

WCAG 2 Compliance With PDF Forms

A few Notes For This Webinar

1. Please mute your phone line
2. You can post questions in Connect during the session and we will answer what we can at the end of the webinar
 - Further questions will be answered over email or through our blog. www.paciellogroup.com/blog/
 - This session is being captioned and a link to the recording of the webinar will be provided.
 - The recording won't be there right away, so please keep checking the blog and it will be available soon.
 - A PDF version of the slides will also be available.

In This Presentation

1. Background to PDF Forms & LiveCycle
2. Methods for Creating Accessible PDF Forms
3. PDF Forms & WCAG 2.0 Compliance
 - Principle 1: Perceivable
 - Principle 2: Operable
 - Principle 3: Understandable
 - Principle 4: Robust

Why PDF Forms

- Make your PDF documents interactive
- Allows both offline and online completion of forms
- Makes development and distribution of complex forms easier

Building PDF Forms: How?

- Adobe Acrobat Professional
 - Basic form tools available
 - Sufficient for simple / short forms
 - Useful to make scanned documents interactive
- Adobe LiveCycle Designer:
 - More advanced form capabilities
 - Recommended for large, complex forms
 - More accessibility features

LiveCycle Designer Features

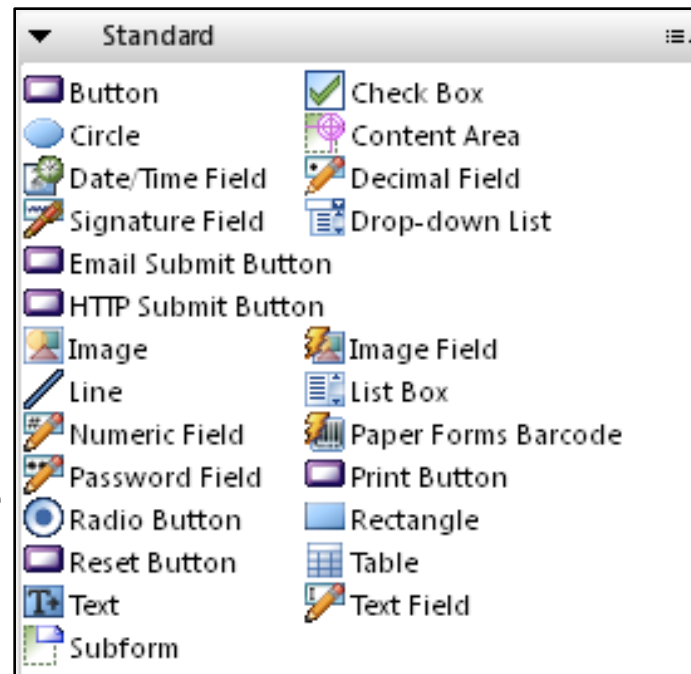
- Accessible Components:

- Choose from the ‘standard’ panel rather than ‘custom’ in Object Library Palette

- Come with accessibility ‘built in’

- Keyboard accessible
- Part of tab order
- name / role / state information exposed to AT

- Custom controls require a lot more manual work to be accessible.



LiveCycle Designer Features (2)

- Accessibility Palette, used to:

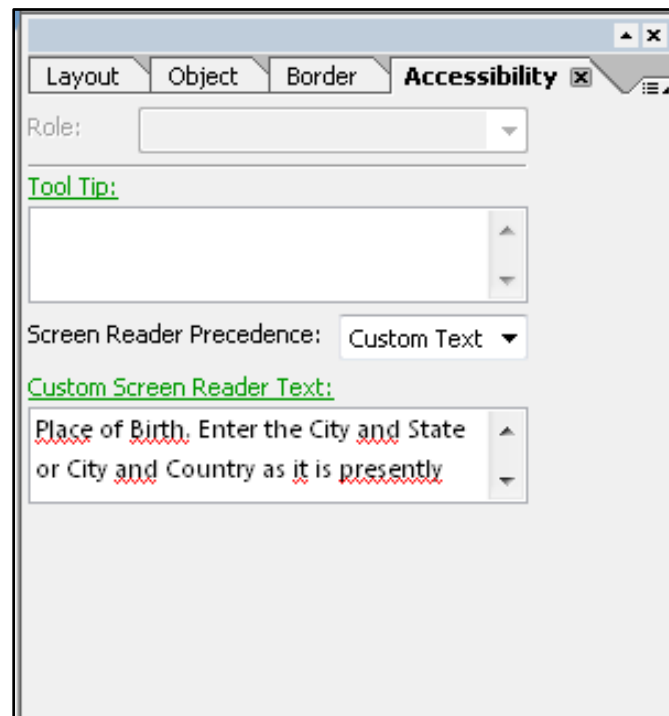
- Provide AT friendly text alternatives for form controls and images.

- Specify ‘screen reader precedence’: which alternative should AT use?

- Custom, caption, tooltip, name, none

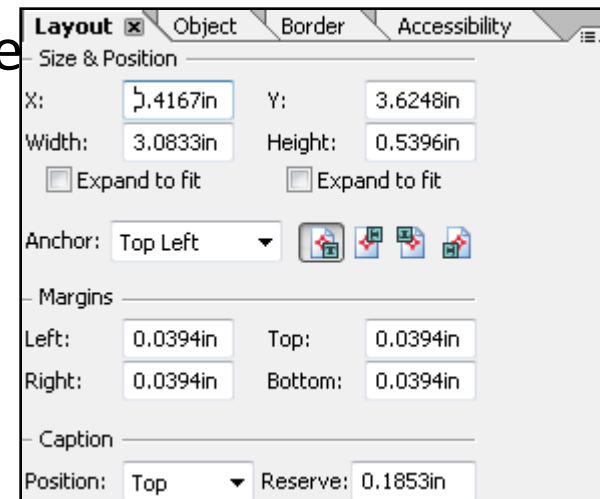
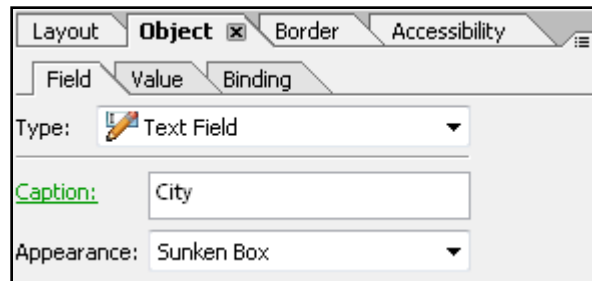
- Assign roles to indicate:

- Heading structure
- Lists
- Tables



LiveCycle Designer Features (3)

- Captions (i.e. Form control labels)
 - Automatically associated with the control
 - Can be positioned visually using ‘Layout’ palette
 - Use default positions best usability
 - Not necessarily the same as name announced by AT
 - Depends on ‘screen reader precedence’



LiveCycle Designer Features (4)

- Tab order palette
 - Allows you to control tab / reading order
 - Determines the order in which objects appear to keyboard and screen reader users
 - Choose either the default or a custom tab order
 - Influence tab order with (Nested) subforms:
 - Self contained units with ‘mini tab order’ for child objects

51	TextField17[0]
52	TextField17[1]
53	TextField18[0]
54	TextField19[0]
55	Text12[0]
56	FatherUS
57	TextField17[2]
58	TextField17[3]
59	TextField18[1]
60	TextField19[1]
61	Text12[1]

Parent No. 45 (Maiden, Previous Marriage, Legal Name Change)? If yes, please complete. (Attach additional pages if nes 46

47 2. 48

49 10. Parents' Informa 50

Father's Name - First & Middle 51 Last 52

Date of Birth (mm/dd/yyyy) 53 Father's Place of Birth 54 U.S. Citizen 55

yes 56

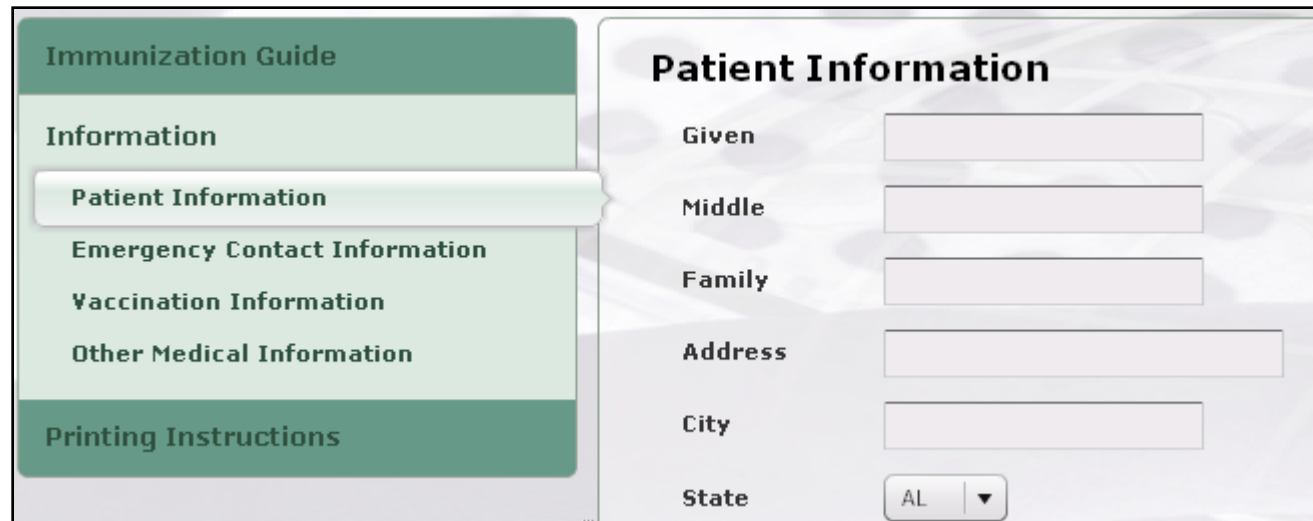
no 58

Mother's Name - First & Middle 57 Last

LiveCycle Designer Features (5)

• Form Guides

- Allow you to split up complex forms in to manageable segments
 - Different 'paths' through form based on user's choices
 - Make forms easier to use through step based approach
- Form guides are rendered as Adobe Flex interface



The screenshot displays a user interface for a form guide. On the left is a vertical navigation menu with a dark green header 'Immunization Guide' and a dark green footer 'Printing Instructions'. The menu items are: 'Information', 'Patient Information' (highlighted with a white background and a dark green border), 'Emergency Contact Information', 'Vaccination Information', and 'Other Medical Information'. On the right is the 'Patient Information' form, which includes the following fields: 'Given', 'Middle', 'Family', 'Address', 'City', and 'State'. The 'State' field is a dropdown menu currently showing 'AL'.

PDF Forms and WCAG 2

About WCAG

- Web Content Accessibility Guidelines (WCAG)
 - Explain the requirements for accessible content
 - Version 2.0 released December 2008
- Organized around the following 4 principles:
 - Perceivable
 - Operable
 - Understandable
 - Robust
- Each principle has guidelines
- Each guideline has success criteria

Principle 1: Perceivable

“Information and user interface components must be presentable to users in ways they can perceive.”

Guideline 1.1: Alternative text

- *“Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.”*
- Use Accessibility Palette to specify text alternatives:
 - Images
 - Form controls
 - Is visual caption sufficient? If so use it for AT
 - Otherwise, set ‘custom screen reader text’

Guideline 1.3: Adaptable

- *“Create content that can be presented in different ways (for example simpler layout) without losing information or structure.”*
 - Basic tables:
 - Use the ‘table’ object
 - Complicated Tables: Use table related roles
 - Apply roles to nested subforms, indicating:
 - Table, header row, body row, footer row
 - Text objects inside rows are considered cells
- Keep tables simple where possible
- Never use tables for layout purposes
 - Unnecessary, LC designer provides positioning freedom

Guideline 1.3: Adaptable (2)

- Create a heading structure
 - Used by screen readers to provide efficient heading navigation:
 - Heading lists
 - Skipping to next / previous heading
 - To apply, use heading roles:
 1. Place text object at beginning of section
 2. Add heading role to the text object
 - Choose either 'heading' or 'heading level <1 – 6>' for nesting
 3. Use formatting to make text look like a heading visually

Guideline 1.4: Distinguishable

- *“Make it easier for users to see and hear content including separating foreground from background.”*
- Don't use color as only means of conveying information
- Make sure foreground / background contrast is sufficient
- Rely on Acrobat Reader features
 - Users can also change their color theme
 - Text can be displayed in any size

Principle 2: Operable

“User interface components and navigation must be operable.”

Guideline 2.1: Keyboard Accessible

- *“Make all functionality available from a keyboard.”*
 - All interactive controls must be reachable and operable without a mouse.
- Where possible, use standard components
 - Keyboard accessibility built in
 - For custom controls, manually handle keyboard input

Guideline 2.1: Keyboard Accessible (2)

- Provide a logical tab/reading order
 - Tab and reading order are handled as one, but:
 - Reading order: all objects (including static text)
 - Tab order: only interactive controls
 - Default order
 - left to right, top to bottom
 - Use subforms or fragments to manipulate order, e.g. Into columns
 - Each subform has its own sub- taborder
 - Use tab order palette to inspect tab order
 - Numbers and arrows visualize the current order
 - If needed, switch to custom tab order
 - Rearrange one or more items at a time to modify the current order
 - Moving them up or down in tab order palette
 - Test with keyboard and screen reader to confirm results

Guideline 2.4: Navigable

- *“Provide ways to help users navigate, find content, and determine where they are.”*
- Create form guides to enhance form navigation
- Provide heading structure
- Use ‘list’ and ‘list item’ roles to mark subforms up as lists
 - Lists can be nested

Principle 3: Understandable

“Information and the operation of user interface must be understandable.”

Guideline 3.1: Readable

- *“Make text content readable and understandable.”*
- Set the language
 - For the entire document, specify ‘form locale’ using the ‘form properties’ window
 - For parts of the form, use the ‘object palette’ to set the locale
 - E.g. on a subform object
- Provide glossaries and Indices
- Prevent disruptive changes of context
 - Be careful with scripting
- Form validation & handling
 - Set field as ‘required’ in object value palette
 - Specify validation patterns and warnings

Principle 4: Robust

“Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.”

Guideline 4.1: Compatible

- *“Maximize compatibility with current and future user agents, including assistive technologies.”*
- Use standardized components
 - Role will be automatically provided
 - State will automatically provided, e.g.
 - Checked
 - Read only

Wrapping Up

- To Summarize:
 - PDF forms can be made fully accessible and WCAG 2.0 compliant
 - Use standard form controls
 - Provide screen reader friendly text alternatives for controls and images
 - Provide a logical heading structure using heading roles
- Adobe and Accessibility
 - For more information, visit the Adobe Accessibility Portal:
 - <http://www.adobe.com/accessibility>
 - Questions?

Further Reading

- Go to:
 - <http://www.adobe.com/accessibility/>
 - Detailed guides for creating accessible PDF documents and forms
- For recordings and slides from this session:
 - <http://www.paciellogroup.com/blog/>
- Questions?
 - cpike@paciellogroup.com
 - access@adobe.com