Adobe LiveCycle

MISA AODA Web Accessibility Toolkit

MISA

Ontario

TABLE OF CONTENTS

Creating Accessible Forms in Adobe LiveCycle

This document details steps needed to add accessibility features to a form created using Adobe LiveCycle; it also outlines some “best practices”. This document walks through creating a form and is provided for example purposes only. Use the same steps to create your own document.

Best Practices for Form Creation

A form is not accessible if it is not easy to use. The first step is to try to design forms that are uncluttered and logically arranged with clear and simple instructions to help all users fill them in.

General Best Practices:

- Keep forms simple and easy to use
- Configure form properties to generate accessibility information (tags)
- Provide text equivalents for images
- Ensure form controls are keyboard accessible
- Provide heading cells for tables
- Provide navigable form structure
- Identify natural language and any changes in language

Provide proper labels for form controls

- All form controls need labels to answer a user’s question: What is this for?
- Captions provided with controls are sufficient, providing the wording is appropriate

Ensure reading and tab order are correct

Tab order is highly important for keyboard users and for users of accessibility technologies.

Use colour responsibly (colour contrast is important)

- Many users with vision impairment rely on high contrast between text and the background to read forms; when the contrast between background and foreground colours is not sufficient a form can be difficult or impossible for some users
- Do not convey information, such as indicating required fields, by colour alone

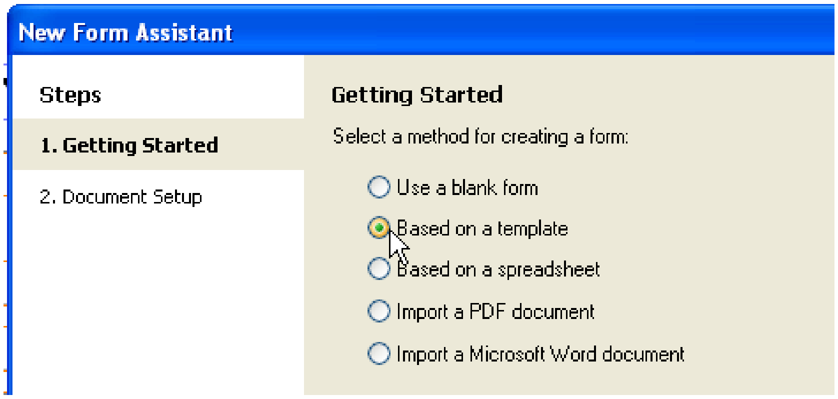
Accessible PDF forms include a logical structure plus additional information about the form’s contents that increases accessibility. For a form to be accessible, it must be perceivable by assistive technology. For example, most screen readers will not consider the visual layout of your form, but rather the underlying structure.

Form creation from a template

Forms in LiveCycle Designer can be designed in another application, such as Microsoft Word or InDesign, and then imported to LiveCycle. Forms can also be designed exclusively in LiveCycle Designer.

Forms can be created from scratch in LiveCycle, or from one of the many templates available within the program. This document uses a template from within Adobe LiveCycle to walk through accessibility features.

To follow along using this template, choose open, Select **File** > **New** and then select **Based on a Template** (as shown below). Click **Next**.



New Form Assistant

Steps

1. Getting Started

2. Document Setup

Getting Started

Select a method for creating a form:

○ Use a blank form

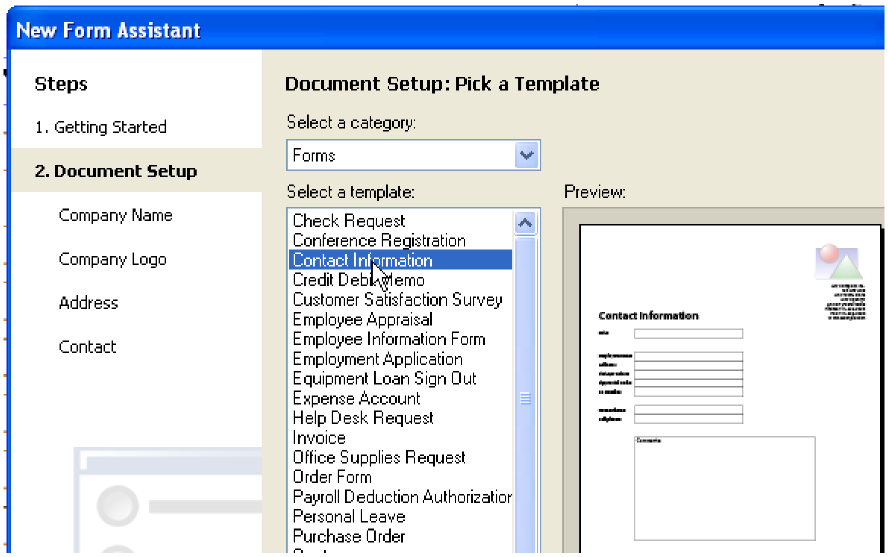
● Based on a template

○ Based on a spreadsheet

○ Import a PDF document

○ Import a Microsoft Word document

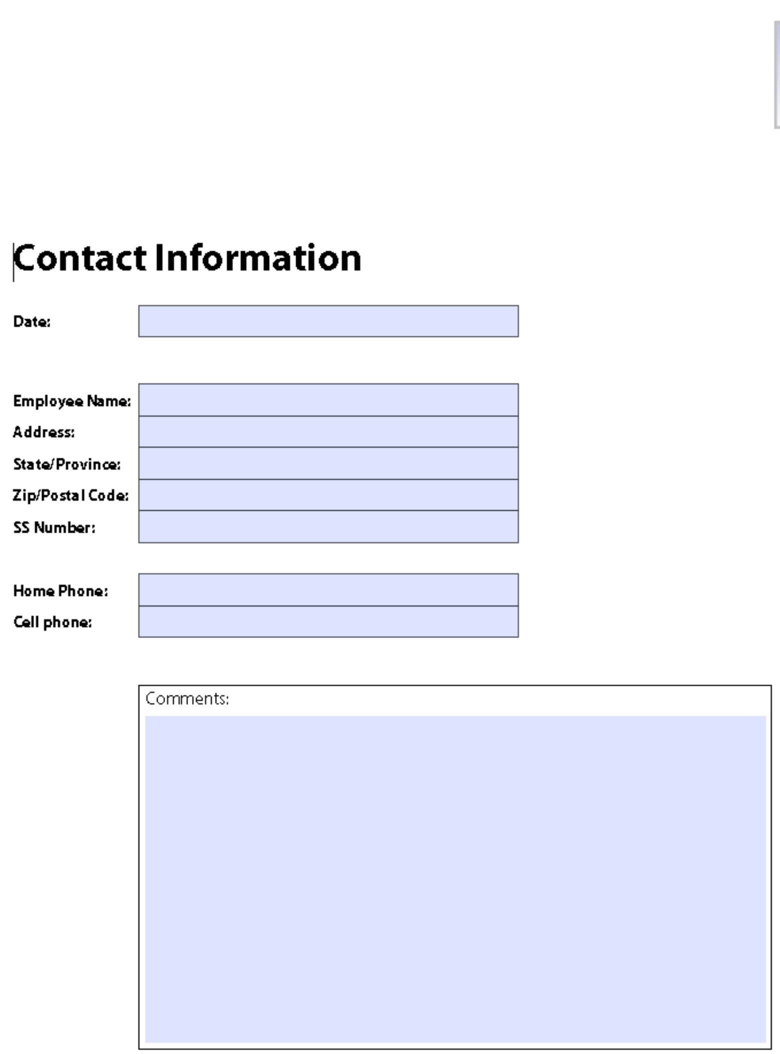
With the category Forms selected, in the Document Setup window select **Contact Information** from the list of available templates.



Select a category: Forms

Select a template: Contact Information

Below is the resulting template (in Preview Mode) which will be used for the purpose of demonstration.



MISA WCAG groups
123 Any Ave
Any Town, State
Any Country
Any ZIP/Postal Code
Phone: 111-222-3333
Fax: 111-222-4444
www.example.com

Contact Information

Date:

Employee Name:
Address:
State/Province:
Zip/Postal Code:
SS Number:

Home Phone:
Cell phone:

Comments:

Click the Next button to move through the panes until you reach the Form Return Setup window.

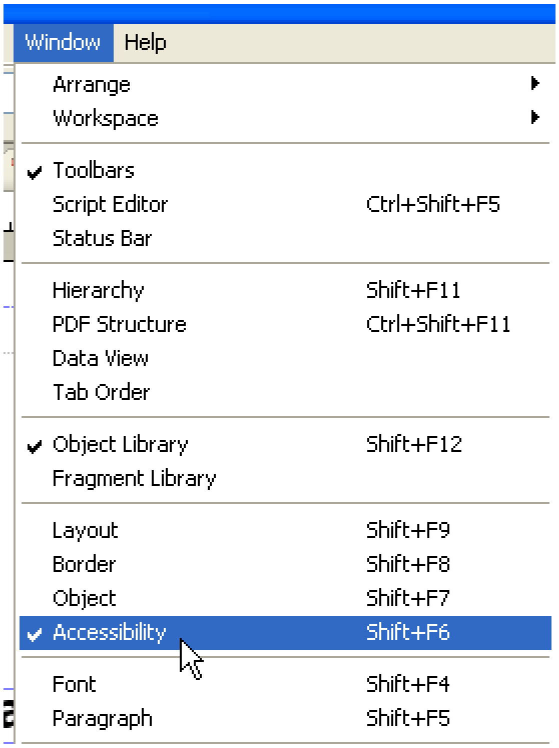
For the purposes of this document, uncheck both the Add and email button and the Add a print button check boxes, then click the Finish button.

Setting Up for Accessibility

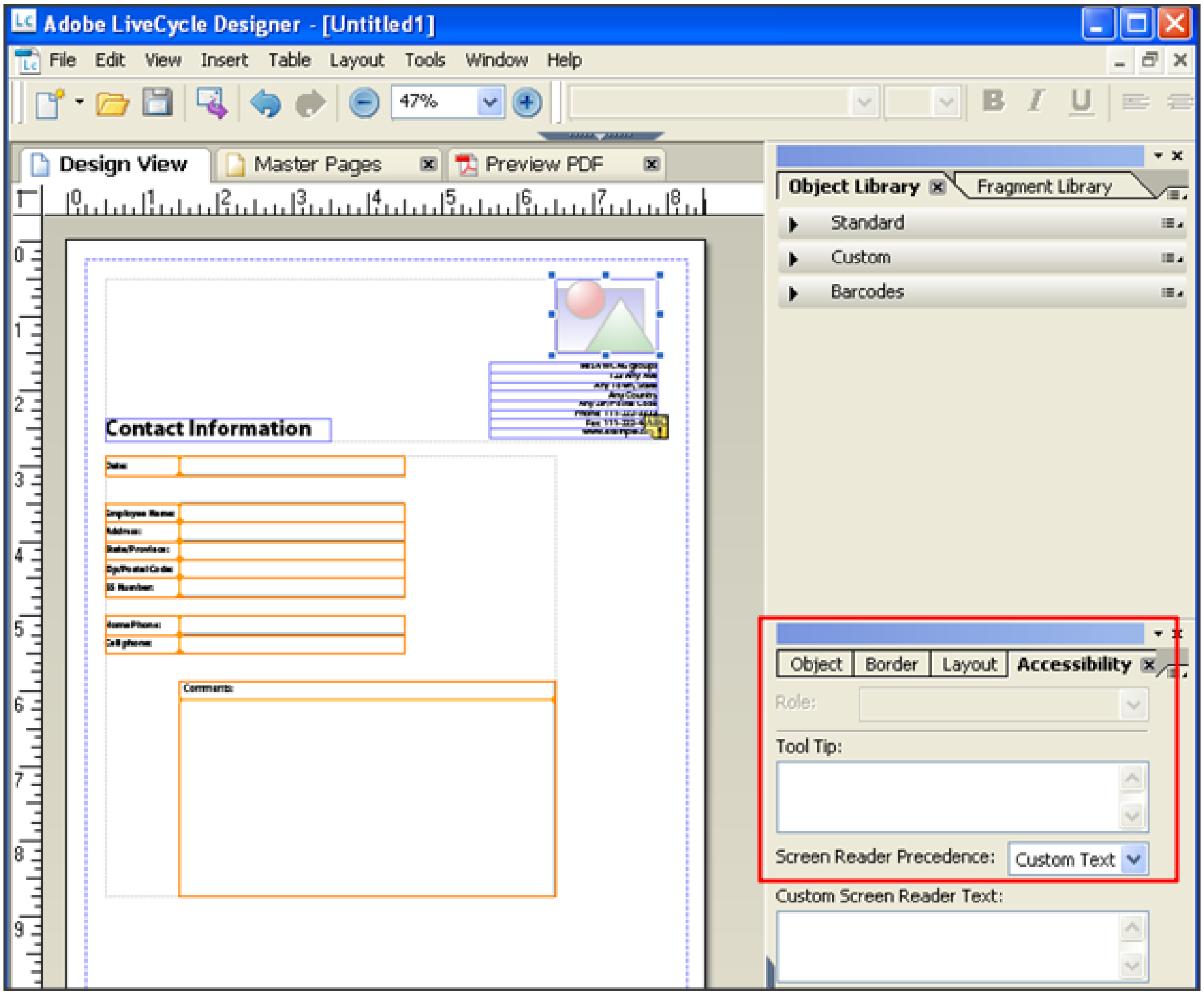
One of the biggest hurdles to using LiveCycle Designer is to know where to find your tools. You will use the tools on the Toolbars to select and format objects on your form. If you do not see the Toolbars window above your form, you will need to open it.

In Design View, open the Toolbars (Window > Toolbars).

Make the Accessibility Palette available (Window > Accessibility).



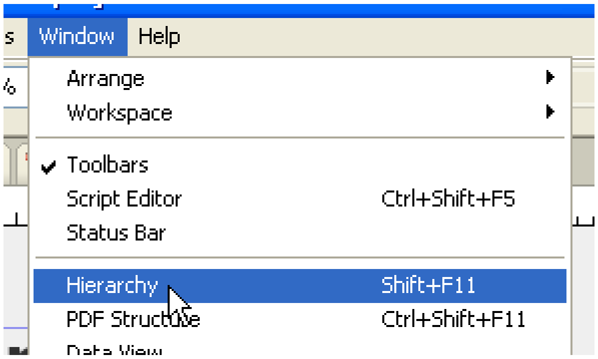
The Accessibility Palette should now appear on the right hand side of the form.



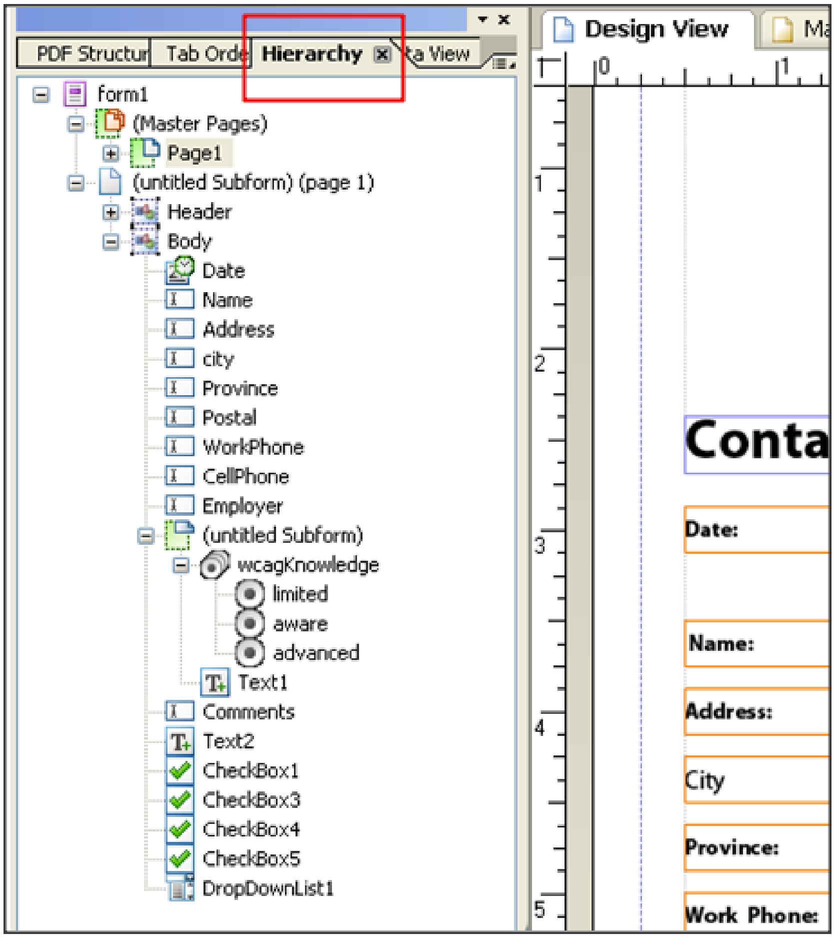
Screen Reader Precedence: Custom Text

It is also handy to have the hierarchy palette open at all times. This provides a view of how your form will be put together as each element is added (or deleted).

To open the Hierarchy palette, choose Window > Hierarchy.



In addition to the Hierarchy palette, the PDF Structure, Data View, and Tab Order palettes all reside in this area of the LiveCycle Interface. This palette area will now be anchored to the left of your form.

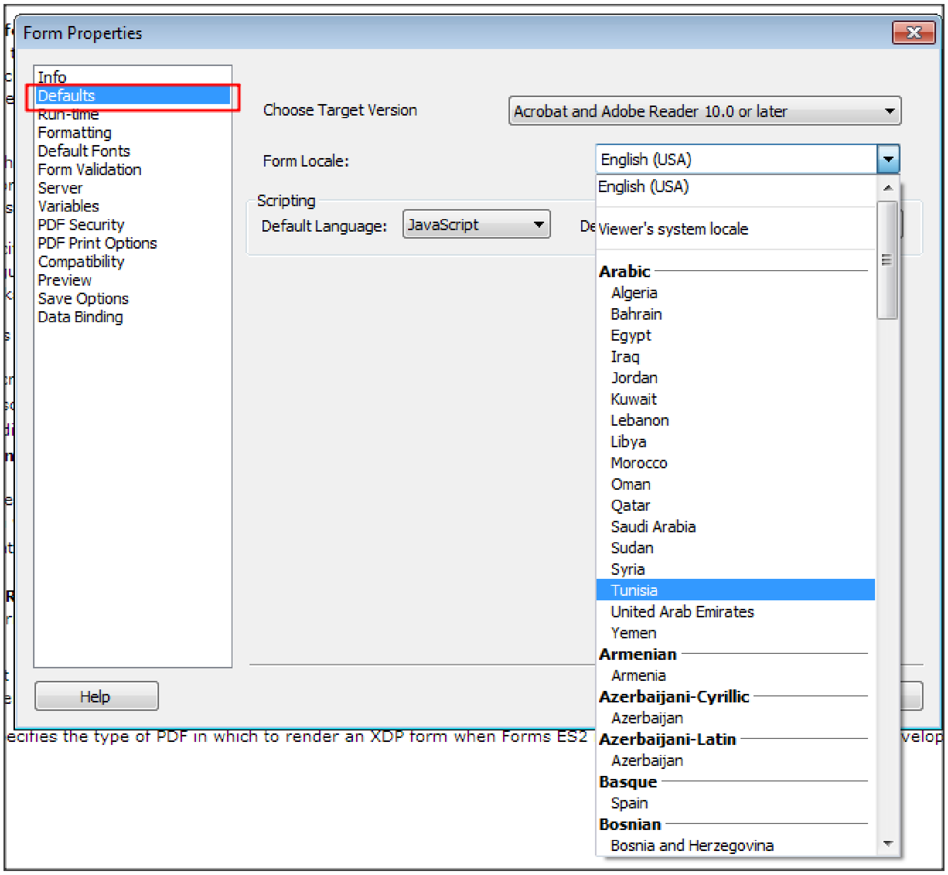


Set Document Language

Ensure that the language for the document is set so that assistive technologies can read the form in its intended language.

Success Criterion:
3.1.1 Language of a Page – Level A

Choose File > Form Properties and select the Default tab. Select the appropriate language for the Form Locale (see the illustration at right). Click OK.



Choose Target Version: Acrobat and Adobe Reader 10.0 or later

Form Locale: English (USA)

Scripting
Default Language: JavaScript

ecifies the type of PDF in which to render an XDP form when Forms ES2 ... velop

In LiveCycle Designer, setting the language of a form is done by setting the Locale property of the form.

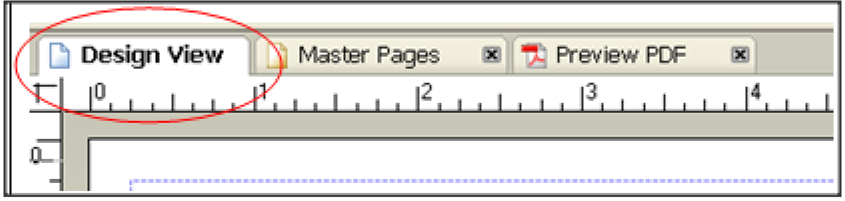
Alternative Text

Ensure that all images have alternative text (or equivalent textual descriptions).

Success Criterion:
1.1.1 Non-text Content – Level A

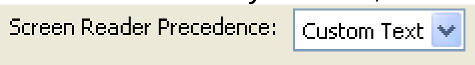
To add alternative text:

1. Ensure you are using Design View



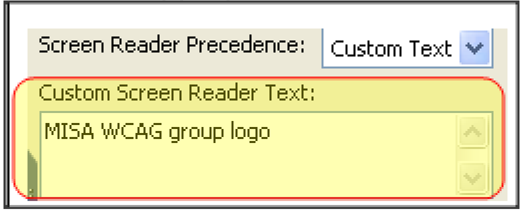
2. Click on the image

3. In the Accessibility Palette, under Screen Reader Preference, select Custom Text



Screen Reader Precedence: Custom Text

4. Enter the appropriate alternative text in the input field



Screen Reader Precedence: Custom Text

Custom Screen Reader Text:

MISA WCAG group logo

5. Test to see that you've added the alternative text by hovering over the image. The alt text will also be read out loud by screen readers.

Headings and Roles

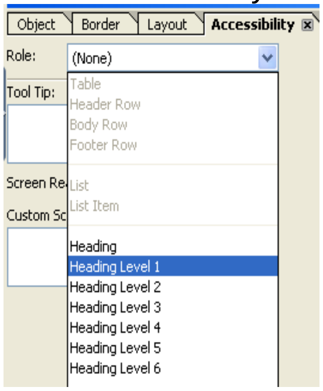
For your document to be accessible it is important to provide proper structure behind its creation, so that it can reflow properly in any device. Creating a document with headings that identify logical segments to the form is the easiest and most effective way to provide navigable structure. To do this, assign a heading “role” to your text, through the Accessibility palette.

ALL documents should have (at minimum) a Heading Level 1. If there are further sub-sections (or segments) then they should be assigned the appropriate heading level in sequential order.

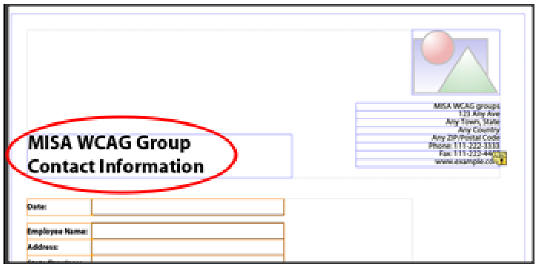
Success Criterion:
1.3.1 Info and Relationships – Level A and 1.3.2 Meaningful Sequence – Level A

To assign a heading role:

1. Select the text you are identifying as a heading
2. In the Accessibility Palette, select the proper heading level under Role



The sample form we have been working through has just one heading:



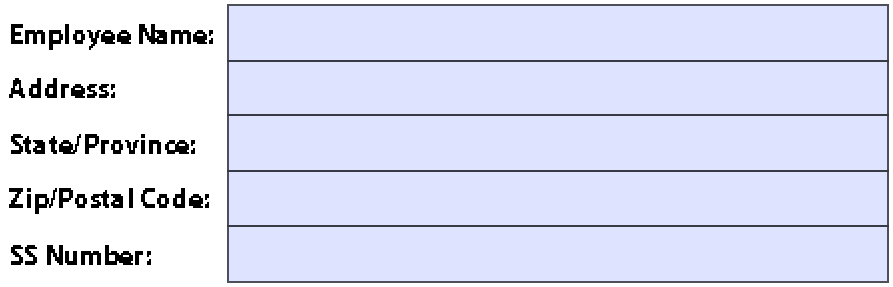
Spacing

Ensure that there is proper spacing (leading) between lines of text AND input fields. The CNIB recommends that line spacing be set somewhere in the range between 1.25 and 2.0

The default template we have started with can be altered by simply dragging elements around on the form.

Example of Poor Spacing

In the screenshot below, there is zero white space (leading) between the input fields which could present difficulties for certain individuals.



Employee Name:

Address:

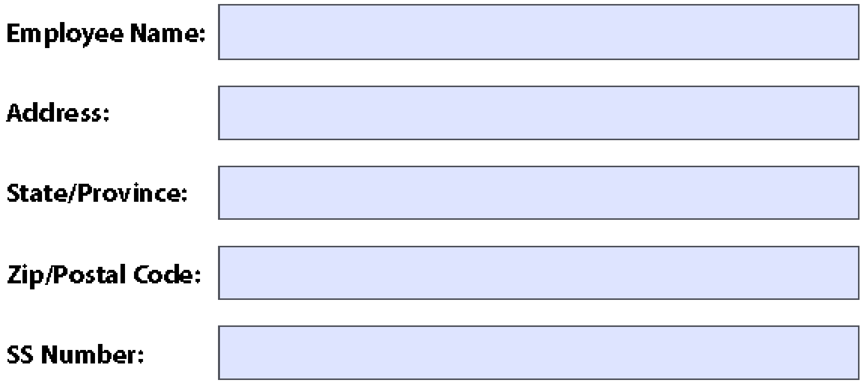
State/Province:

Zip/Postal Code:

SS Number:

Example of Good Spacing

Below is an example of good spacing.



Employee Name:

Address:

State/Province:

Zip/Postal Code:

SS Number:

Labels on input fields

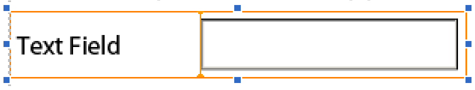
All form inputs have associated text labels OR, if labels cannot be used, a descriptive title attribute. For sighted and low vision users it is important that the label is properly positioned adjacent to the control; however it is not enough to just place a text object next to the control.

For **most** input fields, the **Caption** option will be used (as the label) but LiveCycle provides the ability to add Custom Screen Reader Text or Tool Tip that can substitute for the “title” attribute.

Success Criterion:
1.1.1 Non-text Content – Level A
1.3.1 Info and Relationships – Level A

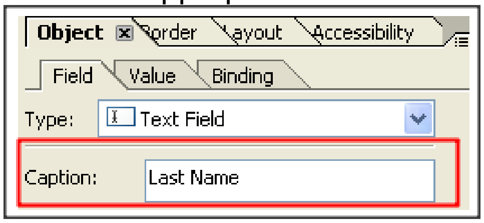
To create an input field with a caption

1. Drag/Drop an input text field Text Field from the Object Library to the spot in the form you want it to appear
2. A default input field will appear



Text Field

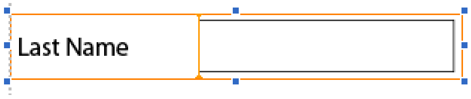
3. Enter the appropriate text for the label (caption) in the Object Palette



Type: Text Field

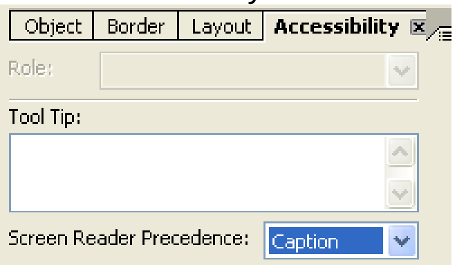
Caption: Last Name

4. The input field will be updated on the page



Last Name

5. In the Accessibility Palette select Screen Reader Precedence: Caption.



Role:

Tool Tip:

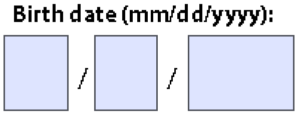
Screen Reader Precedence: Caption

By **default**, LiveCycle presents text to a screen reader in the order of Custom Text, Tool Tip, Caption and Name.

When to use Custom Text

Generally speaking, a caption should always be used; in rare cases where it can’t be used, or the caption doesn’t accurately describe the purpose of the input field, then use Custom Text (which will substitute for the “title” attribute).

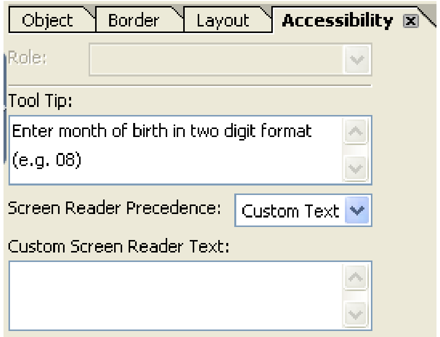
In the example below, a visible caption for each individual input field is not viable.



Birth date (mm/dd/yyyy):

/ /

In the first field (the month field), type instructions in the **Tool Tip**: field



Role:

Tool Tip: Enter month of birth in two digit format (e.g. 08)

Screen Reader Precedence: Custom Text

Custom Screen Reader Text:

Instructions should include the data to be entered and should also provide an example of the desired format.

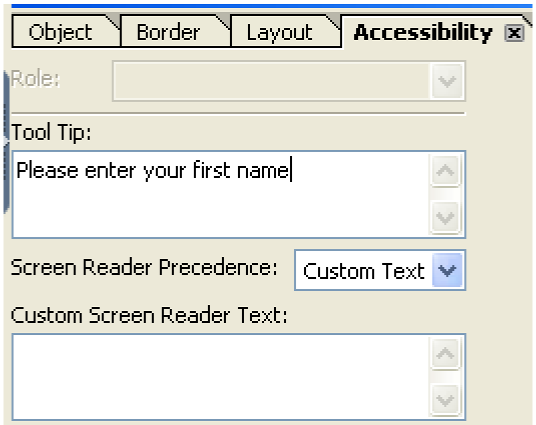
Repeat steps for the remaining two fields, making note that the year field is a four digit entry.

Note: For most input fields there are Tool Tip, Screen-Reader Precedence and Custom Screen-Reader-Text options that can be used in various combinations, all of which have pros and cons in terms of what gets recognized by a Screen Reader versus what gets recognized on hover (visually).

Screen readers will check for accessibility content in the order of Custom Text, then Tool Tip, then Caption, Form Name, None. If it finds nothing (for example under Custom Text) it moves to the next option. As soon as information is found, it stops looking.

Overall Recommendation: use caption whenever possible. When the caption cannot be used, be sure to:

- Enter instructions in the **Tool Tip** Option,
- Leave the **Screen Reader Precedence** at the default **Custom Text** setting
- Leave the **Custom Screen Reader Text** input field blank (see example in image below)



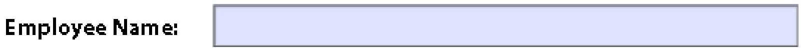
Tool Tip: Please enter your first name

Screen Reader Precedence: Custom Text

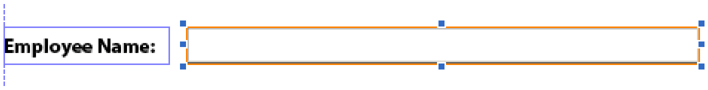
Custom Screen Reader Text:

When is a label not a label?

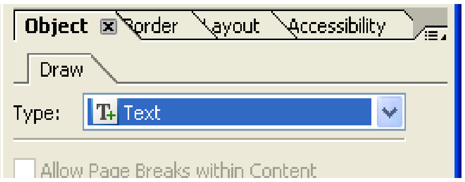
If you were to look at the field below in Preview mode, it is easy to tell that Employee Name and the field as associated with one another.


Employee Name:

However, if in Design View, the label appears as two separate elements, the label is not associated with the field. This would make it difficult for a screen reader user to associate the label with the appropriate field.

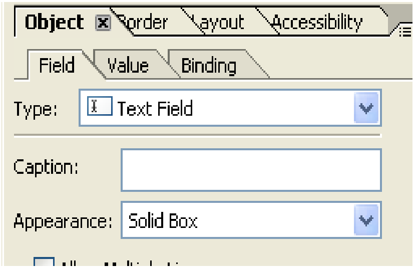

Employee Name:

In the example above, the Object Text Field is selected and no value has been provided in the “Caption” for the Text Input Field



Type: Text

□ Allow Page Breaks within Content



Type: Text Field

Caption:

Appearance: Solid Box

While this method creates a proper input field from a strictly visual standpoint, it fails S.C. 1.1.1 as there is no association between the visual label (Employee Name) and the actual input field.

Success Criterion:
1.1.1 Non-text Content – Level A

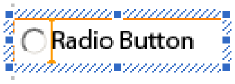
Radio Buttons

A radio button lets you select only one item from a group of items.

To make radio buttons accessible in LiveCycle, we must ensure they are “grouped” and that the purpose of the grouping is identified (for Screen Reader users) and finally, that each radio button is properly labelled.

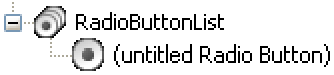
To group radio buttons:

1. Drag/Drop a **radio button** ◉Radio Button from the Object Library to the form Library to the spot in the form you want it to appear

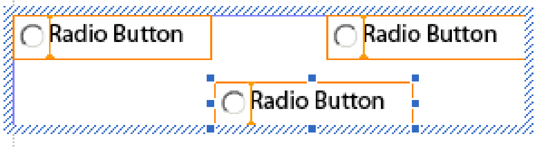


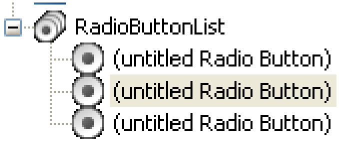
The blue outline indicates a new exclusion grouping (even though there is currently only one item).

Note: In the ***Hierarchy palette*** we also see that a new exclusion grouping (Radio Button List) has been created that contains one untitled Radio Button



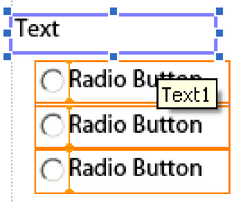
2. Drag/Drop Additional radio buttons to add to your grouping. They will automatically be included in the group, regardless of where they are placed on the page.



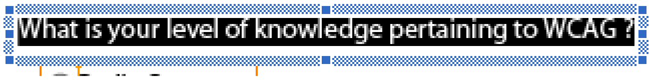


3. Rearrange the grouping to your desired layout

4. Drag/Drop An instructional text block T̲+Text from the Object palette to a place in your form immediately preceding your radio button List.

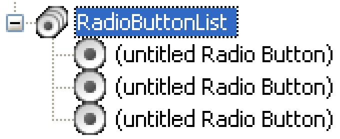


5. Double Click on the text field and **type** the question you would like to ask

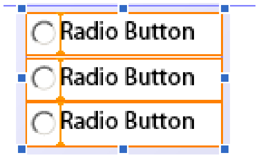


What is your level of knowledge pertaining to WCAG ?

6. Select the exclusion group in the hierarchy palette



RadioButtonList
(untitled Radio Button)
(untitled Radio Button)
(untitled Radio Button)



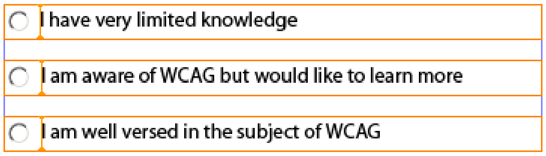
○ Radio Button
○ Radio Button
○ Radio Button

7. Select the accessibility palette

8. Enter Text (in the **Tool Tip** input field) that you would like read out by the screen reader for the radio button group.

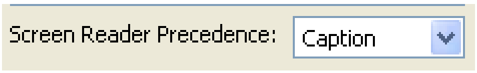
   Note: The text should not be a repeat of the actual question, but more of an action summary. e.g. in our example, the Screen Reader text should be: "***Select your WCAG knowledge***"

9. Type appropriate **label (Caption)** text for each individual Radio Button that gives a unique option for selection



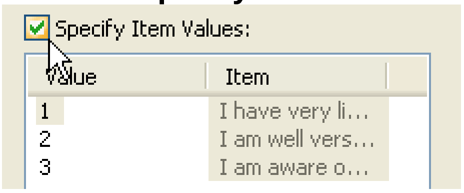
○ I have very limited knowledge
○ I am aware of WCAG but would like to learn more
○ I am well versed in the subject of WCAG

10. Select **Caption** for Screen Reader Precedence in the Accessibility palette for each Radio Button



Screen Reader Precedence: Caption

11. Uncheck **Specify Item Values** from the Object | Binding tab



☑ Specify Item Values:

Value	Item
1	I have very li...
2	I am well vers...
3	I am aware o...

Check Boxes

A check box allows you to select multiple items from a group of items.

To create accessible check boxes:

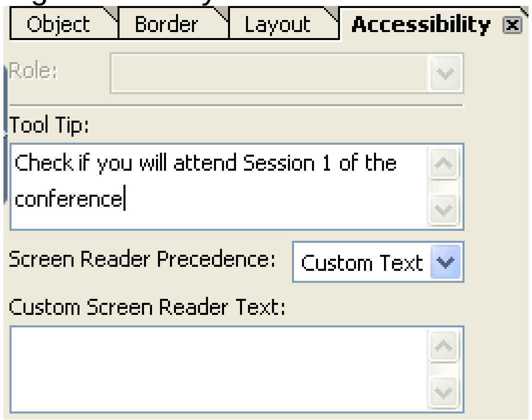
1. Drag/Drop an instructional **text block** Text from the Object palette to a place in your form immediately preceding your check box options and enter the qualifying text for the options you want to present.

2. Drag/Drop Check Box elements ☑ Check Box from the Object palette to the desired location on your form (position as desired)

3. Select “Checkbox”

4. Enter A **Tool Tip** for the Check Box within the Accessibility palette.

   Note: The Tool Tip should include a shortened repeat of the question or statement for the grouping, as well as the desired answer
   e.g. “Check if you will attend Session 1 of the conference”



Tool Tip: Check if you will attend Session 1 of the conference

Screen Reader Precedence: Custom Text

Custom Screen Reader Text:

5. Repeat for each check box

It is a good practice to place the caption to the right of the checkbox (default).

Drop List or Combo Box

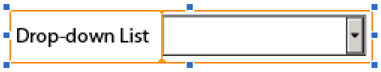
Drop-down lists (also called combination boxes) provide users with a way to select a single choice from a set list of text options.

These types of form controls can create problems for many assistive technologies. For example, providing the multiple select options in a select list will render it totally inaccessible for certain assistive technology and screen reader users. If the intent in using a drop-down list or list box is to allow multiple select options, it is advisable to use checkboxes instead.

If you prefer to create dropdown lists, limit their usage.

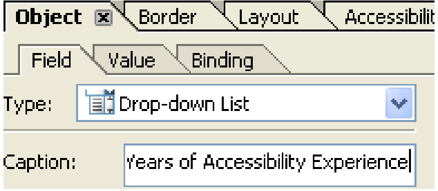
To create accessible drop lists:

1. Drag/Drop the Drop-down list Drop-down List select element from the Object Library onto the desired location within your form.



Drop-down List

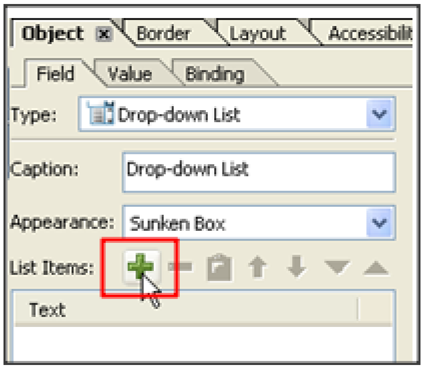
2. Rename the label (**Caption**) for the field in the Object palette | Field Tab



Type: Drop-down List

Caption: Years of Accessibility Experience

3. Add Items in the Object palette | Field Tab by selecting the **Add** button for **each** new item



Type: Drop-down List

Caption: Drop-down List

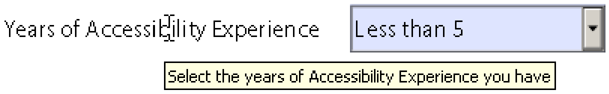
Appearance: Sunken Box

List Items:

4. Enter the question (or statement text) under Tool Tip in the Accessibility Palette

5. Uncheck specify item values from the Object > Binding Tab (this enables assistive technologies to speak the displayed values)

The tooltip will appear when the field (or label) is hovered over and will be read by screen readers.


Years of Accessibility Experience
Less than 5
Select the years of Accessibility Experience you have

Tables

Tables are an effective way to organize and present content in accessible forms. When used appropriately, a table's rows and columns provide a predictable and consistent structure for form content.

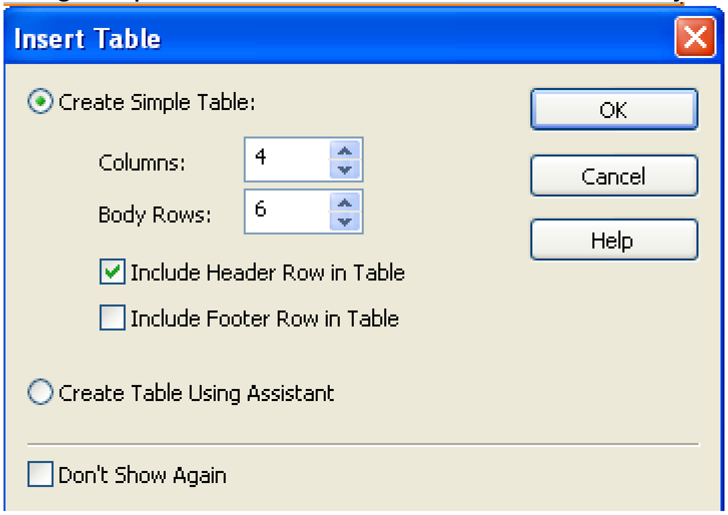
It is best to avoid overly complex tables, such as those with nested tables and sections. Also, they should NOT be used for layout purposes.

The best practice is to keep the table structure as SIMPLE as possible. Simple tables begin with a single header row followed by the body rows.

Success Criterion:
1.3.1 Info and Relationships: Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text – Level A

To create accessible tables:

1. Drag/Drop a **table** element Table from the Object Library and select the appropriate



Insert Table

● Create Simple Table:

Columns: 4

Body Rows: 6

☑ Include Header Row in Table

□ Include Footer Row in Table

○ Create Table Using Assistant

□ Don't Show Again

2. Ensure “Header Row in Table” is selected

Tab Order

Setting a tab order is extremely important for accessibility in a PDF Form. It tells the screen reader the order in which to read your form to a sight-impaired person. The tab order starts with the first object encountered on the form and works its way from top to bottom, then left to right.

Success Criterion:
1.3.2 Meaningful Sequence: When the sequence in which content is presented affects its meaning, a correct reading sequence can be programmatically determined – Level A

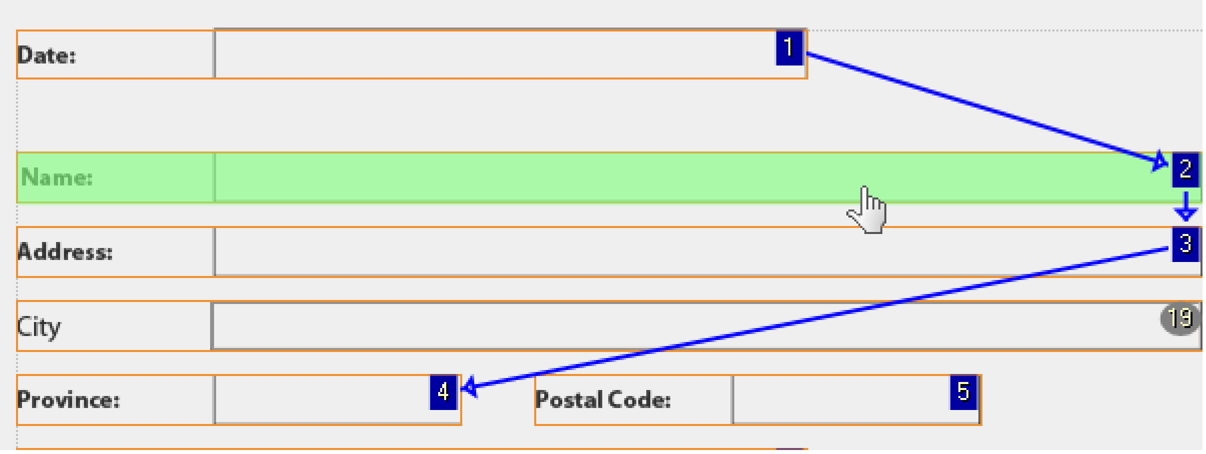
2.1.1 Keyboard: All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user’s movement and not just the endpoints – Level A

2.4.3 Focus Order – Level A
Preparing your PDF form for final delivery means testing and fixing the tab order of your form. As one of the final checks before saving, it is important to ensure that the tabbing/reading order is logical, intuitive and correct.

To check/adjust the tabbing/reading order:

1. Open the **Tab Order** palette to show the current reading/tabbing order of the form from within the Design view.
2. Read the directions the **Tab Order Tips and Hints** window, then click the **OK** button to close it.

   Note: Each element is assigned a number according to its position in the document flow
3. Hover over any element to give you a visual flow of the adjacent ordered elements



4. Reorder Various elements by simply **dragging** them to the appropriate spot in the tab order.

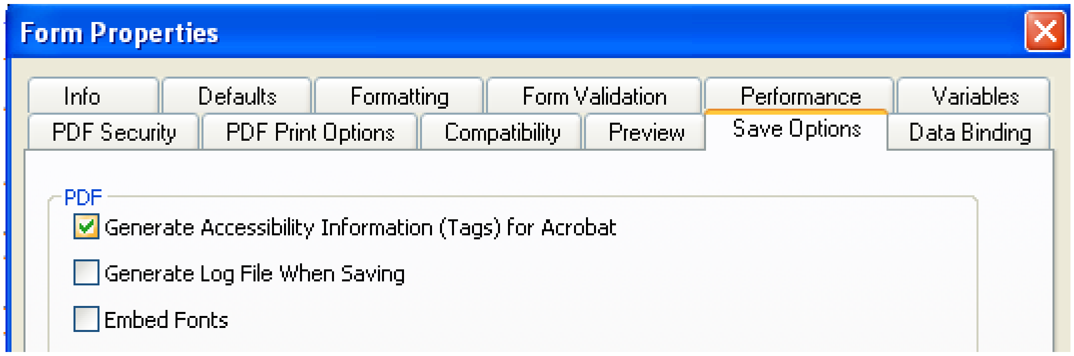
   Note: be sure you have selected “Custom” for the tab order
5. Repeat until the reading/tab order is logical

Final Steps

To create an accessible form, you must configure form properties to have LiveCycle Designer generate accessibility information when the form is saved as a PDF file:
To assign the needed properties:

1. Choose File > Form Properties

2. Select the Save Options tab and check Generate Accessibility Information (Tags) for Acrobat



PDF

☑ Generate Accessibility Information (Tags) for Acrobat

□ Generate Log File When Saving

□ Embed Fonts

3. Choose "OK"