



PDF accessibility acceptance checklist

What is the document content engine?

PDFs are generated using the document content engine. The benefits of using this advanced engine are:

The Document Content Engine framework allows Authors to create document templates with semantic HTML and CSS. The rendering of documents remains consistent across multiple channels. Client branding enables complex reuse. Paragraph rules in Pega segregate content, allowing reusable headers and footers, similar to web pages with regions and landmarks.

Additional visual elements can be included, such as assets from CDN (content delivery network) or different types of image files (JPG, SVG). Custom fonts can also be used but provide fallbacks using the OS.

General checks in making an accessible PDF

These checks are not agnostic to the generation of a PDF. Using the Document Content Engine will add reassurance as you create a PDF with semantic HTML, but follow these checks if HTML expertise is limited:

Document structure

- The document must include a title and indicate the default language.
- To check the title for the PDF, go to File > Properties > Description, and the title field will have the same title as the document. See the related [WCAG](#) section: 2.4 Page Titled (Level A) and example technique in [G88 providing descriptive titles for a web page](#)
- Documents over twenty pages must include bookmarks to make navigation easier. See related [WCAG](#) section: 1.3.1 Info and Relationships. Ensure the table of contents is marked with the appropriate semantic HTML tags (ul or unordered list. See [PD4ML](#) if the engine is available for their TOC tags).
- Use headers and footers to provide orientation for the user. Define header and footer layouts using HTML elements. See related [WCAG](#) section: 2.4.8 location. See [PD4ML](#) header footer options.

- Set your document language if supporting different locales. The 'lang' attribute can be set on the html tag. See the related [WCAG](#) section: Language of Page and related techniques in [H57 using the language attributes on the html element](#)

Content order

- Manually check the reading order of your content. See the related [WCAG](#) section: 1.3.1 Info and Relationships, Meaningful Sequence 1.3.2, Keyboard 2.1.1, Keyboard no exception 2.1.3 and Focus Order 2.4.3

To test:

- Verify that the document's reading order is correct using an accessibility API.
- Listen to the elements being announced correctly using a screen reader.

Colour

- Manually check the colour contrast of your text sufficiently stands out from the background.
- Meet a 4.5:1 colour contrast ratio regardless of what size your text is. See [WCAG](#) section understanding contrast minimum.
- Test high-contrast colours, as some use custom OS system colours.
- Use the colour contrast analyser tool from [TPGi](#)
- *Note:* Large text is defined as text of 24px or larger, or larger than 19px for bold text

Images

- Check your images. See the related [WCAG](#) section: 1.1.1. Non-text content
- Make sure that images in the document either have alternate text or are marked as 'Decorative figure' by selecting the checkbox.
- See the example technique in [G94 providing alternative text for non-content](#)

Tables

- Use table headers appropriately.
- Define a table caption. They can be hidden from sighted users using CSS, but they will still be read aloud by screen readers. This makes it easier for visually impaired users to understand the information presented in the table.
- Merged table cells using the Rowspan or Colspan attribute.
- See the related [WCAG](#) section: 1.3.1 Info and Relationships and example technique in [H51 using table markup](#)

Lists

- Ensure you use the HTML semantic list element. See the related [WCAG](#) section: 1.3.1 Info and Relationships. See example techniques in [G115 using semantic elements to mark up structure](#)

Forms

- Use a descriptive label and place them next to the corresponding field. Indicate mandatory fields
- Check that the keyboard can access the fields
- Provide a name, role and value
- See the related [WCAG](#) section: 1.3.1 Info and Relationships

To test:

- Use an accessibility API tool to verify the form control and its name, role, and value.

Links

- When using a screen reader to read a PDF document, ensure the link is read correctly and accurately describes its destination. See the related [WCAG](#) section.

Testing tools

- Install for windows the [PDF accessibility checker \(PAC\)](#)
- Run the [Acrobat Pro accessibility checker](#)
- Run through this simple checklist above: check the document properties and markup of lists, links, forms and tables.