rowspan	Combo Relate Column Index	Formula	Nun
				product_name	Product Name	250				
				desire_of_purchase	Desire of Purchase (0-5)	180				

14. Now you can select different answers in both columns and rows.

Tell us your feeling:

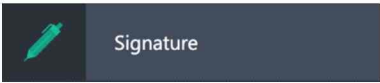
	Product Name	Desire of Purchase (0-5)
1	Product A	5
2	Product B	2
3	Product C	0
4	Product D	1 
5		

15. Apart from label and width, Grid Detail has other advanced setting. Please refer to Section J: Advanced Grid Data Setting.

Grid Detail									
Label	Width	Sum total	Total Data Binding	Hidden	Rowspan	Combo Relate Column Index	Formula	Numeric Mask	

II. Signature

1. Drag and drop the Signature object onto a Section.



2. The Signature object is placed on the eForm Section.

Tell us your feeling:

	Product Name	Desire of Purchase (0-5)
1		
2		
3		
4		
5		

Sign Here

Sign Here

3. Click the configuration icon to open the Signature Properties window.

Properties Update Delete ✕

Input

Title

Signature

Position

Section

Section1

Object ID

Signature Complex1

Display Signer Name

☐

Display Signing Date Time

☐

Field Name	Description
Section	Section ID
Object ID	Object ID of the Signature
Display Signer Name Display Signer Date Time	Display both signer name and date time when user signs Sign Here Sign Here Signed by Signed on Sign Here Signed by Jennifer Tam Signed on 2024-09-04 01:41:04

4. The Signature Control

Properties

Update Delete

Input Title Signature Position

Input Event

Section

Signature Mode

Predefined

Label Id

Signature Complex1-1

Data bind

Mandatory if section is editable

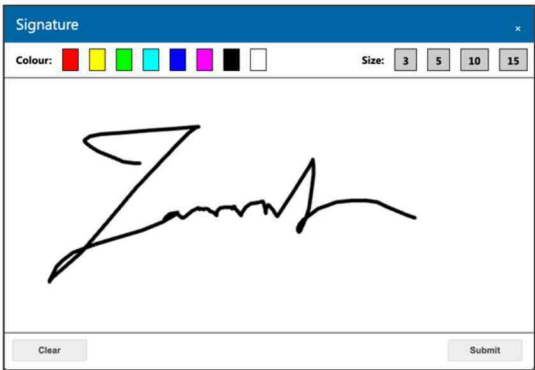
OFF

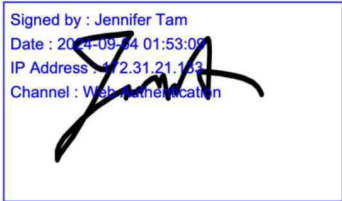
Generate Border in PDF

Default to Form Setting

Render Value In PDF

ON

Field Name	Description
Section	This field is disabled.
Signature Mode: Password	User is required to input the password and will paste the pre-upload signature image to the sign-area. 
Signature Mode: Drawing	User is required to draw the signature. 
Data bind	Data bind to Document Profile field name.

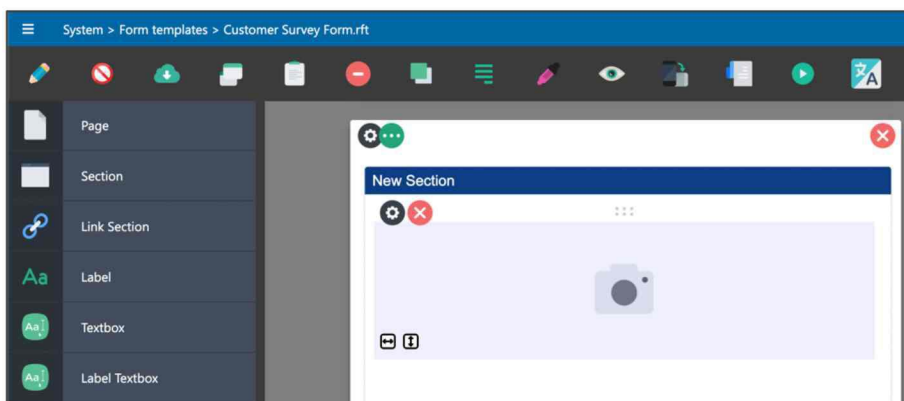
Mandatory if section is editable	Set the Signature area to mandatory.
Generate Border in PDF	The signature area will have a border when the eForm is export as PDF file.
Render Value in PDF	The signature information will have text value generated in the PDF Sign Here 

III. Image

- The Image object serves two purposes:
 - Allow user to upload an image to eForm (e.g. Site inspection report)
 - Pre-embedded an image to eForm (e.g. company logo)
- Drag and drop the Image object onto a Section.



- A new Section is added.



4. Click the configuration icon to open the Properties window.

Field Name	Description
Section	Section Id
Image Id	Object Id
Enable Border	Enable image object with border
Style	CSS style. For example, background-color: #eeeeee
Mandatory if section is editable	Image field is mandatory
Generate Border in PDF	When the eForm export as PDF, whether the object contains border
Enable User Upload Image	Allow user to click the object to upload image, or trigger the camera function on mobile app

5. When a user uploads an image to the Image object, they can adjust the view to either fit-to-width or fit-to-height. It is not recommended to embed a very high-resolution photo, as the photo data is embedded into the eForm, which will significantly increase its size.



IV. Index Lookup

- 1. Index Lookup is similar to a Combo Box; however, the former can display multiple pieces of information in a drop-down list, while the latter can only display a single value

Combo Box	Index Lookup
Gender Male ▾	SampleCorp Innovation INV24001 SampleCorp Innovation \$300.0000 INV24002 SampleCorp Innovation \$800.0000

- 2. In this example, you may have different invoices stored in either Antelope 6 repository or third-party databases.

Home > Invoices								
+ New ▾ Upload Index ▾ View ▾ Change Column View Recycle Bin								
☐		File Name	Invoice No.	Invoice Date	PO No.	Due Date	Supplier Name	Total Amount Line Items
☐		329504.pdf	329594-03	2014-08-30	6546451		The Private Wine	799.0000 Details (4)
☐		329505.pdf	329600-01	2017-05-02	7546005		The Private Wine	8888.0000 Details (2)
☐		INV24001.pdf	INV24001	2024-06-10	PO123016	2024-07-10	SampleCorp Innovation	300.0000 Details (2)
☐		INV24002.pdf	INV24002	2024-06-24	PO123017	2024-07-24	SampleCorp Innovation	800.0000 Details (4)
☐		PT302923.pdf	PT302925	2018-10-30			佳能果欄	720.0000 Details (1)
☐		S15343-20.pdf	S15343-20	2022-09-15	PT10293	2022-10-15	森普敦科技有限公司	5000.0000 Details (1)

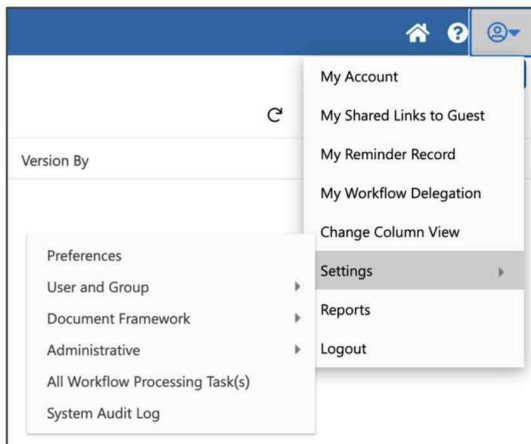
You want to create a drop-down list in eForm so that user can search the record by either Invoice No., Supplier Name and Total Amount.

SampleCorp Innovation

INV24001
SampleCorp Innovation
\$300.0000

INV24002
SampleCorp Innovation
\$800.0000

- The first step is to create a Database Connection, select “Settings > Administrative > Database Connection”.



- From the Database Connection, select your local Antelope repository. If you do not have this connection create, you can create one with Server = localhost, no account and database is required. For more information, please refer to Antelope 6 Administration Manual.

Database Connection						
	Server Type	Name	Server	Database	Account Name	Password

	System	Antelope (Local)	localhost			*****

- Click the detail icon  to bring up the DataSource window, then click  icon to add a new DataSource.

DataSource				
	Name	GUID	Query or Folder Path	Field List
	Expenses Claim	1ea92706550e45509bb1fb8c8d2158af	Workflow\Expenses Claim	Apply By,Apply Date,Expenses

- By referring to Step 2, the invoice data are stored in Home\Invoices folder, therefore:
 - DataSource Type = Folder
 - Folder Path = Home\Invoices

Add

Update

×

Name

Invoices

Type

Folder

Folder Path

Home\Invoices

Field List

Criteria

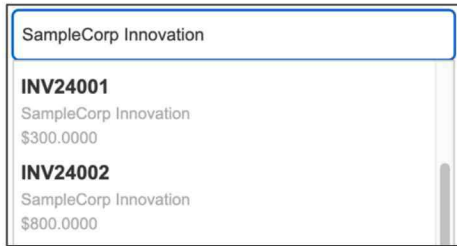
7. The Index Lookup drop-down list shows information including Invoice No., Supplier Name and Total Amount, therefore select these three index fields in the Field List option.

8. Click Update button to save your changes, a new DataSource is created.

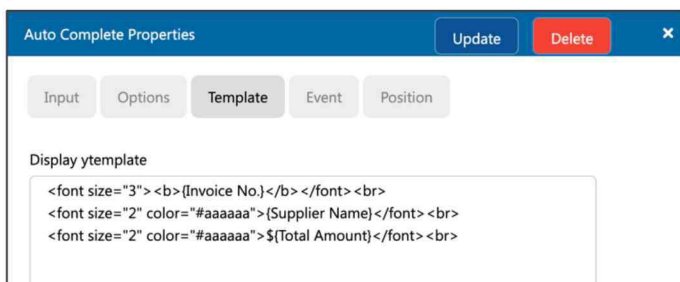
DataSource				
	Name	GUID	Query or Folder Path	Field List
	+			
	Invoices	2e240f318ce84e159c672cb3688cc47a	Home\Invoices	Invoice No.,Supplier Name,Total Amount
	Expenses Claim	1ea92706550e45509bb1fb8c8d2158af	Workflow\Expenses Claim	Apply By,Apply Date,Expenses Category,Am

9. Back to the Index Lookup Properties window from eForm Designer. Connect the Data Source to “Invoices”, which you have created from Step 4-8. The Key field name, which you plan to save to index field = Invoice No.

10. To set the Index Lookup List to display multiple index fields, you need to define the “Template”.



11. Go to Template and write the HTML code. Bear in mind the Index Lookup List does not support Javascript nor HTML5. Try to make the drop-down list as simple as possible.



V. Folder Selector

1. The Folder Selector object allows user to select the Antelope 6 repository folder path, when click on the



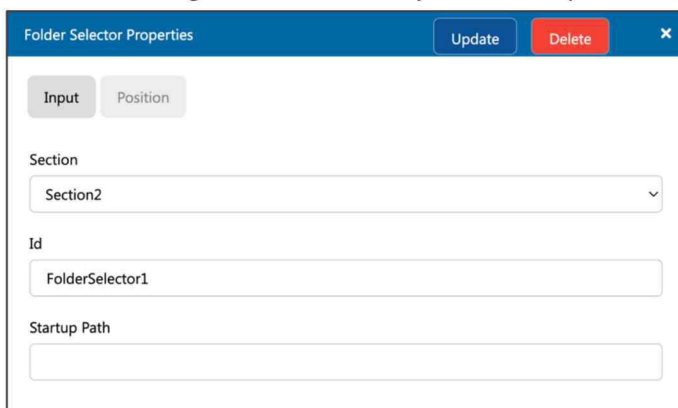
2. Drag and drop the Folder Selector object onto a Section.



3. A new Folder Selector object is added.



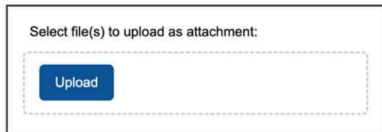
4. Click the configuration icon to open the Properties window.



Field Name	Description
Section	Section Id
Id	Folder Selector object Id
Startup Path	The starting location of folder when user click the folder selector. For example: Home\Legal\Agreement

VI. File Uploader

1. File Uploader allows user to select file(s), and upload to workflow as attachment(s).



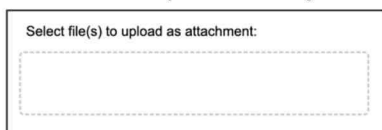
Select file(s) to upload as attachment:

Upload

2. Drag and drop the File Uploader object onto a Section.

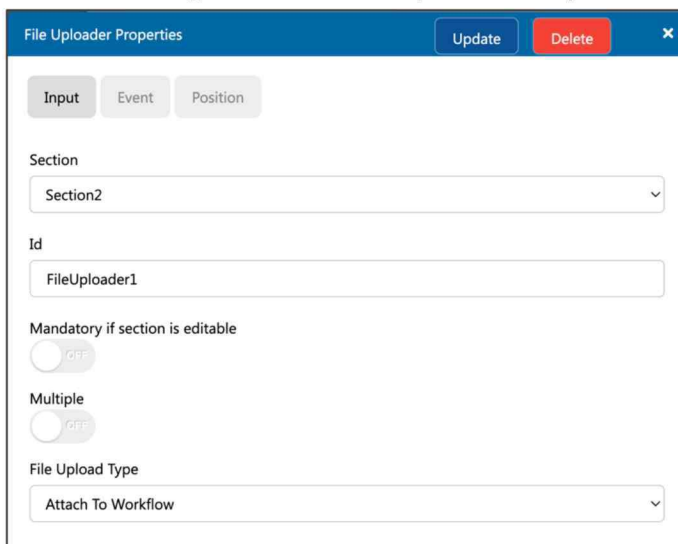


3. A new File Uploader object is added.



Select file(s) to upload as attachment:

4. Click the configuration icon to open the Properties window.



File Uploader Properties

Update Delete

Input Event Position

Section

Section2

Id

FileUploader1

Mandatory if section is editable

OFF

Multiple

OFF

File Upload Type

Attach To Workflow

Field Name	Description
Section	Section Id
Id	File Uploader object Id
Mandatory if section is editable	The File Uploader is mandatory
Multiple	Allows multiple files upload
File Upload Type	Attach to Workflow: Attach the file to workflow as attachment. Trigger Callback to Excel: Import Excel spreadsheet to Data Grid

5. To trigger callback to Excel like below:

	A	B	C	D	E	F	G	H	I
	SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country	Phone	
1	1	Exotic Liquid	Charlotte Cooper	49 Gilbert St.	Londona	EC1 4SD	UK	(171) 555-2222	
2	2	New Orleans Cajun Delights	Shelley Burke	P.O. Box 78934	New Orleans	70117	USA	(100) 555-4822	
3	3	Grandma Kelly's Homestead	Regina Murphy	707 Oxford Rd.	Ann Arbor	48104	USA	(313) 555-5735	

You need to create a data grid.

SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country	Phone

Using the eForm Script Editor, write the following Javascript to map each data grid column to Excel column.

```
//the first argument will be the excel data in an array
var a = arguments[0];

//get the grid object
var objGrid = eform.get('Grid1');

//loop the data and set to grid
for (i = 0; i < a.length; i++) {
    objGrid.setData(i, 0, a[i].SupplierID);
    objGrid.setData(i, 1, a[i].SupplierName);
    objGrid.setData(i, 2, a[i].ContactName);
    objGrid.setData(i, 3, a[i].Address);
    objGrid.setData(i, 4, a[i].City);
    objGrid.setData(i, 5, a[i].PostalCode);
    objGrid.setData(i, 6, a[i].Country);
    objGrid.setData(i, 7, a[i].Phone);
}
```

Finally trigger a workflow, upload the excel to the file uploader, the data will be imported to the grid.

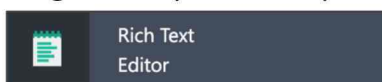
SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country	Phone
1	Exotic Liquid	Charlotte Cooper	49 Gilbert St.	Londona	EC1 4SD	UK	(171) 555-2222
2	New Orleans Cajun Delights	Shelley Burke	P.O. Box 78934	New Orleans	70117	USA	(100) 555-4822
3	Grandma Kelly's Homestead	Regina Murphy	707 Oxford Rd.	Ann Arbor	48104	USA	(313) 555-5735

VII. Rich Text Editor

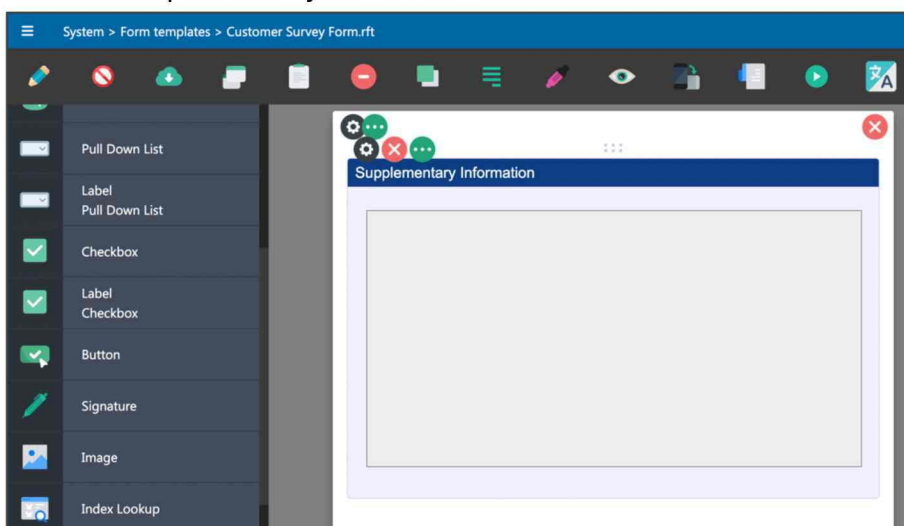
1. Rich Text Editor object allows user to input rich text in eForm, however, unlike other text-based object, Rich Text Editor cannot write into index field, as text displayed inside is in HTML format.



2. Drag and drop the File Uploader object onto a Section.



3. A new File Uploader object is added.



4. Click the configuration icon to open the Properties window.

Field Name	Description
Section	Section Id
Id	Rich Text Editor object Id

E. Script Library

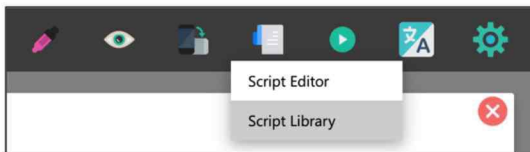
I. Background

Antelope 6 eForm Designer allows you to write customized JavaScript for front-end user interface. However, they are some build-in script libraries which you can add-on without any scripting. The following is a list of Script Library which you can use in the designer.

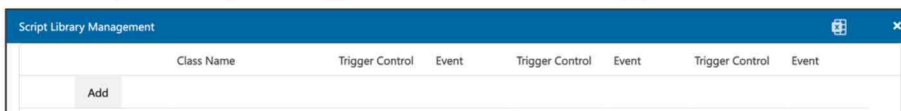
Category	Function	Description
DateTime	SubstractWithHoliday	Calculate the exact days between two given dates, by eliminate all public holiday
Image	QRCode	Generate QRCode by textbox value
Maths	MultiplyValue	Multiply two values from textboxes or pull-down lists
	SumValue	Adding two values from textboxes or pull-down lists
UI	CheckboxControl	When user click a checkbox, will enable or disable other objects.
	ComboboxControl	When user select an option from pull-down list, will enable or disable other objects.

II. Create New Library

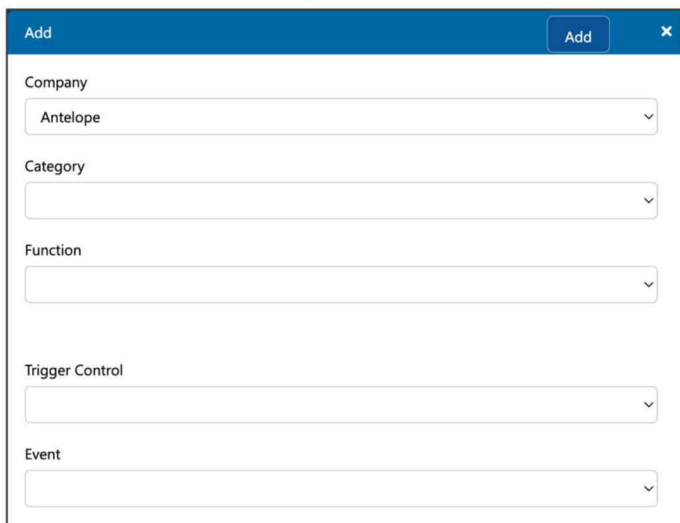
1. To create a new script library, select Script Library.



2. The Script Library Management window will appear. Click the Add button.



3. The Add window will appear. Select “Antelope” from the Company field.



The screenshot shows a modal window titled "Add" with a blue header bar containing the word "Add" and a close button (X). The window contains five dropdown menus, each with a label and a selection box:

- Company:** The dropdown menu is open, showing "Antelope" as the selected option.
- Category:** The dropdown menu is empty.
- Function:** The dropdown menu is empty.
- Trigger Control:** The dropdown menu is empty.
- Event:** The dropdown menu is empty.

III. Image.QRCode

1. Given an eForm with a Textbox, Button and Image Objects.

2. Create a new Script Library action with the following setting. Click Add button to save your changes.

3. Click the Params button to setup parameters.

Class Name	Trigger Control	Event
Antelope.Image.QRCode	Button1	onclick

4. The value is from the textbox, while the QR Code will be generated on the image object. Click Save button to save your changes.

Value Control	Textbox1-1
Target Image	Image1

5. After user input the value in the textbox and click the green button.

Script Library > Image > QRCode

Input a Value

[Generate QR Code](#)

[Click to Add Photo](#)



6. eForm will auto create the QR Code on the image object.

Script Library > Image > QRCode

Input a Value

[Generate QR Code](#)



IV. Maths.MultiplyValue

1. Given an eForm with three textboxes: Multiply_Value1, Multiply_Value2, Multiply_Result1.

Script Library > Maths > MultiplyValue

Value 1	
Value 2	
Result	

2. Create a new Script Library action with the following setting. When your cursor move to Result textbox, it will calculate the multiplication of Value1 x Value2 = Result.

Script Library Management

Add

Class	Company
	Antelope
Category	Math
Function	MultiplyValue
Multiply controls value	
Trigger Control	Multiply_Result1-1
Event	onfocus

3. Click the Params button to setup parameters.

Script Library Management

	Class Name	Trigger Control	Event
Add			
Remove	Antelope.Math.MultiplyValue	Multiply_Result1-1	onfocus

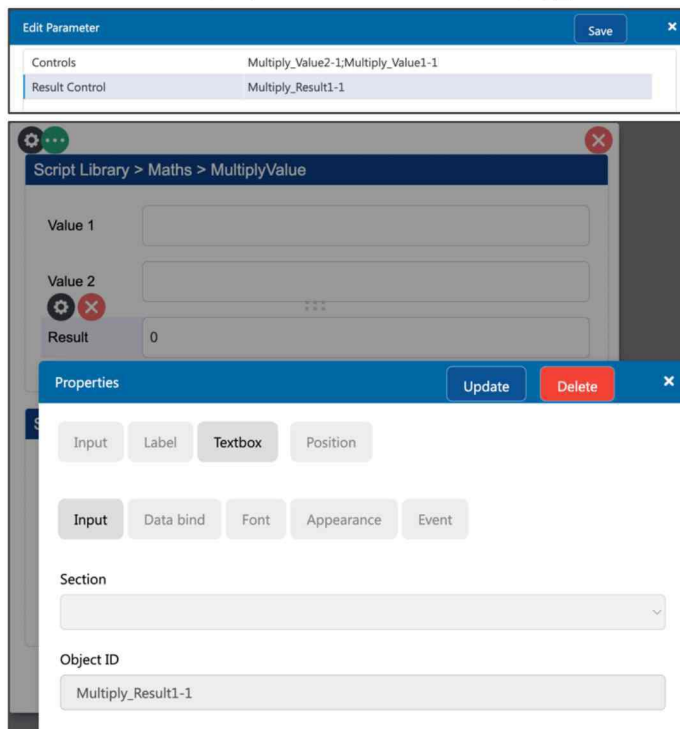
4. The multiplication is controlled by textboxes Multiply_Value1, Multiply_Value2.

Edit Parameter

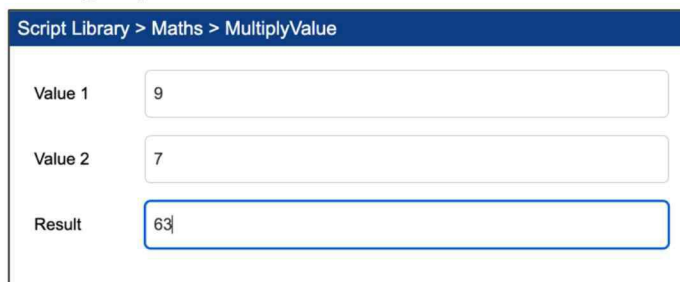
Save

Controls	Multiply_Value2-1;Multiply_Value1-1
Result Control	☐ Multiply_Result1-1 ☒ Multiply_Value2-1 ☒ Multiply_Value1-1

5. The result is displayed on textbox Multiply_Result1-1



6. Testing: Input 9 x 7 = 63



V. Maths.SumValue

1. Drag and drop the label object onto the first Section. Given an eForm with three textboxes: Sum_Value1, Sum_Value2, Sum_Result1.

Script Library > Maths > SumValue

Value 1

Value 2

Result

2. Create a new Script Library action with the following setting. When your cursor move to Result textbox, it will calculate the multiplication of Value1 x Value2 = Result.

Script Library Management

Add

Class: Antelope

Category: Math

Function: SumValue

Sum controls value

Trigger Control: Sum_Result2-1

Event: onfocus

3. Click the Params button to setup parameters.

Script Library Management

	Class Name	Trigger Control	Event
Remove	Antelope.Math.SumValue	Sum_Result2-1	onfocus

4. The multiplication is controlled by textboxes Sum_Value1, Sum_Value2.

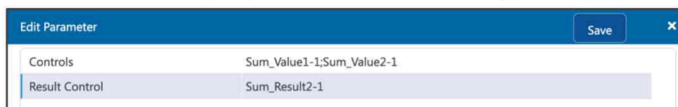
Edit Parameter

Controls: Sum_Value1-1;Sum_Value2-1

Result Control

- ☐ Multiply_Result1-1
- ☐ Multiply_Value2-1
- ☐ Multiply_Value1-1
- ☒ Sum_Value1-1
- ☒ Sum_Value2-1

5. The result is displayed on textbox Sum_Result1-1



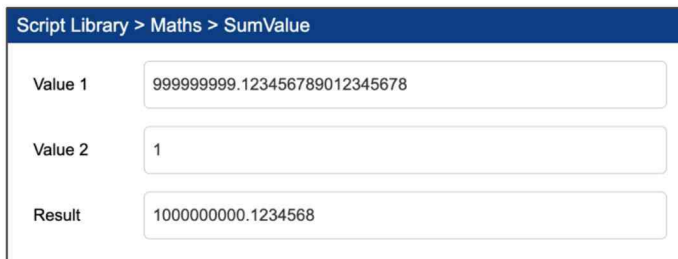
Edit Parameter	
Controls	Sum_Value1-1;Sum_Value2-1
Result Control	Sum_Result2-1

6. Testing: Input 9 x 7 = 63



Script Library > Maths > SumValue	
Value 1	9.2
Value 2	7
Result	16.2

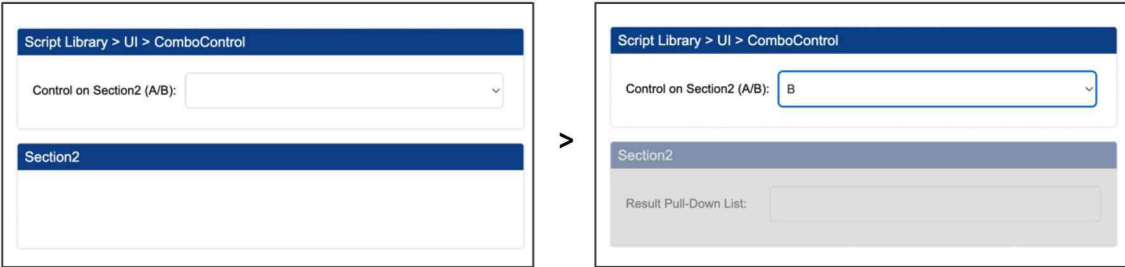
7. Remind both multiplication and addition can only support up to 7 decimal digits.



Script Library > Maths > SumValue	
Value 1	999999999.123456789012345678
Value 2	1
Result	1000000000.1234568

VI. UI.ComboboxControl

- 1. The UI.ComboboxControl allows user to enable / disable any objects (even the whole section) when they select a value from the pull-down list.

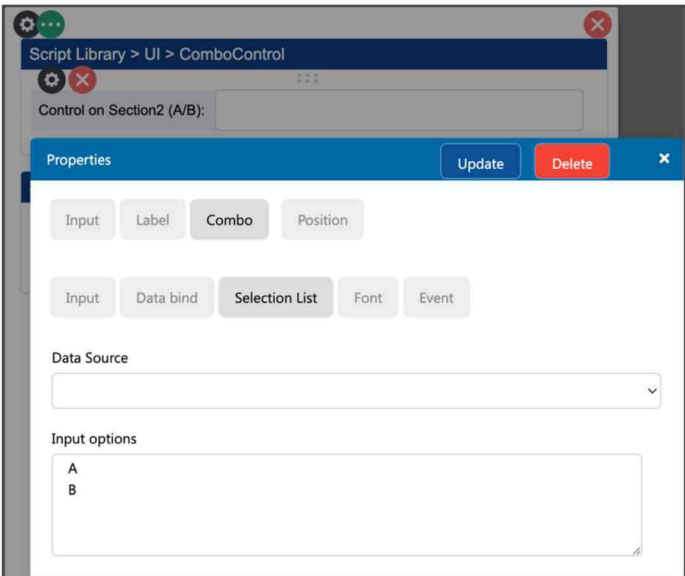


- 2. First, setup an eForm with the following controls:



Object ID	Description
Section1	The first section
Combobox1-1	The pull-down list "Control on Section2 (A/B)"
Section2	The second section
Combobox2-1	The pull-down list "Result Pull-Down List"

- 3. The first pull-down list with two options (A,B).



4. Create a new Script Library on UI.ComboboxControl

5. Click the Params to configure the parameters.

Script Library Management				
	Class Name	Trigger Control	Event	
Add				
Remove	Antelope.UI.ComboboxControl	Combobox1-1	onChange	
Params				

6. Set the following parameters so that when user change the value on Combobox1-1 to “B”, it will disable Section2.

7. Testing: When user change the value on Combobox1-1 to “B”, it will disable Section2.

8. You can control multiple objects if the eForm contains more components.

VII. UI.CheckboxControl

1. The UI.CheckboxControl allows user to enable / disable any objects (even the whole section) when they select a value from the pull-down list.

The diagram illustrates the UI.CheckboxControl interface. It consists of two panels separated by a greater-than sign (>). Both panels have a title bar 'Script Library > UI > CheckboxControl'. The left panel shows the initial state: 'Pull-Down List' and 'Textbox' are both unchecked. The right panel shows the state after a selection: 'Pull-Down List' is checked, and 'Textbox' is unchecked. Both panels have a 'Section2' header with a 'Pull-Down List' dropdown and a 'Section3' header with a 'Textbox' input field.

2. First, setup an eForm with the following controls:

The diagram shows the eForm setup. It has a title bar 'Script Library > UI > CheckboxControl'. Below the title bar are two checkboxes: 'Pull-Down List' and 'Textbox'. Below these are three sections: 'Section2' with a 'Pull-Down List' dropdown, and 'Section3' with a 'Textbox' input field.

Object ID	Description
Section1	The first section
Checkbox1	The checkbox with 2 options
Section2	The second section
Combobox2-1	The pull-down list on Section2
Section3	The third section
Textbox3-1	The textbox on Section3

3. Create a new Script Library on UI.CheckboxControl

Script Library Management - Add

Class: Antelope

Category: UI

Function: CheckboxControl

Enable or Disable controls based on checkbox state

Trigger Control: Checkbox1-0

Event: oncheck

The Checkbox1-0 refers to the first checkbox. Unlike pull-down list, you need to setup a script library for each checkbox.

☐ Pull-Down List

4. Click the Params to configure the parameters.

Script Library Management				
	Class Name	Trigger Control	Event	
Add				
Remove	Antelope.UI.CheckboxControl	Checkbox1-0	oncheck	

5. Set the following parameters so that when user clicks the first checkbox, it will disable the pull-down list. Reverse enable or disable means when user un-click the checkbox, the pull-down list will enable.

Edit Parameter

Checkbox Control: Checkbox1-0

Target Controls: Combobox2-1

Reverse enable or disable: true

Mandatory when enable: true

6. Testing: When clicks the first checkbox, it will disable the pull-down list, or vice-versa.

Script Library > UI > CheckboxControl

☒ Pull-Down List ☐ Textbox

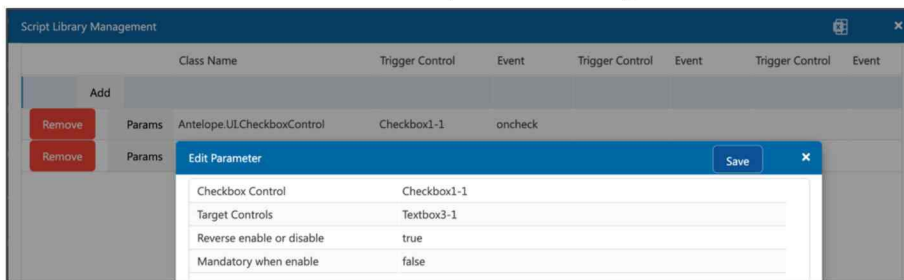
Section2

Pull-Down List: [Dropdown]

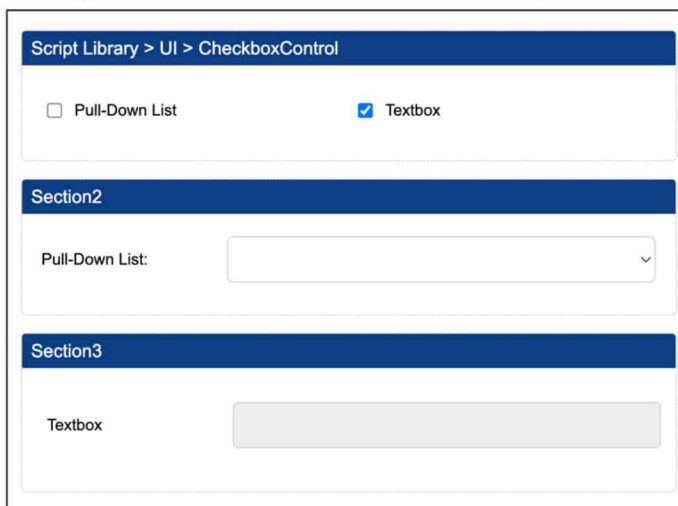
Section3

Textbox: [Text Input]

7. Continue with the second control (Checkbox1-1), in which it disables the Textbox3-1.



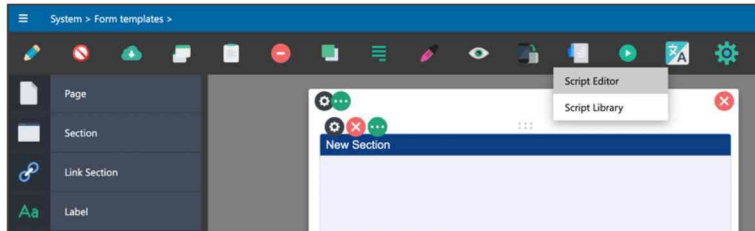
8. Testing: When clicks the second checkbox, it will disable textbox, or vice-versa.



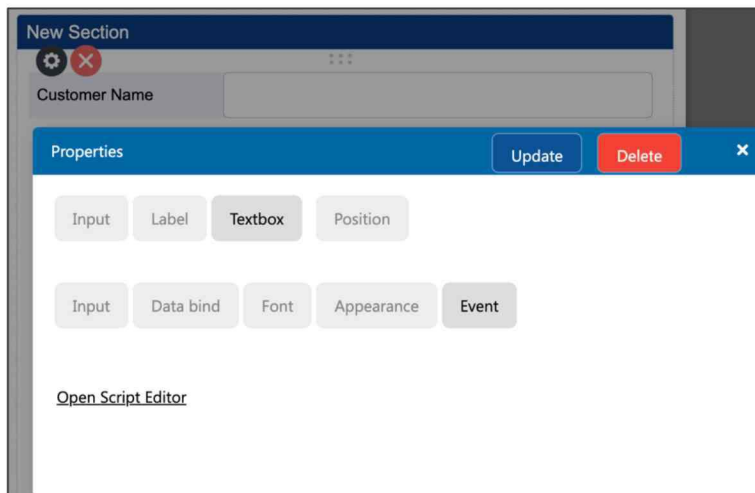
F. Script Editor

I. Load Script Editor

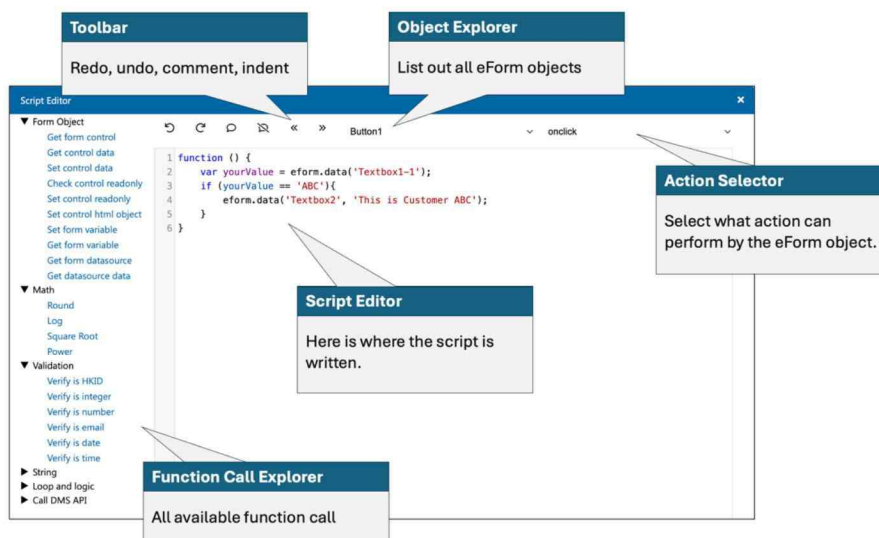
1. There are two methods to load the Script Editor:
 - a. From the horizontal toolbar, select Script Editor.



- b. From the Event tab of each object, click Open Script Editor.



2. The Script Editor allows you to write Javascript to control the user interface of the eForm. The following are descriptions of different components:



G. Object Functions and Events

I. Label

EFLabel.getData()	
Description	Get the label caption.
Example	<pre>var objLabel = eform.get('Label1'); var val = objLabel.getData(); alert(val);</pre>

EFLabel.getVisible()	
Description	Return the visibility of the label as true or false
Example	<pre>var objLabel = eform.get('Label1'); var val = objLabel.getVisible(); alert(val);</pre>

EFLabel.setData(val)		
Description	Set the label caption	
Parameters	Field	Remarks
	val	A string value
Example	<pre>var objLabel = eform.get('Label1'); objLabel.setData('ABC');</pre>	

EFLabel.setVisible (isVisible,isPermenant)		
Description	Set the label as visible or not	
Parameters	Field	Remarks
	isVisible	true/false, for set visible to true or false and disable the input
	isPermenant	Save the visible state or not. If this set to true, the visibility of the control is saved with the eform, if not that visibility control is one off only
Example	<pre>var objLabel = eform.get('Label1'); objLabel.setVisible(true,true);</pre>	

II. Textbox

EFormTextbox.getData()	
Description	Get the textbox content
Example	<pre>var objTextbox = eform.get('Textbox1'); var val = objTextbox.getData(); alert(val);</pre>

EFormTextbox.getDisplay()	
Description	Get the display value which may be different to the data store.
Example	<pre>var objTextbox = eform.get('Textbox1'); var val = objTextbox.getDisplay(); alert(val);</pre>

EFormTextbox.getVisible()	
Description	Return the visibility of the textbox as true or false
Example	<pre>var objTextbox = eform.get('Textbox1'); var val = objTextbox.getVisible(); alert(val);</pre>

EFormTextbox.isReadOnly()	
Description	Return whether the textbox is read-only as true or false
Example	<pre>var objTextbox = eform.get('Textbox1'); var val = objTextbox.isReadOnly(); alert(val);</pre>

EFormTextbox.setData(val)		
Description	Set the textbox content	
Parameters	Field	Remarks
	val	A string value
Example	<pre>var objTextbox = eform.get('Textbox1'); objTextbox.setData('ABC');</pre>	

EFormTextbox.setReadOnly(true/false)		
Description	Set the textbox as read-only or not	
Parameters	Field	Remarks
	isReadOnly	true/false, for set read only to true or false and disable the input
Example	<pre>var objTextbox = eform.get('Textbox1'); objTextbox.setReadOnly(true);</pre>	

EFormTextbox.setVisible (isVisible,isPermenant)		
Description	Set the textbox as visible or not	
Parameters	Field	Remarks
	isVisible	true/false, for set visible to true or false and disable the input
	isPermenant	Save the visible state or not. If this set to true, the visibility of the control is saved with the eform, if not that visibility control is one off only
Example	<pre>var objTextbox = eform.get('Textbox1'); objTextbox.setVisible(true,true);</pre>	

III. Combobox (Pull-Down List)

EFCombobox.getData()	
Description	Get the combobox selected option
Example	<pre>var objCombo = eform.get('Combo1'); var val = objCombo.getData(); alert(val);</pre>
Result	The following example, which is a pull-down list of product name (display value) and product code (key value). The product code is stored as key value which is invisible to end user. We can use the HTML code to describe the relationship between display and key value: <pre><select> <option value="P0001">Product A</option> <option value="P0002">Product B</option> </select></pre> The combobox will return Product A P0001 as the value, the first value is the display value, the second one is key value, which are separated by a double pipe. If you want to get only the display value, you can use getDisplay(), on the other hand, the key value can use getKey(), which will be described in next two tables.

EFCombobox.getDisplay()	
Description	Get the combobox display value of the selected item.
Example	<pre>var objCombo = eform.get('Combo1'); var val = objCombo.getDisplay(); alert(val);</pre>

EFCombobox.getKey()	
Description	Get the combobox key value of the selected item.
Example	<pre>var objCombo = eform.get('Combo1'); var val = objCombo.getKey(); alert(val);</pre>

EFCombobox.getSelected()	
Description	Get the combobox key value of the selected item.
Example	<pre>var objCombo = eform.get('Combo1'); var val = objCombo.getSelected(); alert(val);</pre>

EFCombobox.getVisible()	
Description	Return the visibility of the combobox as true or false
Example	<pre>var objCombo = eform.get('Combo1'); var val = objCombo.getVisible(); alert(val);</pre>

EFCombobox.isReadOnly()	
Description	Return whether the combobox is read-only as true or false
Example	<pre>var objCombo = eform.get('Combo1'); var val = objCombo.isReadOnly(); alert(val);</pre>

EFCombobox.setData(val)		
Description	Set the combobox content	
Parameters	Field	Remarks
	val	A string value
Example	<pre>var objCombo = eform.get('Combo1'); objCombo.setData('Product B');</pre>	

EFCombobox.setReadOnly(true/false)		
Description	Set the combobox as read-only or not	
Parameters	Field	Remarks
	isReadOnly	true/false, for set read only to true or false and disable the input
Example	<pre>var objCombo = eform.get('Combo1'); objCombo.setReadOnly(true);</pre>	

EFCombobox.setVisible (isVisible,isPermenant)		
Description	Set the combobox as visible or not	
Parameters	Field	Remarks
	isVisible	true/false, for set visible to true or false and disable the input
	isPermenant	Save the visible state or not. If this set to true, the visibility of the control is saved with the eform, if not that visibility control is one off only
Example	<pre>var objCombo = eform.get('Combo1'); objCombo.setVisible(true,true);</pre>	

IV. AutoComplete (Pull-Down List)

EFAutoComplete.getDisplay()	
Description	Get the combobox display value of the selected item.
Example	<pre>var objAutoComplete = eform.get('AutoCompletem1'); var val = objAutoComplete.getDisplay(); alert(val);</pre>

EFAutoComplete.getKey()	
Description	Get the combobox key value of the selected item.
Example	<pre>var objAutoComplete = eform.get('AutoCompletem1'); var val = objAutoComplete.getKey(); alert(val);</pre>

EFAutoComplete.getSelected()	
Description	Get the combobox key value of the selected item.
Example	<pre>var objAutoComplete = eform.get('AutoCompletem1'); var val = objAutoComplete.getSelected(); alert(val);</pre>

EFAutoComplete.getVisible()	
Description	Return the visibility of the combobox as true or false
Example	<pre>var objAutoComplete = eform.get('AutoCompletem1'); var val = objAutoComplete.getVisible(); alert(val);</pre>

EFAutoComplete.isReadOnly()	
Description	Return whether the combobox is read-only as true or false
Example	<pre>var objAutoComplete = eform.get('AutoCompletem1'); var val = objAutoComplete.isReadOnly(); alert(val);</pre>

EFAutoComplete.setData(val)		
Description	Set the combobox content	
Parameters	Field	Remarks
	val	A string value
Example	<pre>var objAutoComplete = eform.get('AutoCompletem1'); objAutoComplete.setData('Product B');</pre>	

EFCombobox.setReadOnly(true/false)		
Description	Set the combobox as read-only or not	
Parameters	Field	Remarks
	isReadOnly	true/false, for set read only to true or false and disable the input
Example	<pre>var objAutoComplete = eform.get('AutoCompleter1'); objAutoComplete.setReadOnly(true);</pre>	

EFCombobox.setVisible(isVisible,isPermenant)		
Description	Set the combobox as visible or not	
Parameters	Field	Remarks
	isVisible	true/false, for set visible to true or false and disable the input
	isPermenant	Save the visible state or not. If this set to true, the visibility of the control is saved with the eform, if not that visibility control is one off only
Example	<pre>var objAutoComplete = eform.get('AutoCompleter1'); objAutoComplete.setVisible(true,true);</pre>	

V. Signature

EFSignature.getVisible()	
Description	Return the visibility of the signature as true or false
Example	<pre>var objSignature = eform.get('Signature1'); var val = objSignature.getVisible(); alert(val);</pre>

EFSignature.isReadOnly()	
Description	Return whether the signature is read-only as true or false
Example	<pre>var objSignature = eform.get('Signature1'); var val = objSignature.isReadOnly(); alert(val);</pre>

EFSignature.setReadOnly(true/false)		
Description	Set the signature as read-only or not	
Parameters	Field	Remarks
	isReadOnly	true/false, for set read only to true or false and disable the input
Example	<pre>var objSignature = eform.get('Signature1'); objSignature.setReadOnly(true);</pre>	

EFSignature.setVisible(isVisible,isPermenant)		
Description	Set the signature as visible or not	
Parameters	Field	Remarks
	isVisible	true/false, for set visible to true or false and disable the input
	isPermenant	Save the visible state or not. If this set to true, the visibility of the control is saved with the eform, if not that visibility control is one off only
Example	<pre>var objSignature = eform.get('Signature1'); objSignature.setVisible(true,true);</pre>	

VI. Image

EFormImage.getVisible()	
Description	Return the visibility of the image as true or false
Example	<pre>var objImage = eform.get('Image1'); var val = objImage.getVisible(); alert(val);</pre>

EFormImage.isReadOnly()	
Description	Return whether the image is read-only as true or false
Example	<pre>var objImage = eform.get('Image1'); var val = objImage.isReadOnly(); alert(val);</pre>

EFormImage.setReadOnly(true/false)		
Description	Set the image as read-only or not	
Parameters	Field	Remarks
	isReadOnly	true/false, for set read only to true or false and disable the input
Example	<pre>var objImage = eform.get('Image1'); objImage.setReadOnly(true);</pre>	

EFormImage.setVisible (isVisible,isPermanent)		
Description	Set the image as visible or not	
Parameters	Field	Remarks
	isVisible	true/false, for set visible to true or false and disable the input
	isPermanent	Save the visible state or not. If this set to true, the visibility of the control is saved with the eform, if not that visibility control is one off only
Example	<pre>var objImage = eform.get('Image1'); objImage.setVisible(true,true);</pre>	

VII. Data Grid

EFGGrid.getCellStatus(row,col)		
Description	Get a specific cell current status	
Parameters	Field	Remarks
	row	The row id
	col	The column index, start from 0
Example	<pre>var objGrid = eform.get('Grid1'); var val = objGrid.getCellStatus(0,0); alert(val);</pre>	
Result	N = Normal input R = Read-only M = Mandatory input	

EFGGrid.getData(row,col)		
Description	Get a specific cell data	
Parameters	Field	Remarks
	row	The row id
	col	The column index, start from 0
Example	<pre>var objGrid = eform.get('Grid1'); var val = objGrid.getData(0,0); alert(val);</pre>	

EFGGrid.getColCount	
Description	Get number of columns
Example	<pre>var objGrid = eform.get('Grid1'); var val = objGrid.getColCount(); alert(val);</pre>

EFGGrid.getRowCount	
Description	Get number of rows
Example	<pre>var objGrid = eform.get('Grid1'); var val = objGrid.getRowCount(); alert(val);</pre>

EFGGrid.getVisible()	
Description	Return the visibility of the data grid as true or false
Example	<pre>var objGrid = eform.get('Grid1'); var val = objGrid.getVisible(); alert(val);</pre>

EForm.isReadOnly()	
Description	Return whether the data is read-only as true or false
Example	<pre>var objGrid = eform.get('Grid1'); var val = objGrid.isReadOnly(); alert(val);</pre>

EForm.setCellStatus(row,col,status)		
Description	Set a specific cell current status	
Parameters	Field	Remarks
	row	The row id
	col	The column index, start from 0
	status	N = Normal input R = Read-only M = Mandatory input
Example	<pre>var objGrid = eform.get('Grid1'); var val = objGrid.setCellStatus(0,0,'R'); alert(val);</pre>	

EForm.setData(row,col,val)		
Description	Set a specific cell data	
Parameters	Field	Remarks
	row	The row id
	col	The column index, start from 0
	Val	The value
Example	<pre>var objGrid = eform.get('Grid1'); var val = objGrid.setData(0,0,'Customer A'); alert(val);</pre>	

EForm.setReadOnly(true/false)		
Description	Set the data grid as read-only or not	
Parameters	Field	Remarks
	isReadOnly	true/false, for set read only to true or false and disable the input
Example	<pre>var objGrid = eform.get('Grid1'); objGrid.setReadOnly(true);</pre>	

EFGGrid.setVisible (isVisible,isPermenant)		
Description	Set the data grid as visible or not	
Parameters	Field	Remarks
	isVisible	true/false, for set visible to true or false and disable the input
	isPermenant	Save the visible state or not. If this set to true, the visibility of the control is saved with the eform, if not that visibility control is one off only
Example	<pre>var objGrid = eform.get('Grid1'); objGrid.setVisible(true,true);</pre>	

EFGGrid.appendRow(row,value_array)		
Description	Append a row to data grid and start writing value to each cell(s)	
Parameters	Field	Remarks
	row	The row id 0 = Append on the first row 1 = Append on the second row null = Add a new row at the bottom
	value_array	['Value 1','Value 2',...]
Example	<pre>var objGrid = eform.get('Grid1'); objGrid.appendRow(0,['Value 1','Value 2']);</pre>	

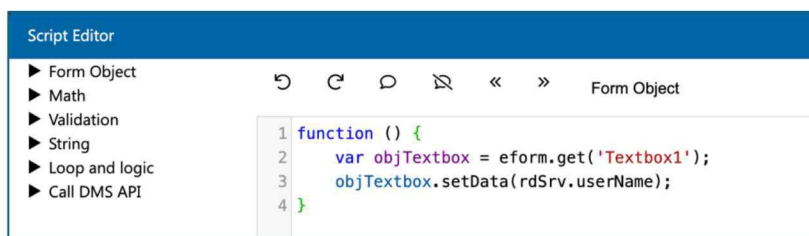
Change Data Grid Row Color	
Description	Change the data grid row color
Example	<pre>var objGrid = eform.get('Grid1'); objGrid.htmlObj.lastChild.childNodes[0].rows[1]. style.background = '#CCCCCC';</pre>

H. Global Variable

I. All Variables

Variable	Return
eform	The current eform object in EFTemplate type
rdSrv.docName	The current eform object name
rdSrv.userId	Current user login id
rdSrv.userName	Current user login name
rdSrv.userDesc	Current user description
rdSrv.groupName	Current user first group name
rdSrv.groupId	Current user first group id
rdSrv.allGroup	Current user all user group
rdSrv.allFuncLevel	Current user all functional level
rdSrv.ouName	Current user company name
rdSrv.phone	Current user phone number (sync from AD)
rdSrv.jobTitle	Current user job title (sync from AD)
rdSrv.props	To access current workflow index field e.g. rdSrv.props.getItem('Invoice No') this get the current workflow process invoice no index value
rdSrv.currentTask	The current workflow task label
rdSrv.currentWitrId	The current task recipient id
rdSrv.currentWitId	The current task id
rdSrv.currentWiId	The current workflow instance id
parent.currentAction	The current selected response in task
rdSrv.today	Today value in yyyy-MM-dd string format
rdSrv.user	Current user object

The following example write the current login username to Textbox1 when the eForm is load.



```

Script Editor
Form Object
Math
Validation
String
Loop and logic
Call DMS API
1 function () {
2   var objTextbox = eform.get('Textbox1');
3   objTextbox.setData(rdSrv.userName);
4 }
  
```

More Examples?

<https://antelope-global.zendesk.com/hc/en-gb/sections/4405130442905-eForm-Script-Samples>

I. Language Translation

1. Antelope 6 eForm Designer supports English, Traditional Chinese, Simplified Chinese and Portuguese language translations.

2. Click the translation icon from the horizontal toolbar.

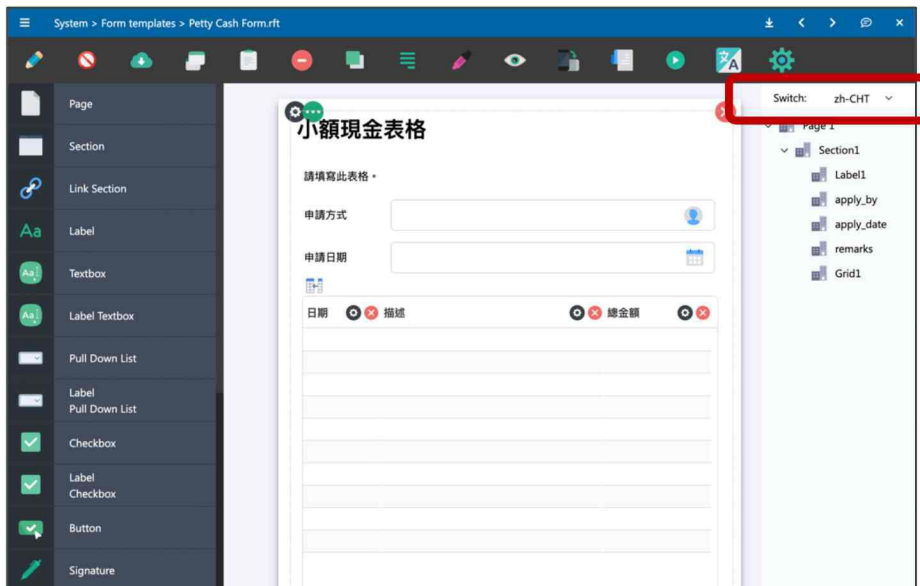


3. Translate into different languages.

Resource Management

	ID	English	zh-CHS	zh-CHT	pt-PT
Save	Section1	Petty Cash Form	小額現金表格	小額現金表格	
Save	Label1	Please fill out this form	请填写此表格	請填寫此表格	
Save	apply_by-0	Apply By	申請方式	申請方式	
Save	apply_by-1				
Save	apply_date-0	Apply Date	申請日期	申請日期	
Save	apply_date-1				
Save	remarks-0	Remarks	注解	註解	
Save	remarks-1				
Save	Grid1.item_date	Date	日期	日期	
Save	Grid1.item_desc	Item Description	描述	描述	
Save	Grid1.item_amount	Amount	总金额	總金額	

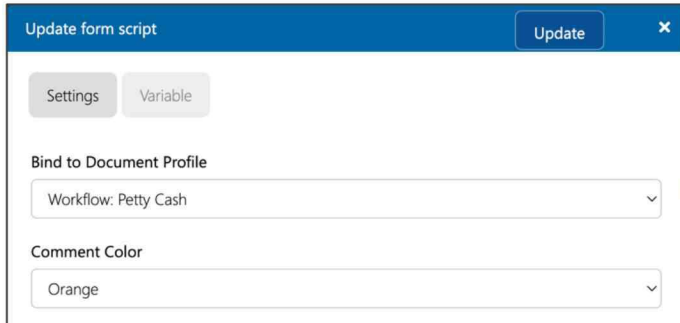
4. You can look at the eForm translation by selecting the language from the Object Explorer.



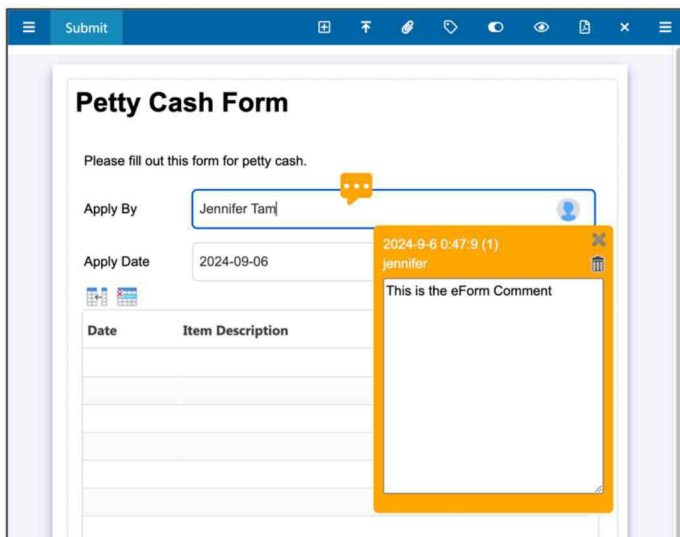
J. eForm Advanced Setting

I. Comment Color

1. The comment color is the color of the comment box where you placed on eForm.



2. You can put an annotation comment when submitting a workflow.



II. Generate Input Border for PDF

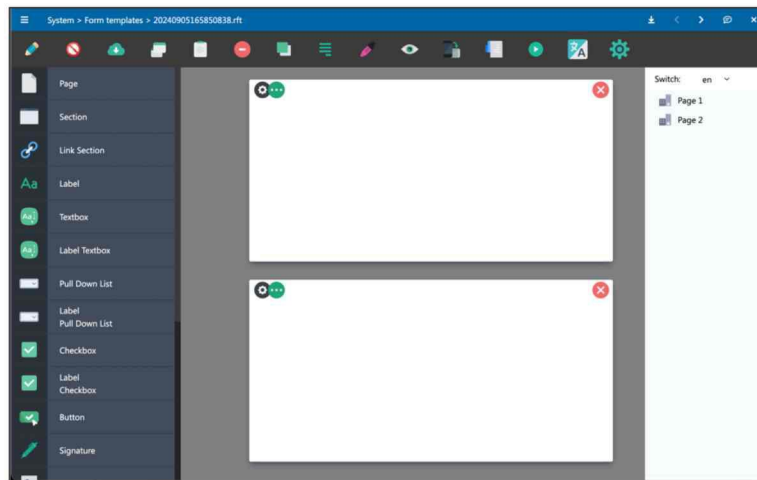
When you export an eForm into PDF file, all input objects such as textbox, pull-down list will have a border.

III. Generate Signature Border for PDF

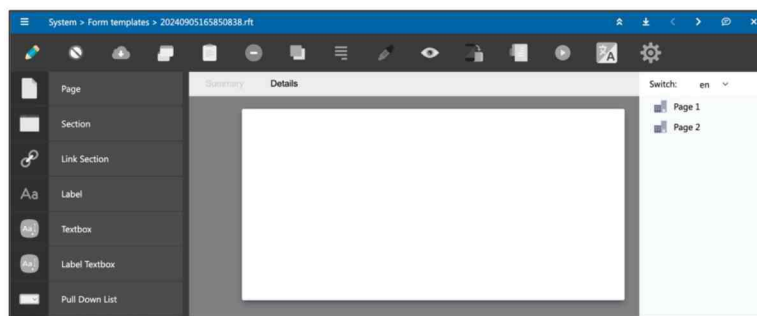
When you export an eForm into PDF file, the Signature object will have a border.

IV. Page Rendering Mode

1. For multiple page eForm, there are two page-rendering mode:
 - a. Page Mode



- b. Tab Mode

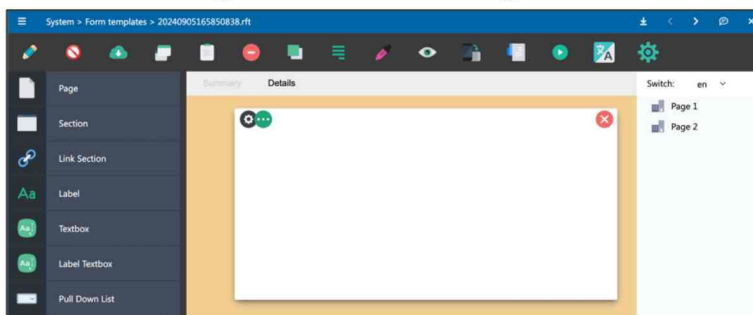


V. Canvas Background Color

1. To change the canvas background color, select the color from the color picker.



2. The canvas background color will change.



VI. Variable

1. You can set a global value which will be used in the entire eForm.

The 'Update form script' dialog box has a blue header with 'Update form script' and an 'Update' button. Below the header are two tabs: 'Settings' and 'Variable'. The 'Variable' tab is active. Under 'Add Variable', there is a text input with 'Testing', another with 'My Value', and an 'Add' button. Below this, the 'Form Variables' section shows a list with 'TotalPages=2' and 'Testing=My Value', and a red 'Del' button.

2. In the Script Editor, you can call the global value at any time.

The 'Script Editor' window has a blue header. On the left is a tree view for 'Form Object' with items like 'Get form control', 'Get control data', 'Set control data', 'Check control readonly', 'Set control readonly', 'Set control html object', 'Set form variable', 'Get form variable', 'Get form datasource', and 'Get datasource data'. The main area shows a code editor with the following JavaScript code:

```
1 function () {  
2   var val = eform.Var('Testing');  
3   alert(val);  
4 }
```

3. In reverse, you can write to the global value.

The 'Script Editor' window is the same as the previous one, but the code in the editor is:

```
1 function () {  
2   eform.Var('Testing', '123');  
3 }
```

4. The change will reflect in Form Variables.

The 'Update form script' dialog box is the same as the first one, but the 'Form Variables' section now shows 'TotalPages=2' and 'Testing=123', reflecting the change made in the script.

K. Validation Test

I. Validation Icon

1. Drag and drop the label object onto the first Section.

L. Advanced Setting of Data Grid

I. Sum Total

1. Drag and drop the label object onto the first Section.

II. Total Data Binding

1. Drag and drop the label object onto the first Section.

III. Hidden

1. Drag and drop the label object onto the first Section.

IV. Rowspan

1. Drag and drop the label object onto the first Section.

V. Combo Related Column Index

1. Drag and drop the label object onto the first Section.

VI. Formular

1. Drag and drop the label object onto the first Section.

VII. Numeric Mask

1. Drag and drop the label object onto the first Section.

Antelope International Limited
Flat G, 16/F, Seabright Plaza, 9-23 Shell Street
North Point, Hong Kong



Visit us on at
<https://www.antelopezglobal.com>