PDF Techniques for WCAG 2.0

This Web page lists PDF Techniques from Techniques for WCAG 2.0: Techniques and Failures for Web Content Accessibility Guidelines 2.0. Technology-specific techniques do not replace the general techniques: content developers should consider both general techniques and technology-specific techniques as they work toward conformance.

Publication of techniques for a specific technology does not imply that the technology can be used in all situations to create content that meets WCAG 2.0 success criteria and conformance requirements. Developers need to be aware of the limitations of specific technologies and provide content in a way that is accessible to people with disabilities.

For information about the techniques, see Introduction to Techniques for WCAG 2.0. For a list of techniques for other technologies, see the Table of Contents.

Table of Contents

PDF Technology Notes

Introduction

The Portable Document Format (PDF) is a file format for representing documents in a manner independent of the application software, hardware, and operating system used to create them, as well as of the output device on which they are to be displayed or printed. PDF files specify the appearance of pages in a document in a reliable, device-independent manner. The PDF specification was introduced by Adobe Systems in 1993 as a publicly available standard. In January 2008, PDF 1.7 became an ISO standard (ISO 32000-1).

Of note for accessibility is PDF/UA (Universal Accessibility) which became an ISO Standard in July 2012 (ISO 14289-1:2012 (See PDF/UA (ISO 14289-1:2012).) The scope of PDF/UA is not meant to be a techniques (how-to) specification, but rather a set of guidelines for creating more accessible PDF. The specification describes the required and prohibited components and the conditions governing their inclusion in or exclusion from a PDF file in order for the file to be available to the widest possible audience, including those with disabilities. The mechanisms for including the components in the PDF stream are left to the discretion of the individual developer, PDF generator, or PDF viewing agent. PDF/UA also specifies the rules governing the behavior for a conforming reader.

PDF Accessibility Support

PDF includes several features in support of accessibility of documents to users with disabilities. The core of this support lies in the ability to determine the logical order of content in a PDF document, independently of the content's appearance or layout, through logical structure and Tagged PDF. Applications can extract the content of a document for presentation to users with disabilities by traversing the structure hierarchy and presenting the contents of each node. For this reason, producers of PDF files must ensure that all information in a document is reachable by means of the structure hierarchy.

Logical Structure

PDF's logical structure features (introduced in PDF 1.3) provide a mechanism for incorporating structural information about a document's content into a PDF file. Such information might include, for example, the organization of the document into chapters, headings, paragraphs and sections or the identification of special elements such as figures, tables, and footnotes. The logical structure features are extensible, allowing applications that produce PDF files to choose what structural information to include and how to represent it, while enabling PDF consumers to navigate a file without knowing the producer's structural conventions.

PDF logical structure shares basic features with standard document markup languages such as HTML, SGML, and XML. A document's logical structure is expressed as a hierarchy of structure elements, each represented by a dictionary object. Like their counterparts in other markup languages, PDF structure elements can have content and attributes. In PDF, rendered document content takes over the role occupied by text in HTML, SGML, and XML.

A PDF document's logical structure is stored separately from its visible content, with pointers from each to the other. This separation allows the ordering and nesting of logical elements to be entirely independent of the order and location of graphics objects on the document's pages.

The logical structure of a document is described by a hierarchy of objects called the structure hierarchy or structure tree. At the root of the hierarchy is a dictionary object called the structure tree root, located by means of the StructTreeRoot entry in the document catalog. See Section 14.7.2, ("Structure Hierarchy") in PDF 1.7 (ISO 32000-1): Table 322 shows the entries in the structure tree root dictionary. The K entry specifies the immediate children of the structure tree root, which are structure elements.

Tagged PDF

Tagged PDF (PDF 1.4) is a stylized use of PDF that builds on PDF's logical structure framework. It defines a set of standard structure types and attributes that allow page content (text, graphics, and images) to be extracted and reused for other purposes. It is intended for use by tools that perform the following types of operations:

- Simple extraction of text and graphics for pasting into other applications.
- Automatic reflow of text and associated graphics to fit a page of a different size than was assumed for the original layout.
- Processing text for such purposes as searching, indexing, and spell-checking.
- Conversion to other common file formats (such as HTML, XML, and RTF) with document structure and basic styling information preserved.
- Making content accessible to people who rely on assistive technology.

PDF File Production and Accessibility

PDF files may be produced either directly by application programs or indirectly by conversion from other file formats or imaging models. In addition, tools exist for inspecting, checking, and repairing PDF files for accessibility. The following sections provide representative lists of applications and tools typically used for these functions.

These notes do not, and cannot, provide an exhaustive list, nor do they endorse particular applications and tools. Rather they provide a snapshot of tools in fairly wide use at the time the WCAG Working Group undertook to review and publish techniques for producing PDF documents. As with any software, application support for PDF accessibility will vary with different versions, with the formatting requirements of specific PDF documents, and with actual usage of the application. That is, the tools can be used properly to produce appropriate tags, etc..

In general, newer tools will provide greater support than earlier ones. The tools' vendors are the source of authoritative information about their support for PDF accessibility.

Generating PDF Files

Many applications can generate PDF files directly, and some can import them as well. This direct approach is preferable, since it gives the application access to the full capabilities of PDF, including the imaging model and the interactive and document interchange features. Alternatively, applications that do not generate PDF directly can produce PDF output indirectly. There are two principal indirect methods:

- The application describes its printable output by making calls to an application programming interface (API) such as GDI in Microsoft® Windows® or QuickDraw in the Apple Mac OS. A software component called a printer driver intercepts these calls and interprets them to generate output in PDF form.
- The application produces printable output directly in some other file format, such as PostScript, PCL, HPGL, or DVI, which is converted to PDF by a separate translation program.

Although these indirect strategies are often the easiest way to obtain PDF output from an existing application, the resulting PDF files may not make the best use of the high-level PDF imaging model relied upon to expose the semantics of the document. This is because the information embodied in the application's API calls or in the intermediate output file often describes the desired results at too low a level. Any higher-level information maintained by the original application has been lost and is not available to the printer driver or translator.

For example, since the printer driver or translator targets correct visual output, information about the semantics of the document may not be sent at all or may be ignored when creating the PDF output. As a result, headings may not be tagged as such, or link text may not be associated with its link object. Check with the vendor of any PDF authoring tool in order to understand how to use the tool in a way that produces the best tagged output.

PDF Authoring Tools that Provide Accessibility Support

- Adobe Acrobat's PDFMaker - PDFMaker is part of Adobe Acrobat which adds macros to many business applications such as Microsoft Office, AutoCAD and Lotus Notes that support the conversion of content from the original format to tagged PDF.
- Adobe FrameMaker - Desktop publishing application from Adobe Systems that directly exports tagged PDF and provides support for alternative text descriptions.
- Adobe InDesign - Page layout and desktop publishing application from Adobe Systems that directly exports tagged PDF and provides support for alternative text descriptions.
- Adobe LiveCycle Designer - Windows-based forms design application from Adobe Systems that directly exports tagged PDF interactive forms and provides support for alternative text descriptions; can be invoked standalone or from within Acrobat Pro.
- LibreOffice - Open-source word processing software from The Document Foundation that can export tagged PDF.
- Lotus Symphony Documents - Word-processing software from IBM that can export tagged PDF.
- Microsoft® Word - Word processing application from Microsoft Corporation that can export tagged PDF using the save as XPS or PDF utility.
- OpenOffice.org Writer - Open source word-processing software from Sun Microsystems Inc. that can export tagged PDF using the Export as PDF utility.
- CommonLook Office for Microsoft Office from Netcentric Technologies is an add-in to Microsoft® Word and PowerPoint that makes it possible to create tagged PDF documents. CommonLook Office provides tools to allow content authors to run accessibility tests in the Microsoft Word and PowerPoint environments and to remediate accessibility issues prior to conversion to PDF.
- Xenos Axess™ for Accessible Statements - PDF software integrates with an organization's existing enterprise content management

(ECM) infrastructure to capture high-volume print streams and automatically transform them into tagged PDFs.
- WordPerfect® Office - Word-processing software from Corel that can be used to create, mark up, and share tagged PDF documents.
- Microsoft Office 10 - a suite of desktop office applications that creates tagged PDF.

Note: Care should be taken when choosing PDF creation tools from the many available, as some may not support creation of tagged PDF files.

Accessibility Checking and Repair

Adobe Acrobat Pro. Adobe Acrobat Pro is an application that creates and edits PDF files. It has a number of tools for evaluating and repairing the accessibility of PDF files, including access to the structure root through the tags panel, the ability to directly manipulate the reading order through the order panel, a built-in accessibility checker, and the Touch Up Reading Order tool which provides a graphical mechanism for assessing and repairing the accessibility of a PDF document.

Commonlook™ PDF. Commonlook PDF. Commonlook PDF is a plug-in for Adobe Acrobat Pro from Netcentric Technologies. CommonLook PDF helps identify, report and correct the most common accessibility problems, including the proper tagging of images, tables, forms and other non-textual objects.

PAC - the PDF Accessibility Checker. PAC is a free tool developed and distributed by the «Access for all» Foundation to evaluate the accessibility of PDF documents and PDF forms. PAC offers the added possibility of displaying a preview of the structured PDF document in a web browser. The PAC preview shows which tags are included in the PDF document and presents the accessible elements in the same way as they would be interpreted by assistive technologies (such as screen readers). PAC also provides an accessibility report which lists the detected accessibility errors. Clicking the links in the report displays the most probable source of the error within the document.

API Inspection Tools

- aDesigner - a disability simulator from the Eclipse Foundation that helps designers ensure that content is accessible and usable by visually impaired users.
- inspect32 - part of the Microsoft Windows Software Development Kit (SDK) that allows developers and testers to examine the accessible properties of UI components.
- PDDOMView - part of Acrobat_Accessibility_9.1.zip which contains files that can be used by Windows clients of the accessibility interfaces described in the Accessibility API Reference document.
- UISpy - part of the Microsoft Windows Software Development Kit (SDK) that allows developers and testers to view and interact with the user interface (UI) elements of an application.

User Agents

PDF User Agents with accessibility support include:

- Adobe Acrobat Pro - PDF Authoring Tool, Editor, and Viewer from Adobe Systems which is compatible with MSAA devices on the Windows platform. Has a number of built in accessibility features including text to speech (Read Out Loud), high contrast display, reflow for large print display, auto scroll, accessibility full check, accessibility quick check, touch up reading order tool, and an accessibility setup assistant.
- Adobe Reader - Freely distributed PDF Viewer from Adobe Systems which is compatible with MSAA devices on the Windows platform. Has a number of built in accessibility features including text to speech (Read Out Loud), high contrast display, reflow for large print display, auto scroll, accessibility quick check, and an accessibility setup assistant.
- Kurzweil 3000™ - a comprehensive reading, writing and learning software solution from Kurzweil Educational Systems® which reads PDF files using text to speech facilities.

Adobe Reader and Acrobat Support for Accessibility APIs

Adobe provides methods to make the content of a PDF file available to assistive technology such as screen readers:

- On the Microsoft® Windows® operating system, Acrobat and Adobe Reader export PDF content as Component Object Model (COM) objects. Accessibility applications such as screen readers can interface with Acrobat or Adobe Reader in two ways:
  - Through the Microsoft Active Accessibility (MSAA) interface, using MSAA objects that Acrobat or Adobe Reader exports
  - Directly through exported COM objects that allow access to the PDF document's internal structure, called the Document Object Model (DOM).
- On UNIX® platforms, Adobe Reader supports the Gnome accessibility architecture. C-based Accessibility Toolkit (ATK) interfaces are available.

The DOM and MSAA models are related, and developers can use either or both. Acrobat issues notifications to accessibility clients about interesting events occurring in the PDF file window and responds to requests from such clients. Recent versions of Acrobat and Reader have enhanced the support for accessibility interfaces:

- MSAA interfaces are supported in Acrobat/Reader 5.0 and later.
- In Acrobat/Reader 6.0 and later, information about the underlying PDF structure is made available through direct COM objects that represent the PDF DOM. The DOM accessibility interfaces provide somewhat more extensive access.
- In Acrobat/Reader 7.0 and later, ATK and expanded DOM support is available.
- The Linux®, Solaris™, AIX®, and HP-UX versions of Adobe Reader implement C-based ATK interfaces, allowing screen readers, screen magnifiers, and on-screen keyboards to query an Accessibility Technology - Service Providers Interface (AT-SPI) registry for applications that are accessible.
- The DOM has been expanded to provide enhanced caret, selection, and focus support, as well as the new interfaces IPDDomDocument, ISelectText, and IPDDomNodeExt.

Assistive Technology Support

- JAWS 12 for Windows - screen reader from Freedom Scientific. Support for PDF started with JAWS version 4.
- MAGic 11 - screen magnifier from Freedom Scientific
- NVDA 2011.1 - NonVisual Desktop Access, open source screen reader distributed by NV Access. Providing feedback via synthetic speech and Braille, NVDA allows blind and vision-impaired people to access and interact with the Windows operating system and many third party applications.

- Supernova Access Suite 12.02 - full screen reader offering magnification, speech, and Braille support from Dolphin. Support for PDF started with HAL version 5.
- System Access To Go - screen reader from Serotek Corporation
- VoiceOver - screen reader for Mac OS X v10.6 Snow Leopard
- Window-Eyes 7.2 - screen reader from GW Micro. Window-Eyes was the first screen reader to provide support for PDF files, in Window-Eyes 4.2.
- ZoomText 9.1 - screen magnifier and screen reader from Ai Squared, with support for Adobe Acrobat and Reader:
    - PDF documents can be read using both AppReader and DocReader (without special settings)
    - PDF documents can be read in all Windows operating systems (without special settings)
    - AppReader and DocReader start instantly in Adobe Reader
    - PDF documents can be read with greater accuracy and without paging delays
    - PDF documents can be read in Internet Explorer (with the Adobe Reader plug-in)
    - Special Adobe Reader settings are no longer needed to obtain optimal reading

Related References

- Editing PDF Tags in Adobe Acrobat
- Editing PDF Tags with the Touch Up Reading Order Tool
- Adobe Accessibility Resource Center
- Adobe Acrobat Accessibility Training Resources
- Accessing PDF Documents with Assistive Technology
- PDF Specification Archives
- PDF 1.7 Reference: ISO approved copy of the ISO 32000-1
- PDF Accessibility API Reference - How AT developers can use Acrobat MSAA and IPDDom interfaces to provide access to PDF content
- PDF/UA (ISO 14289-1:2012)
- WebAIM PDF Accessibility
- Create accessible PDFs using Microsoft Office 10

PDF1: Applying text alternatives to images with the Alt entry in PDF documents

Applicability

Tagged PDF documents with images

This technique relates to:

- Success Criterion 1.1.1 (Non-text Content)
    - How to Meet 1.1.1 (Non-text Content)
    - Understanding Success Criterion 1.1.1 (Non-text Content)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF1. Also see PDF Technology Notes.

Description

The objective of this technique is to provide text alternatives for images via an /Alt entry in the property list for a Tag. This is normally accomplished using a tool for authoring PDF.

PDF documents may be enhanced by providing alternative descriptions for images, formulas, and other items that do not translate naturally into text. In fact, such text alternatives are required for accessibility: alternate descriptions are human-readable text that can be vocalized by text-to-speech technology for the benefit of users with vision disabilities.

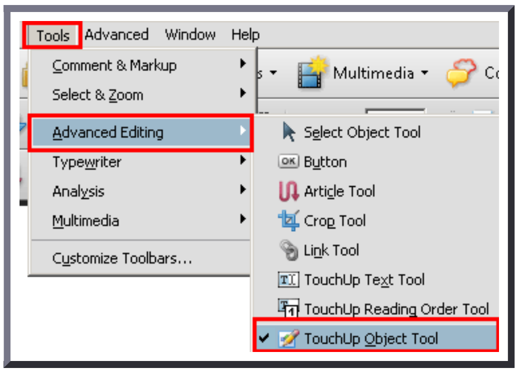
When an image contains words that are important to understanding the content, the text alternative should include those words. This will allow the alternative to accurately represent the image. Note that it does not necessarily describe the visual characteristics of the image itself but must convey the same meaning as the image.

Examples

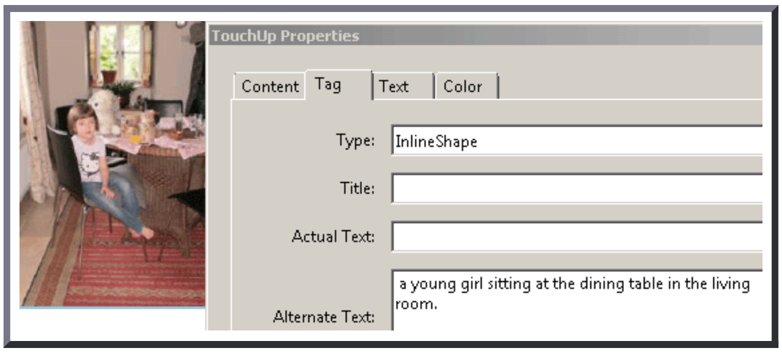
Example 1: Adding an /Alt entry to an image using Adobe Acrobat 9 Pro's TouchUp Object Tool

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. Choose Tools > Advanced Editing > TouchUp Object Tool



2. Access the context menu for the image and choose Properties.
3. On the TouchUp Properties dialog, select the Tag tab.
4. On the Tag panel, type the text alternative in the Alternate Text text box.



Type: InlineShape

Title:

Actual Text:

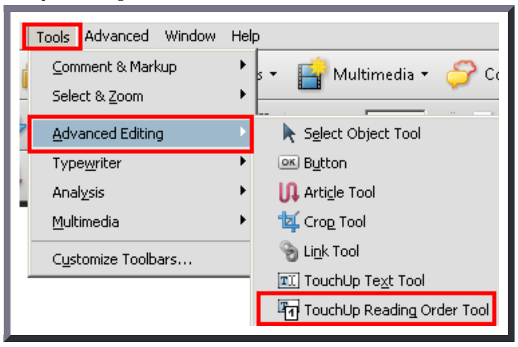
Alternate Text: a young girl sitting at the dining table in the living room.

This example is shown in operation in the working example of Adding an /Alt entry to an image.

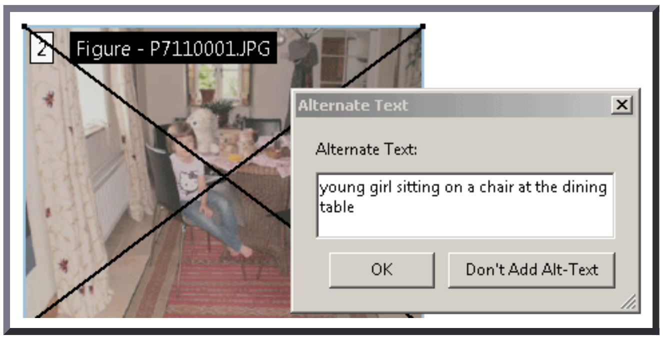
Example 2: Adding an /Alt entry to an image using Adobe Acrobat 9 Pro's TouchUp Reading Order Tool

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. Choose Tools > Advanced Editing > TouchUp Reading Order Tool



2. The TouchUp Reading Order dialog will be displayed.
3. Right-click on the image and choose Edit Alternate Text.
4. The Alternate Text dialog will be displayed.
5. Type the text alternative in the Alternate Text text box.



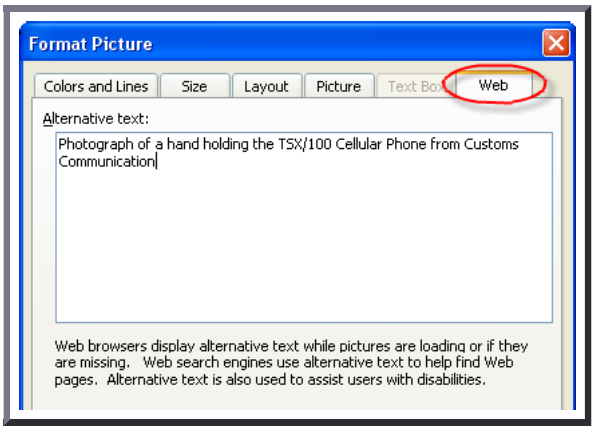
Alternate Text: young girl sitting on a chair at the dining table

Example 3: Adding an /Alt entry to an image in PDF documents generated using Microsoft Word

This example is shown with Microsoft Word. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

Word 2000-2003

1. Right-click on the image and choose Format Picture
2. Select the Web tab
3. Type the alternative text into the text box provided and then click OK.

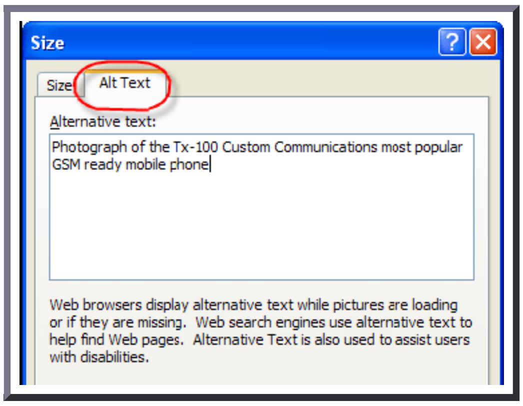


Alternative text:
Photograph of a hand holding the TSX/100 Cellular Phone from Customs Communication

Web browsers display alternative text while pictures are loading or if they are missing. Web search engines use alternative text to help find Web pages. Alternative text is also used to assist users with disabilities.

Word 2007

1. Right-click on the image and choose Size
2. Select the Alt Text tab
3. Type the alternative text into the text box provided and then click OK.



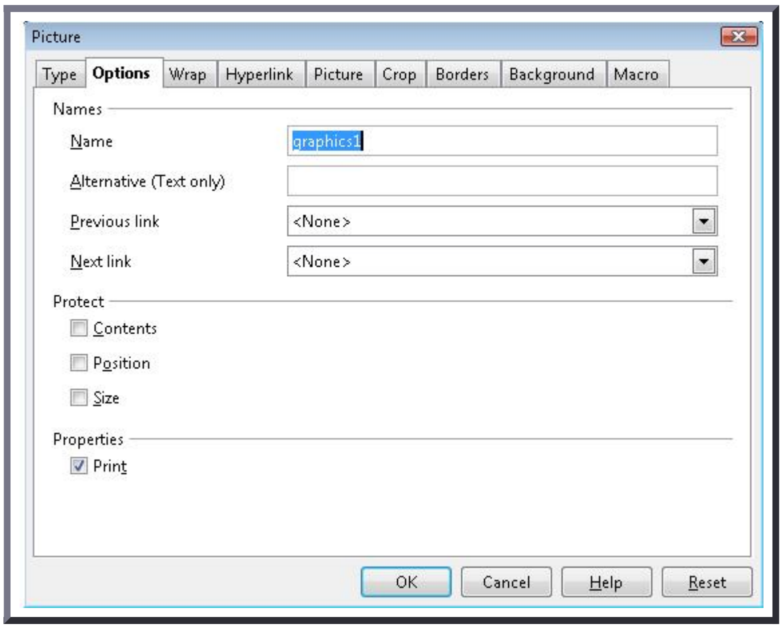
Alternative text:
Photograph of the Tx-100 Custom Communications most popular GSM ready mobile phone

Web browsers display alternative text while pictures are loading or if they are missing. Web search engines use alternative text to help find Web pages. Alternative Text is also used to assist users with disabilities.

Example 4: Adding an /Alt entry to an image in PDF documents generated using OpenOffice.org Writer 2.2

This example is shown with Open Office.org Writer. There are other software tools that perform similar functions. See the list of other software tools in [PDF Authoring Tools that Provide Accessibility Support](#).

1. Access the context menu for the image and choose Picture...
2. Select the Options tab
3. Type the alternative text into Alternate (Text Only) text box and click OK.



Names
Name: graphics1
Alternative (Text only):
Previous link: <None>
Next link: <None>
Protect
☐ Contents
☐ Position
☐ Size
Properties
☑ Print

Example 5: Adding a text alternative to an image in a PDF document using an /Alt entry

The /Alt property used on an image of mountains with a moon and trees typically would be used like this (typically accomplished by an authoring tool):

```
/Figure <</Alt (Sketch of Mountains with moon rising over trees) >>
```

The image might also be represented by a tag with a different name. A different name might be used because the tag name is written in a language other than English or because a specific tool uses a different name for some other reason. In this situation, it is also necessary that the RoleMap contained within the StructTreeRoot for the PDF document contain an entry which explicitly maps the name of the tag used for the image with the standard structure type used in PDF documents (in this case, Figure). If the RoleMap contains only an entry mapping Shape tags to Figure tags, the rolemap information would appear as follows:

```
/RoleMap << /Shape /Figure >>
```

In this case, the usage of the /Alt entry as follows would also be correct:

```
/Shape <</Alt (Crater Lake in the summer, with the blue sky, clouds and crater walls perfectly reflected in the lake) >>
```

Note that the /Alt entry in property lists can be combined with other entries.

Resources

Resources are for information purposes only, no endorsement implied.

- Section 14.9.4 (Replacement Text) in PDF 1.7 (ISO 32000-1)
- Acrobat and Accessibility
- PDF Reference 1.6, 10.8.2 Alternate Descriptions
- PDF and Accessibility

Related Techniques

- G94: Providing short text alternative for non-text content that serves the same purpose and presents the same information as the non-text content

Tests

Procedure

1. Verify the images which need equivalents have /Alt entries on an enclosing tag by one of the following:
   - Read the PDF document with a screen reader, listening to hear that the equivalent text is read when tabbing to the non-text object (if it is tabbable) or hearing the alternative text read when reading the content line-by-line.
   - Using a PDF editor, check that a text alternative is displayed for each image.
   - Use a tool which is capable of showing the /Alt entry value, such as aDesigner, to open the PDF document and view the GUI summary to read the text alternatives for images.
   - Use a tool that exposes the document through the accessibility API and verify that images have required text equivalents.

Expected Results

- Check 1 is true for each image in the document which needs a text equivalent.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF2: Creating bookmarks in PDF documents

Applicability

Tagged PDF documents

This technique relates to:

- Success Criterion 2.4.5 (Multiple Ways)
  - How to Meet 2.4.5 (Multiple Ways)
  - Understanding Success Criterion 2.4.5 (Multiple Ways)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF2. Also see PDF Technology Notes.

Description

The intent of this technique is to make it possible for users to locate content using bookmarks (outline entries in an Outline dictionary) in long documents.

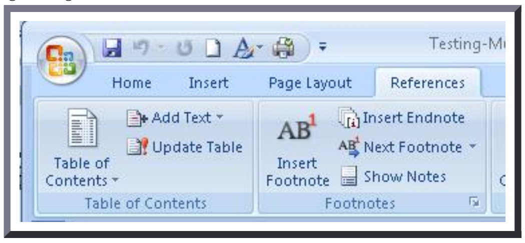
A person with cognitive disabilities may prefer a hierarchical outline that provides an overview of the document rather than reading and traversing through many pages. This is also a conventional means of navigating a document that benefits all users.

Examples

Example 1: Converting a table of contents created with Microsoft Word 2007 and creating bookmarks for Adobe Reader 9 and Acrobat 9 Pro

This example is shown with Microsoft Word and Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

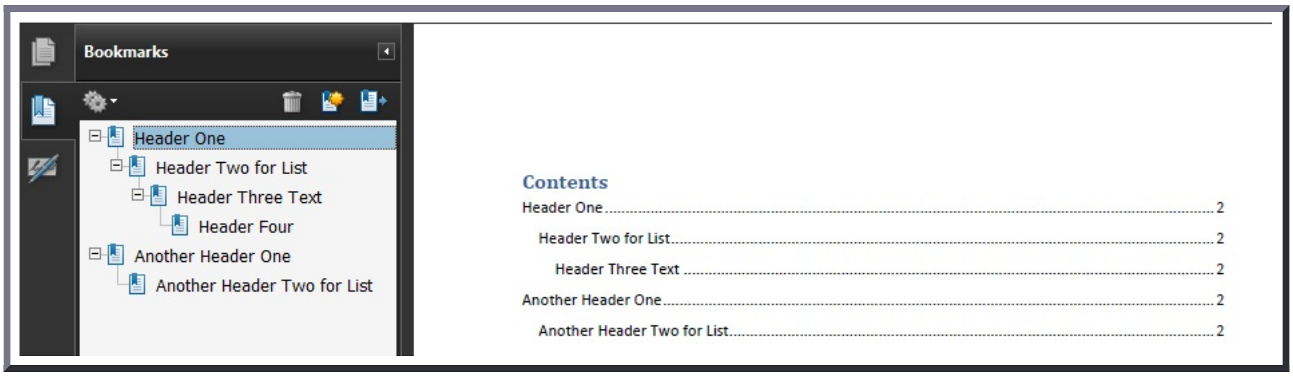
1. Create a table of contents at the beginning of the Word document.



2. Use Save as... > Adobe PDF to convert the Word document to PDF, specifying both of the following:
   - Enable Accessibility and Reflow with Tagged Adobe PDF
   - Convert Word Headings into Bookmarks

The table-of-contents entries in the converted document will be linked to the headings in the document.

In addition, the headings will appear as PDF Bookmarks in the left-hand Navigation pane.


Bookmarks

- Header One
  - Header Two for List
    - Header Three Text
      - Header Four
- Another Header One
  - Another Header Two for List

Contents

Header One 2
Header Two for List 2
Header Three Text 2
Another Header One 2
Another Header Two for List 2

If the document provides a glossary and/or index, these sections should have headings that appear in the table of contents (and thus as bookmarks in the Navigation pane). The table of contents also should be marked up with a heading so it is bookmarked as well.

If this markup has not been done in the authoring tool, Adobe Acrobat Pro can be used to provide the tags. See PDF9: Providing headings by marking content with heading tags in PDF documents if you need to modify converted headings or add new ones.

This example is shown in operation in the working example of creating bookmarks with Word 2007.

Example 2: Converting a table of contents created with OpenOffice.org Writer 2.2 and creating bookmarks for Adobe Reader 9 and Acrobat 9 Pro

This example is shown with OpenOffice.org Writer and Adobe Acrobat Pro and Reader. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. Create a table of contents at the beginning of the OpenOffice.org Writer document:
   - Insert > Indexes and Tables... > Indexes and Tables > Insert Index/Table
2. Use File > Export as PDF... to convert the document to PDF, specifying Tagged PDF in the Options dialog.

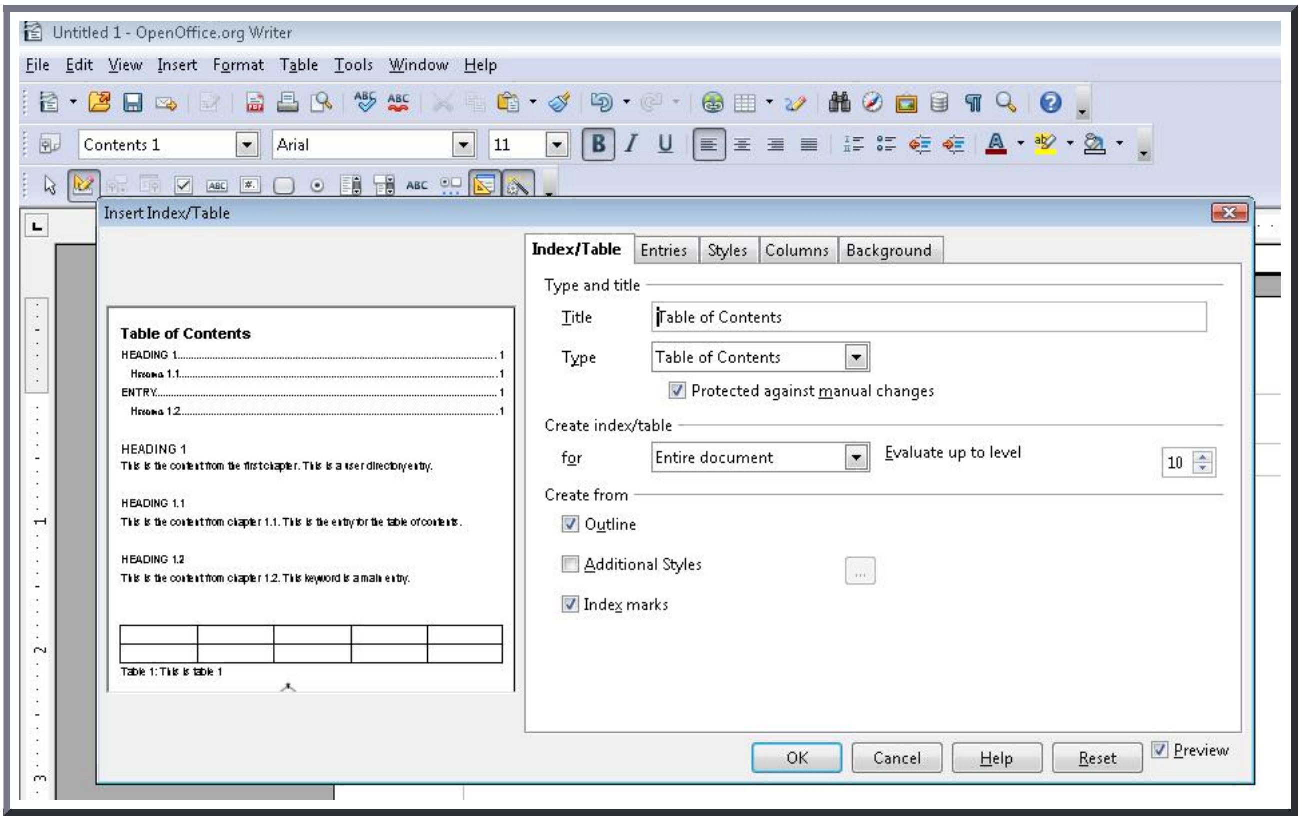


Table of Contents
HEADING 1 1
Heading 1.1 1
ENTRY 1
Heading 1.2 1

HEADING 1
This is the content from the first chapter. This is a user directory entry.

HEADING 1.1
This is the content from chapter 1.1. This is the entry for the table of contents.

HEADING 1.2
This is the content from chapter 1.2. This keyword is a main entry.

Table 1: This is table 1

Type and title
Title: Table of Contents
Type: Table of Contents
☑ Protected against manual changes
Create index/table
for: Entire document
Evaluate up to level: 10
Create from
☑ Outline
☐ Additional Styles
☑ Index marks
☑ Preview

The table-of-contents entries in the converted document will be linked to the headings in the document, and will appear as PDF Bookmarks in the left-hand Navigation pane. The OpenOffice.org Table of Contents and Bookmarks look the same as they appeared in Example 1.

This example is shown in operation in the working example of creating bookmarks with OpenOffice Writer.

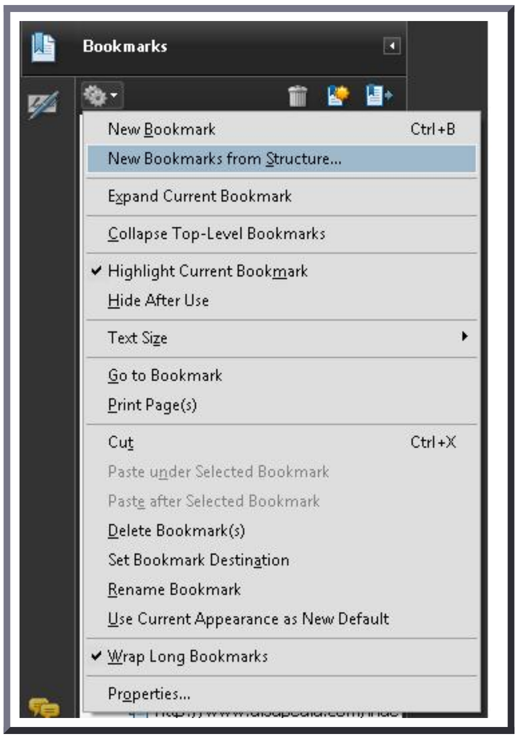
Example 3: Adding bookmarks using Adobe Acrobat 9 Pro after conversion

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

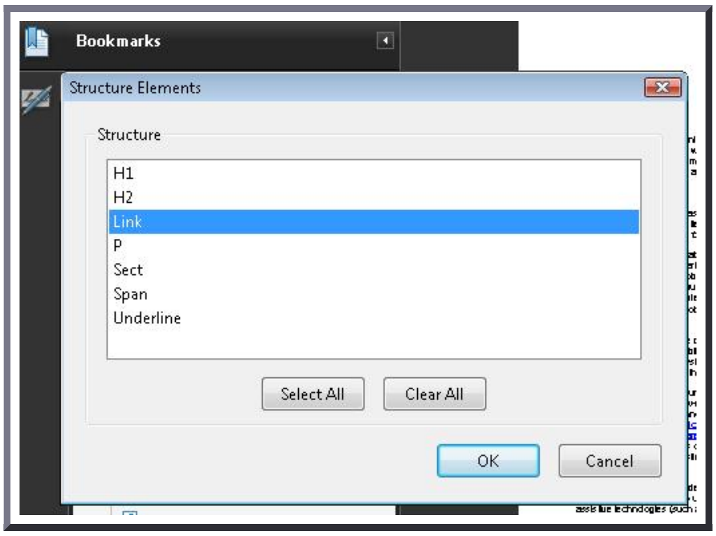
After conversion to tagged PDF, you may decide to add bookmarks that were not automatically generated. Like the converted bookmarks, tagged bookmarks use the underlying structural information in the document.

1. In the Bookmarks panel, choose the options menu, then choose New Bookmarks From Structure...
2. From the Structure Elements dialog, select the elements you want specified as tagged bookmarks.

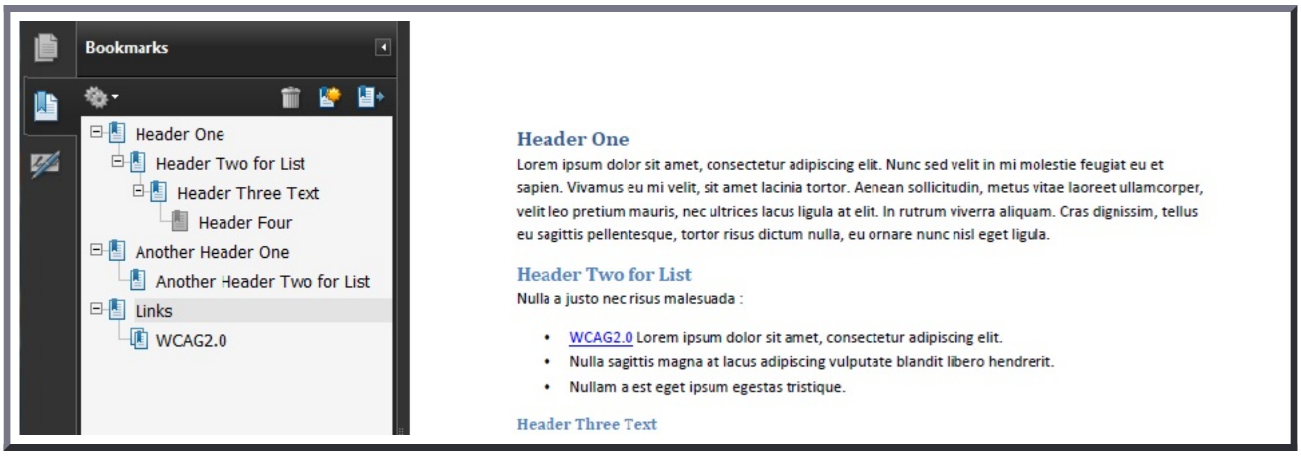
The image below shows the Bookmarks options menu.



The next image shows the selection of links in the document for bookmarking.



The tagged bookmarks are nested under a new, untitled bookmark. Access the context menu for the new bookmark and select the Rename option to rename the new bookmark, as shown in the following image.



Header One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc sed velit in mi molestie feugiat eu et sapien. Vivamus eu mi velit, sit amet lacinia tortor. Aenean sollicitudin, metus vitae laoreet ullamcorper, velit leo pretium mauris, nec ultrices lacus ligula at elit. In rutrum viverra aliquam. Cras dignissim, tellus eu sagittis pellentesque, tortor risus dictum nulla, eu ornare nunc nisl eget ligula.

Header Two for List

Nulla a justo nec risus malesuada :

- WCAG2.0 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- Nulla sagittis magna at lacus adipiscing vulputate blandit libero hendrerit.
- Nullam a est eget ipsum egestas tristique.

Header Three Text

This example is shown in operation in the working example of creating bookmarks with Acrobat Pro.

Example 4: Creating bookmarks with the outline hierarchy

The following code fragment illustrates part of an outline hierarchy used to create bookmarks This is typically accomplished by an authoring tool.

```
121 0 obj
 << /Type /Outlines
    /First 22 0 R
    /Last 29 0 R
    /Count 6
 >>
endobj
22 0 obj
 << /Title (Applying Guerrilla Tactics to Usability Testing by People with Disabilities)
    /Parent 21 0 R
    /Next 29 0 R
    /First 25 0 R
    /Last 28 0 R
    /Count 4
    /Dest [3 0 R /XYZ 0 792 0]
 >>
endobj
25 0 obj
 << /Title (Getting started)
    /Parent 22 0 R
    /Next 26 0 R
    /Dest [3 0 R /XYZ null 701 null]
 >>
endobj
```

Resources

Resources are for information purposes only, no endorsement implied.

- Section 12.3.3 (Document Outline) in PDF 1.7 (ISO 32000-1)
- PDF and Accessibility

Related Techniques

- G64: Providing a Table of Contents

Tests

Procedure

1. Check that the Bookmarks panel displays bookmarks.
2. Check that the bookmarks link to the correct sections in the document.

Expected Results

- Check #1 and Check #2 are true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF3: Ensuring correct tab and reading order in PDF documents

Applicability

Tagged PDF documents

This technique relates to:

- Success Criterion 1.3.2 (Meaningful Sequence)
  - How to Meet 1.3.2 (Meaningful Sequence)
  - Understanding Success Criterion 1.3.2 (Meaningful Sequence)
- Success Criterion 2.1.1 (Keyboard)
  - How to Meet 2.1.1 (Keyboard)
  - Understanding Success Criterion 2.1.1 (Keyboard)
- Success Criterion 2.1.3 (Keyboard (No Exception))
  - How to Meet 2.1.3 (Keyboard (No Exception))
  - Understanding Success Criterion 2.1.3 (Keyboard (No Exception))
- Success Criterion 2.4.3 (Focus Order)
  - How to Meet 2.4.3 (Focus Order)
  - Understanding Success Criterion 2.4.3 (Focus Order)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF3. Also see PDF Technology Notes.

Description

The intent of this technique is to ensure that users can navigate through content in a logical order that is consistent with the meaning of the content. Correct tab and reading order is typically accomplished using a tool for authoring PDF.

For sighted users, the logical order of PDF content is also the visual order on the screen. For keyboard and assistive technology users, the tab order through content, including interactive elements (form fields and links), determines the order in which these users can navigate the content. The tab order must reflect the logical order of the document.

Logical structure is created when a document is saved as tagged PDF. The reading order of a PDF document is the tag order of document elements, including interactive elements.

If the reading order is not correct, keyboard and assistive technology users may not be able to understand the content. For example, some documents use multiple columns, and the reading order is clear visually to sighted users as flowing from the top to the bottom of the first column, then to the top of the next column. But if the document is not properly tagged, a screen reader may read the document from top to bottom, across both columns, interpreting them as one column.

The simplest way to ensure correct reading order is to structure the document correctly in the authoring tool used to create the document, before conversion to tagged PDF. Note, however, that pages with complex layouts with graphics, tables, footnotes, side-bars, form fields, and other elements may not convert to tagged PDF in the correct reading order. These inconsistencies must then be corrected with repair tools such as Acrobat Pro.

When a PDF document containing form fields has a correct reading order, all form fields are contained in the tab order in the appropriate order, and in the correct order relative to other content in the PDF. Common tab-order errors include:

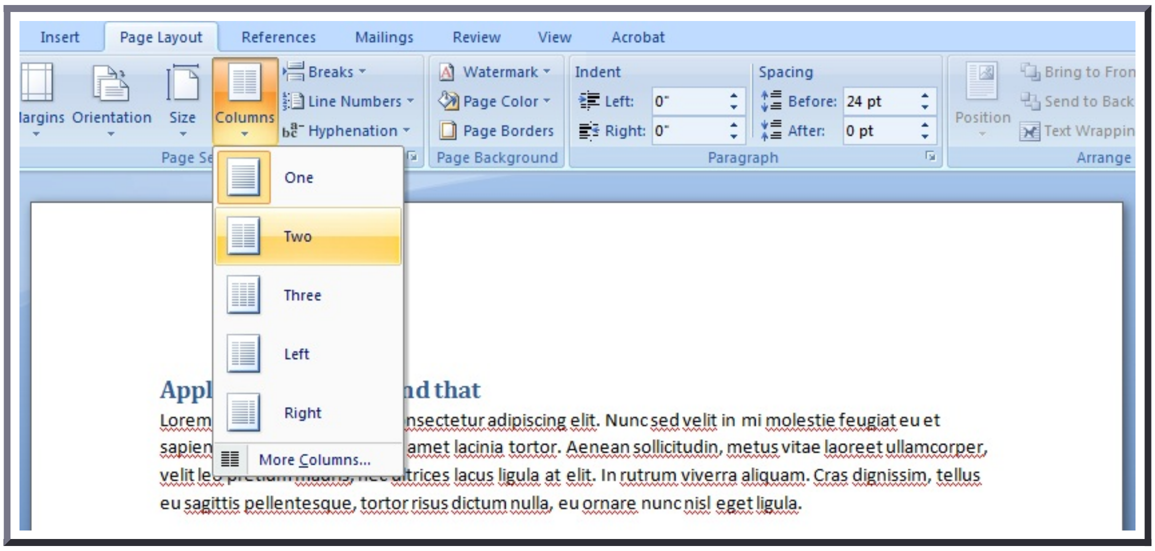
- Form fields missing from the tagged content.
- Form fields in the wrong location in the PDF content; e.g., at the end of non-interactive content.

Examples

Example 1: Creating a 2-column document using Microsoft Word 2007

This example is shown with Microsoft Word. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

Multi-column documents created using Word's Page Layout > Columns... tool typically are in the correct reading order when converted to tagged PDF. The image below shows Word's Columns tool.

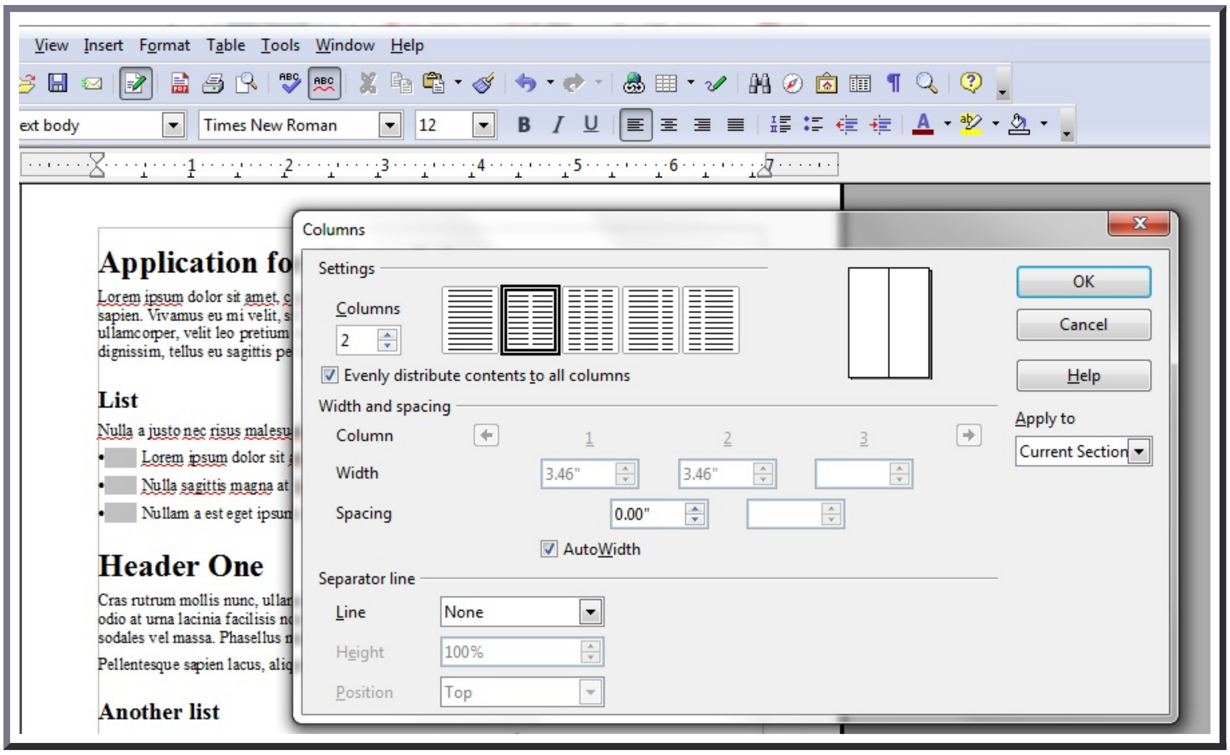


This example is shown in operation in the working example of 2-column document using Word 2007 (Word file) and working example of 2-column document using Word 2007 (PDF file).

Example 2: Creating a 2-column document using OpenOffice.org Writer 2.2

This example is shown with OpenOffice.org Writer. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

Multi-column documents created using OpenOffice.org Writer's Format > Columns... tool typically are in the correct reading order when converted to tagged PDF. The image below shows Writer's Columns tool.



Columns: 2
☑ Evenly distribute contents to all columns
Width: 3.46" 3.46"
Spacing: 0.00"
☑ AutoWidth
Line: None
Height: 100%
Position: Top
Apply to: Current Section

This example is shown in operation in the [working example of 2-column document using OpenOffice Writer (OpenOffice file)](#) and [working example of 2-column document using OpenOffice Writer (PDF file)](#).

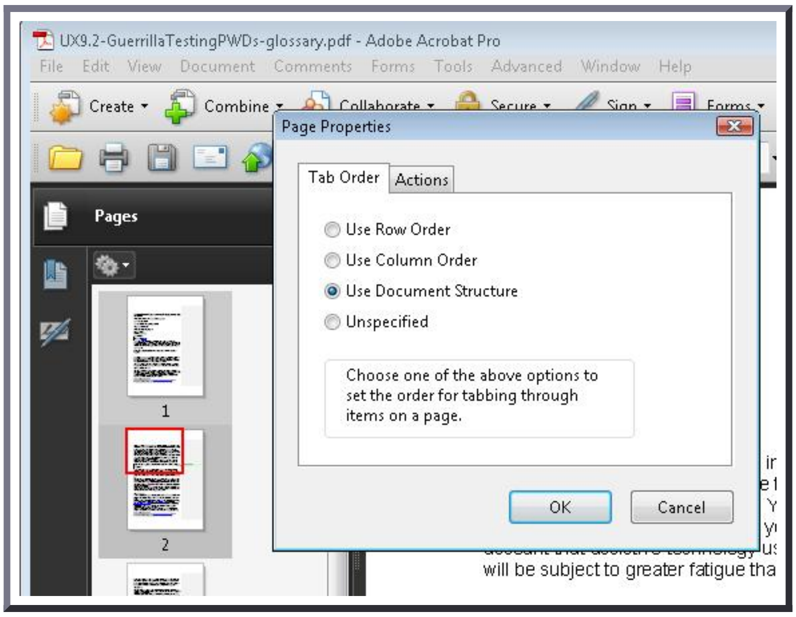
Example 3: Setting the tab order for one or more pages using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in [PDF Authoring Tools that Provide Accessibility Support](#).

In a tagged PDF document:

1. Open the Pages panel by either:
   - Clicking the Pages icon
   - Or selecting View > Navigation Panels > Pages
2. Select one or more page thumbnails.
3. Access the context menu for the selected thumbnail(s) and select Page Properties...
4. Select the Tab Order tab in the Page Properties dialog.
5. If needed, select a tab order option:

Option	Description
Use Row Order	Tabs from the upper left field, moving first left to right and then down, one table row at a time.
Use Column Order	Tabs from the upper left field, moving first from top to bottom and then across from left to right, one table column at a time.
Use Document Structure	For tagged documents, moves in the tag order specified by the authoring application. Note: This is usually the correct reading order and will be selected by default for tagged documents.
Unspecified	If the document was created using an earlier version of Acrobat Pro, the tab order is Unspecified by default. With this setting, form fields are tabbed through first, followed by links and then comments ordered by row. This may not be correct reading order.


UX9.2-GuerrillaTestingPWDs-glossary.pdf - Adobe Acrobat Pro
File Edit View Document Comments Forms Tools Advanced Window Help
Create Combine Collaborate Secure Sign Forms
Pages
Page Properties
Tab Order Actions
OK Cancel

Use Row Order
Use Column Order
[✓ Use Document Structure]
Unspecified

Choose one of the above options to set the order for tabbing through items on a page.

This example is shown in operation in the [working example of setting the tab order (Word file)]() and [working example of setting the tab order (PDF file)]().

Example 4: Checking the reading order using Reflow in Adobe Acrobat 9 Pro

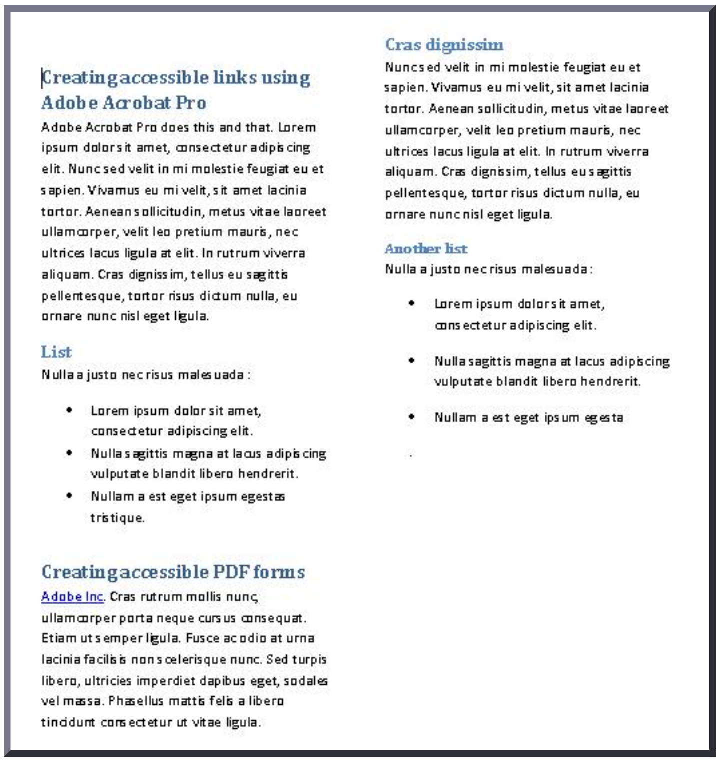
This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in [PDF Authoring Tools that Provide Accessibility Support]().

To quickly check the reading order of your PDF document, use the Reflow view in Adobe Acrobat Pro as follows:

- View > Zoom > Reflow

If the tagged PDF does not reflow in the correct reading order, you can use the authoring tool to repair reading order problems and re-convert the document to tagged PDF. If you do not have access to the original document and authoring tool, you can use the Order panel in the TouchUp Reading Order tool to resolve reading order problems (see Examples 3 and 4).

The following image shows a 2-column document converted to tagged PDF.



Creating accessible links using Adobe Acrobat Pro

Adobe Acrobat Pro does this and that. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc sed velit in mi molestie feugiat eu et sapien. Vivamus eu mi velit, sit amet lacinia tortor. Aenean sollicitudin, metus vitae laoreet ullamcorper, velit leo pretium mauris, nec ultrices lacus ligula at elit. In rutrum viverra aliquam. Cras dignissim, tellus eu sagittis pellentesque, tortor risus dictum nulla, eu ornare nunc nisl eget ligula.

List

Nulla a justo nec risus malesuada:

- Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- Nulla sagittis magna at lacus adipiscing vulputate blandit libero hendrerit.
- Nullam a est eget ipsum egestas tristique.

Creating accessible PDF forms

Adobe Inc. Cras rutrum mollis nunc, ullamcorper porta neque cursus consequat. Etiam ut semper ligula. Fusce ac odio at urna lacinia facilisis non scelerisque nunc. Sed turpis libero, ultricies imperdiet dapibus eget, sodales vel massa. Phasellus mattis felis a libero tincidunt consectetur ut vitae ligula.

Cras dignissim

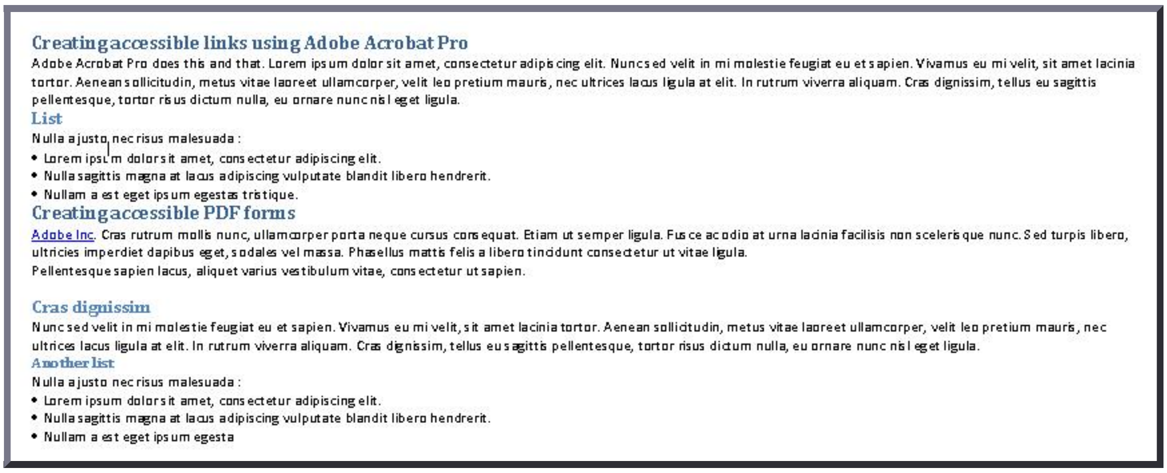
Nunc sed velit in mi molestie feugiat eu et sapien. Vivamus eu mi velit, sit amet lacinia tortor. Aenean sollicitudin, metus vitae laoreet ullamcorper, velit leo pretium mauris, nec ultrices lacus ligula at elit. In rutrum viverra aliquam. Cras dignissim, tellus eu sagittis pellentesque, tortor risus dictum nulla, eu ornare nunc nisl eget ligula.

Another list

Nulla a justo nec risus malesuada:

- Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- Nulla sagittis magna at lacus adipiscing vulputate blandit libero hendrerit.
- Nullam a est eget ipsum egesta

The following image shows the same 2-column document in reflow view.



Creating accessible links using Adobe Acrobat Pro

Adobe Acrobat Pro does this and that. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc sed velit in mi molestie feugiat eu et sapien. Vivamus eu mi velit, sit amet lacinia tortor. Aenean sollicitudin, metus vitae laoreet ullamcorper, velit leo pretium mauris, nec ultrices lacus ligula at elit. In rutrum viverra aliquam. Cras dignissim, tellus eu sagittis pellentesque, tortor risus dictum nulla, eu ornare nunc nisl eget ligula.

List

Nulla a justo nec risus malesuada :

- Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- Nulla sagittis magna at lacus adipiscing vulputate blandit libero hendrerit.
- Nullam a est eget ipsum egestas tristique.

Creating accessible PDF forms

Adobe Inc. Cras rutrum mollis nunc, ullamcorper porta neque cursus consequat. Etiam ut semper ligula. Fusce ac odio at urna lacinia facilisis non scelerisque nunc. Sed turpis libero, ultricies imperdiet dapibus eget, sodales vel massa. Phasellus mattis felis a libero tincidunt consectetur ut vitae ligula.
Pellentesque sapien lacus, aliquet varius vestibulum vitae, consectetur ut sapien.

Cras dignissim

Nunc sed velit in mi molestie feugiat eu et sapien. Vivamus eu mi velit, sit amet lacinia tortor. Aenean sollicitudin, metus vitae laoreet ullamcorper, velit leo pretium mauris, nec ultrices lacus ligula at elit. In rutrum viverra aliquam. Cras dignissim, tellus eu sagittis pellentesque, tortor risus dictum nulla, eu ornare nunc nisl eget ligula.

Another list

Nulla a justo nec risus malesuada :

- Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- Nulla sagittis magna at lacus adipiscing vulputate blandit libero hendrerit.
- Nullam a est eget ipsum egesta

Example 5: Checking the reading order using the Order panel in Adobe Acrobat 9 Pro

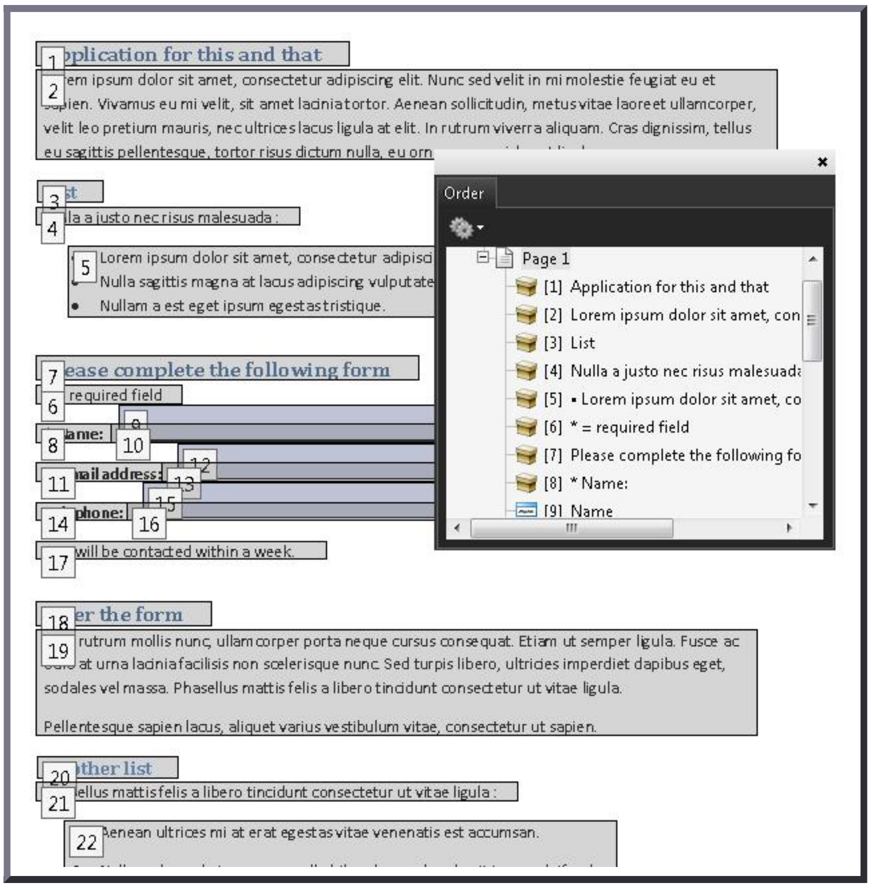
This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in [PDF Authoring Tools that Provide Accessibility Support](#).

To display the reading order of a document:

1. Advanced > Accessibility > Touch Up Reading Order...
2. Select the Show Order Panel button

In the TouchUp Reading Order tool, make sure the Show Page Content Order check box is checked. Each section of contiguous page content appears as a separate highlighted region and is numbered according to its placement in the reading order.

Note in the following image, the required-field text precedes the header in the reading order: it should follow the header. These are items 6 and 7 in the content order. Example 6 shows how to repair the order.



1 pplication for this and that
2 em ipsum dolor sit amet, consectetur adipiscing elit. Nunc sed velit in mi molestie feugiat eu et pien. Vivamus eu mi velit, sit amet lacinia tortor. Aenean sollicitudin, metus vitae laoreet ullamcorper, velit leo pretium mauris, nec ultrices lacus ligula at elit. In rutrum viverra aliquam. Cras dignissim, tellus eu sagittis pellentesque, tortor risus dictum nulla, eu orn
3 st
4 la a justo nec risus malesuada :
5 Lorem ipsum dolor sit amet, consectetur adipisci
Nulla sagittis magna at lacus adipiscing vulputate
Nullam a est eget ipsum egestas tristique.
7 ease complete the following form
6 required field
8 ame:
10
11 ail address:
13
14 phone:
16
17 will be contacted within a week.
18 er the form
19 rutrum mollis nunc, ullamcorper porta neque cursus consequat. Etiam ut semper ligula. Fusce ac at urna lacinia facilisis non scelerisque nunc. Sed turpis libero, ultricies imperdiet dapibus eget, sodales vel massa. Phasellus mattis felis a libero tincidunt consectetur ut vitae ligula.
Pellentesque sapien lacus, aliquet varius vestibulum vitae, consectetur ut sapien.
20 ther list
21 ellus mattis felis a libero tincidunt consectetur ut vitae ligula :
22 Aenean ultrices mi at erat egestas vitae venenatis est accumsan.

Order

Page 1
- [1] Application for this and that
- [2] Lorem ipsum dolor sit amet, con
- [3] List
- [4] Nulla a justo nec risus malesuad
- [5] • Lorem ipsum dolor sit amet, co
- [6] * = required field
- [7] Please complete the following fo
- [8] * Name:
- [9] Name

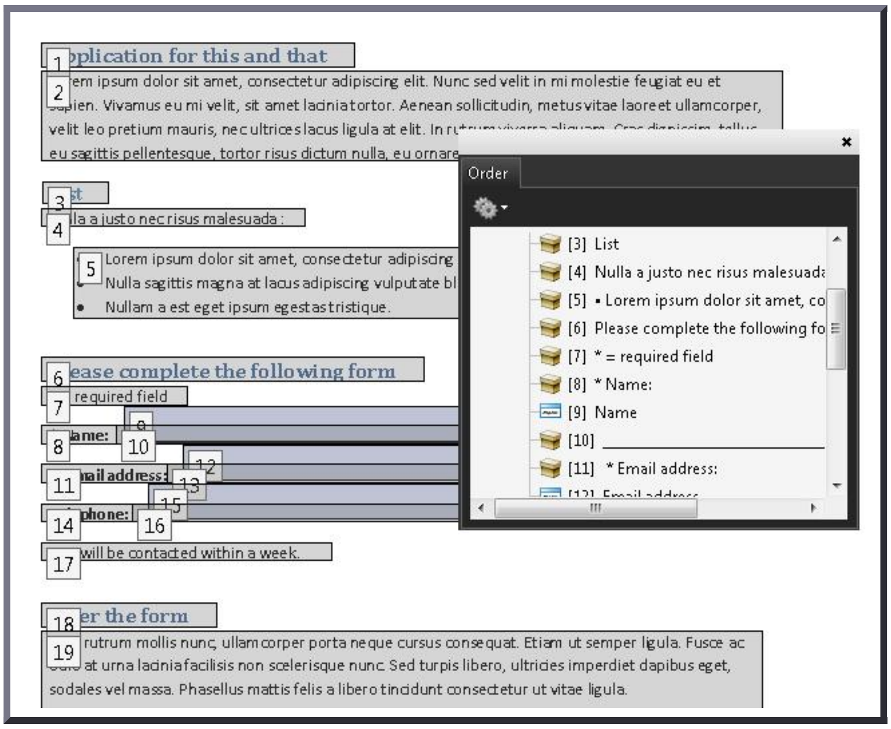
Example 6: Repairing the reading order using the Order panel in Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in [PDF Authoring Tools that Provide Accessibility Support](#).

To correct the reading order in Example 5, use the Order panel:

1. Either:
   - Drag-and-drop the header to precede the required-field text, or
   - Cut-and-paste the text to follow the header.

In the following image, the reading order is correct for the text and header. That is, the content elements numbered 6 and 7 have been switched into the correct reading order.



1 pplication for this and that
2 rem ipsum dolor sit amet, consectetur adipiscing elit. Nunc sed velit in mi molestie feugiat eu et
sapien. Vivamus eu mi velit, sit amet lacinia tortor. Aenean sollicitudin, metus vitae laoreet ullamcorper,
velit leo pretium mauris, nec ultrices lacus ligula at elit. In ru
eu sagittis pellentesque, tortor risus dictum nulla, eu ornare

3 st
4 lla a justo nec risus malesuada :
5 Lorem ipsum dolor sit amet, consectetur adipiscing
Nulla sagittis magna at lacus adipiscing vulputate bl
• Nullam a est eget ipsum egestas tristique.

6 ease complete the following form
7 required field
8 ame: 9 10
11 mail address: 12 13
14 phone: 15 16
17 will be contacted within a week.

18 er the form
19 rutrum mollis nunc, ullamcorper porta neque cursus consequat. Etiam ut semper ligula. Fusce ac
odio at urna lacinia facilisis non scelerisque nunc. Sed turpis libero, ultricies imperdiet dapibus eget,
sodales vel massa. Phasellus mattis felis a libero tincidunt consectetur ut vitae ligula.

Order

[3] List
[4] Nulla a justo nec risus malesuada
[5] • Lorem ipsum dolor sit amet, co
[6] Please complete the following fo
[7] * = required field
[8] * Name:
[9] Name
[10]
[11] * Email address:
[12] Email address

After this correction all content is in the correct reading order: the text preceding the form, the form fields, and the text following the form. Users can navigate from the text into and out of the form fields, in the correct order.

Note: Reordering content with the Order panel is most appropriate for simple text content within a PDF since modifications made with the order panel can affect not only the reading order but the underlying structure of content contained within the PDF. This may impact the z-order for content on a page, including making some content become hidden behind other content. Authors should save their work before using the order panel and verify that the changes do not have adverse effects on the document.

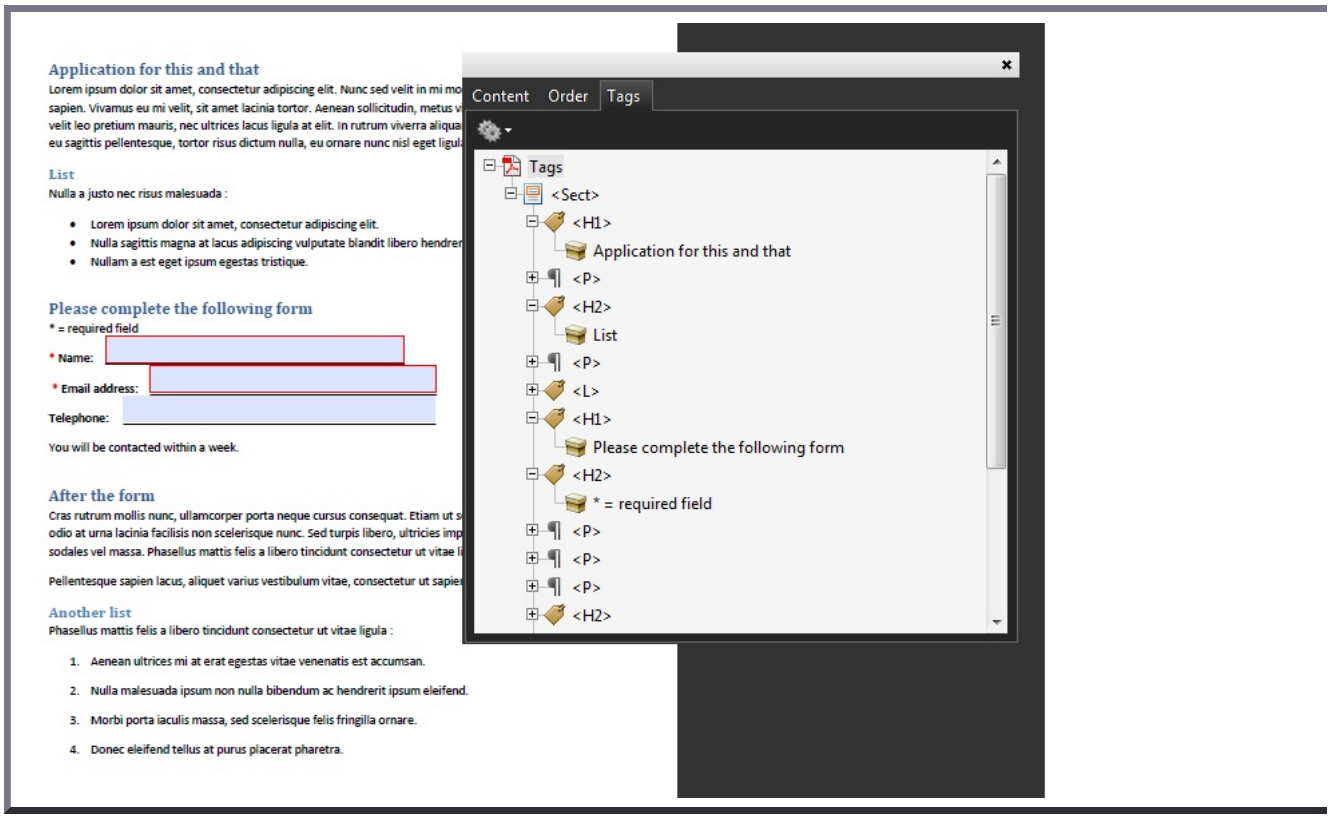
Example 7: Repairing the reading order using the Tags panel in Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support (http://trace.wisc.edu/wcag_wiki/index.php?title=PDF_Technology_Notes).

To correct the reading order in Example 5, use the Tags panel, and either

- Drag-and-drop the H1 tag to precede the required-field text (tagged H2), or
- Cut-and-paste the H2 tag to follow the H1 tag.

In the following image, the reading order is correct for the text and header. That is, the content elements H1 and H2 have been switched into the correct reading order.



Application for this and that
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc sed velit in mi mo
sapien. Vivamus eu mi velit, sit amet lacinia tortor. Aenean sollicitudin, metus v
velit leo pretium mauris, nec ultrices lacus ligula at elit. In rutrum viverra aliqua
eu sagittis pellentesque, tortor risus dictum nulla, eu ornare nunc nisl eget ligul

List
Nulla a justo nec risus malesuada :

- Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- Nulla sagittis magna at lacus adipiscing vulputate blandit libero hendrer
- Nullam a est eget ipsum egestas tristique.

Please complete the following form
* = required field
* Name:
* Email address:
Telephone:
You will be contacted within a week.

After the form
Cras rutrum mollis nunc, ullamcorper porta neque cursus consequat. Etiam ut s
odio at urna lacinia facilisis non scelerisque nunc. Sed turpis libero, ultricies imp
sodales vel massa. Phasellus mattis felis a libero tincidunt consectetur ut vitae l

Pellentesque sapien lacus, aliquet varius vestibulum vitae, consectetur ut sapie

Another list
Phasellus mattis felis a libero tincidunt consectetur ut vitae ligula :

1. Aenean ultrices mi at erat egestas vitae venenatis est accumsan.
2. Nulla malesuada ipsum non nulla bibendum ac hendrerit ipsum eleifend.
3. Morbi porta iaculis massa, sed scelerisque felis fringilla ornare.
4. Donec eleifend tellus at purus placerat pharetra.

Content Order Tags

```
Tags
  <Sect>
    <H1>
      Application for this and that
    <P>
    <H2>
      List
    <P>
    <L>
    <H1>
      Please complete the following form
    <H2>
      * = required field
    <P>
    <P>
    <P>
    <H2>
```

Resources

Resources are for information purposes only, no endorsement implied.

- Section 14.8 (Tagged PDF) in PDF 1.7 (ISO 32000-1)
- PDF and Accessibility
- Adobe Acrobat 9 Pro Help
- Making PDF documents accessible with Adobe Acrobat Pro

Related Techniques

- G57: Ordering the content in a meaningful sequence
- G59: Placing the interactive elements in an order that follows sequences and relationships within the content
- G202: Ensuring keyboard control for all functionality

Tests

Procedure

1. Verify that the content is in the correct reading order by one of the following:
   - Read the PDF document with a screen reader or a tool that reads aloud, listening to hear that each element is read in the correct order.
   - Reflow the pages and visually inspect the reading order.
   - Use a tool that exposes the document through the accessibility API, and verify that the reading order is correct.
2. Verify that the tab order is correct for focusable content by one of the following:
   - Use the tab key to traverse the focus order in the document.
   - Use a tool that is capable of showing the page object entry that specifies the tab order setting to open the PDF document and view the setting.

Expected Results

- #1 and Check #2 are true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF4: Hiding decorative images with the Artifact tag in PDF documents

Applicability

Tagged PDF documents

This technique relates to:

- Success Criterion 1.1.1 (Non-text Content)
  - How to Meet 1.1.1 (Non-text Content)
  - Understanding Success Criterion 1.1.1 (Non-text Content)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF4. Also see PDF Technology Notes.

Description

The purpose of this technique is to show how purely decorative images in PDF documents can be marked so that they can be ignored by Assistive Technology by using the /Artifact tag. This is typically accomplished by using a tool for authoring PDF.

In PDF, artifacts are generally graphics objects or other markings that are not part of the authored content. Examples of artifacts include page header or footer information, lines or other graphics separating sections of the page, or decorative images.

Examples

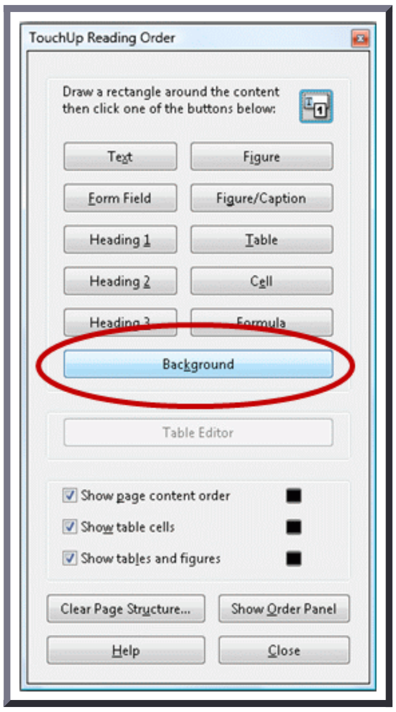
Example 1: Marking a background image as an artifact using Adobe Acrobat 9 Pro's TouchUp Reading Order Tool

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

The TouchUp Reading Order Tool can be used to mark an image as "Background," which removes it from the document tag structure.

1. Open the TouchUp Reading Order Tool in Acrobat Pro: Advanced > Accessibility > TouchUp Reading Order
2. Select the decorative image in the document
3. In the TouchUp Reading Order Tool, click the Background button to remove the selected image from the tag structure

The screenshot below illustrates this example.



This example is shown in operation in the working example of creating a decorative image (Word file) and working example of marking a background image as an artifact (PDF file).

Example 2: Marking an image as an artifact in a PDF document using an /Artifact tag or property list

The PDF specification allows images to be marked as "artifacts" as defined in Section 14.8.2.2 (Real Content and Artifacts) in PDF 1.7 (ISO 32000-1). An artifact is explicitly distinguished from real content by enclosing it in a marked-content sequence with the /Artifact tag.

/Artifact

```
BMC ... EMC
```

or

/Artifact propertyList

```
BDC ... EMC
```

The first is used to identify a generic artifact; the second is used for artifacts that have an associated property list. Note, to aid in text reflow, artifacts should be defined with property lists whenever possible. Artifacts lacking a specified bounding box are likely to be discarded during reflow.

Property list entries for artifacts include Type, BBox, Attached, and Subtype.

Resources

Resources are for information purposes only, no endorsement implied.

- Section 14.8.2.2 (Real Content and Artifacts) in PDF 1.7 (ISO 32000-1)
- PDF and Accessibility

Tests

Procedure

1. For an image that is purely decorative, use one of the following to verify that it is marked as an artifact:
   - Read the PDF document with a screen reader, listening to hear that the decorative image is not announced when reading the content line-by-line.
   - Using a PDF editor, make sure the decorative image is marked as an artifact.
   - Reflow the document and make sure the decorative image does not appear on the page.
   - Use a tool that is capable of showing the /Artifact entry or property list, such as aDesigner, to open the PDF document and verify that decorative images are marked as artifacts.
   - Use a tool that exposes the document through the accessibility API and verify that the decorative image is not exposed through the API.

Expected Results

- #1 is true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF5: Indicating required form controls in PDF forms

Applicability

Tagged PDF documents with forms

This technique relates to:

- Success Criterion 3.3.1 (Error Identification)
  - How to Meet 3.3.1 (Error Identification)
  - Understanding Success Criterion 3.3.1 (Error Identification)
- Success Criterion 3.3.2 (Labels or Instructions)
  - How to Meet 3.3.2 (Labels or Instructions)
  - Understanding Success Criterion 3.3.2 (Labels or Instructions)

  Note: This technique must be combined with other techniques to meet SC 3.3.2. See Understanding SC 3.3.2 for details.
- Success Criterion 3.3.3 (Error Suggestion)
  - How to Meet 3.3.3 (Error Suggestion)
  - Understanding Success Criterion 3.3.3 (Error Suggestion)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF5. Also see PDF Technology Notes.

Description

The objective of this technique is to notify the user when a field that must be completed has not been completed in a PDF form. Required fields are implemented using the /Ff entry in the form field's dictionary (see Table 220 in Section 12.7 (Interactive Forms) of PDF 1.7 (ISO 32000-1). This is normally accomplished using a tool for authoring PDF.

If errors are found, an alert dialog describes the nature of the error in text. This may be accomplished through scripting created by the author (see, for example, SCR18: Providing client-side validation and alert). User agents, such as Adobe Acrobat Pro and LiveCycle, can provide automatic alerts (as described in the examples below).

Note: once the user dismisses the alert dialog, it may be helpful if the script positions the keyboard focus on the field where the error occurred, although some users may expect the focus to remain on the last control focused prior to the alert appearing. Authors should exercise care to ensure that any movement of the focus will be expected. For example, if the alert announces a missing required phone number, positioning the focus on the phone number field when the alert is dismissed can be regarded as helpful and expected. In some cases, however, this may not be possible. If multiple input errors occur on the page, another approach must be taken to error reporting. (See, for example, the Adobe scripting resources.)

Ensuring that users are aware an error has occurred, can determine what is wrong, and can correct it are keys to software usability and accessibility. Meeting this objective helps ensure that all users can complete transactions with ease and confidence.

Labels for required form controls

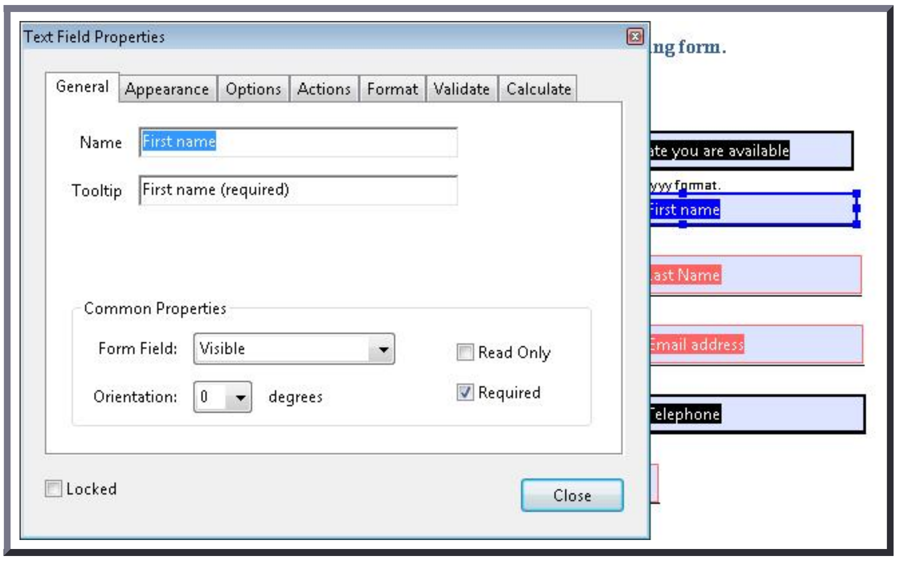
It is also important that users are aware that an error may occur. You can incorporate this information in labels; for example, "Date (required)" or the use of a red asterisk to indicate required fields. (Make sure that a legend appears on each form with required fields, e.g., "* = required field".) See PDF10: Providing labels for interactive form controls in PDF documents.

Examples

Example 1: Creating a required field in a PDF form using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. Access the context menu of the field and select the Properties dialog.
2. If the field is required, check the Required box. This checkbox forces the user to fill in the selected form field. If the user attempts to submit the form while a required field is blank, an error message appears and the empty required form field is highlighted.



Name: First name
Tooltip: First name (required)
Form Field: Visible
Orientation: 0 degrees
Read Only: unchecked
Required: checked

This example is shown in operation in the working example of creating a required field in Acrobat.

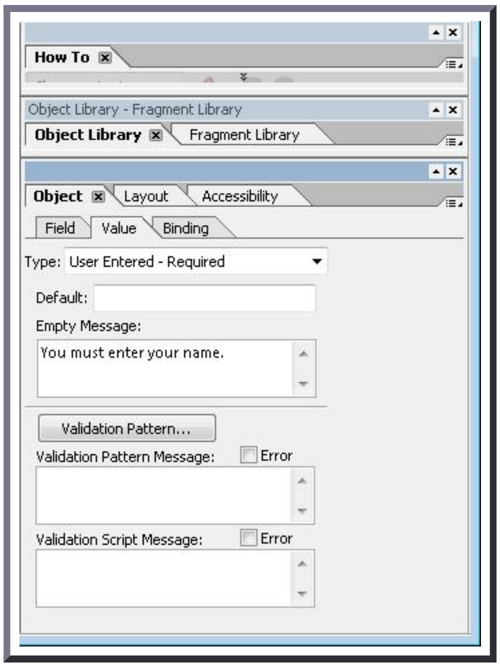
Example 2: Creating a required field in a PDF form using Adobe LiveCycle Designer ES 8.2.1

This example is shown with Adobe LiveCycle Designer. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. Access the context menu of the form control, select Palettes, and select Object.
2. Select "User entered - Required" from the Type pulldown.
3. Enter an error message in the "Empty Message" field. This message appears when a user tries to submit the form without entering a value in the required field.

If the user attempts to submit the form with a required field left blank, the Empty Message text appears and the empty required field is highlighted.

The image below shows the Adobe LiveCycle Object palette with the required selection.



Type: User Entered - Required
Default:
Empty Message: You must enter your name.

You can also add explicit text to the form label to indicate required fields (e.g., "(Required)").

This example is shown in operation in the working example of creating a required field in LiveCycle Designer.

Example 3: Adding a required text field in a PDF form using the /Tx field type and Ff flag

The following code fragment illustrates code that is typical for the object definitions for a typical text field. Note that the text field is required, using the Ff flag. This is typically accomplished by an authoring tool.

```
<< /AP -dict-
   /DA /Helv  0 Tf 0 g
   /DR -dict-
   /F 0x4
   /FT Tx              % FT key set to Tx for Text Field
   /Ff 0x2             % Ff integer 0x2 value indicates required
   /P -dict-
   /Rect -array-
   /StructParent 0x1
```

```
      /Subtype Widget
      /T First           % Partial field name First
      /TU First name (required)  % TU tool tip value serves as short description
      /Type Annot
      /V Pat Jones
   >>
   ...
   <Start Stream>
    BT
     /P <</MCID 0 >>BDC
     /CS0 cs 0  scn
     /TT0 1 Tf
       -0.001 Tc 0.003 Tw 11.04 0 0 11.04 72 709.56 Tm
       [(P)-6(1e)-3(as)10(e)-3( )11(P)-6(rin)2(t)-3( Y)8(o)-7(u)2(r N)4(a)11(m)-6(e)]TJ
     0 Tc 0 Tw 9.533 0 Td
     ( )Tj
     -0.004 Tc 0.004 Tw 0.217 0 Td
     [(\()-5(R)-4(e)5(q)-1(u)-1(i)-3(r)-3(e)-6(d)-1(\))]TJ
    EMC
     /P <</MCID 1 >>BDC
     0 Tc 0 Tw 4.283 0 Td
     [( )-2( )]TJ
      EMC
      /ArtifactSpan <</MCID 2 >>BDC
      0.002 Tc -0.002 Tw 0.456 0 Td
     [(__)11(___)11(___)11(___)11(___)11(_)11(____)11(___)11(___)11(_)]TJ
     0 Tc 0 Tw 13.391 0 Td
     ( )Tj
     EMC
    ET
   <End Stream>
```

Resources

Resources are for information purposes only, no endorsement implied.

- Section 12.7 (Interactive Forms) in PDF 1.7 (ISO 32000-1)
- Adobe XML Forms Architecture (XFA)
- PDF and Accessibility

Related Techniques

- G83: Providing text descriptions to identify required fields that were not completed
- H90: Indicating required form controls using label or legend
- SCR18: Providing client-side validation and alert
- PDF23: Providing interactive form controls in PDF documents
- PDF10: Providing labels for interactive form controls in PDF documents
- PDF22: Indicating when user input falls outside the required format or values in PDF forms

Tests

Procedure

For each form field that is required, verify that validation information and instructions are provided by applying the following:

1. Check that the required status is indicated in the form control's label.
2. Leave the field blank and submit the form. Check that an alert describing the error is provided.
3. Use a tool that exposes the document through the accessibility API, and verify that the required property is indicated.

Expected Results

- #1, #2, and #3 are true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF6: Using table elements for table markup in PDF Documents

Applicability

Tagged PDF documents with tables

This technique relates to:

- Success Criterion 1.3.1 (Info and Relationships)
  - How to Meet 1.3.1 (Info and Relationships)
  - Understanding Success Criterion 1.3.1 (Info and Relationships)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF6. Also see PDF Technology Notes.

Description

The purpose of this technique is to show how tables in PDF documents can be marked up so that they are recognized by assistive technology. This is typically accomplished by using a tool for authoring PDF.

Tabular information must be presented in a way that preserves relationships within the information even when users cannot see the table or the presentation format is changed. Information is considered tabular when logical relationships among text, numbers, images, or other data exist in two dimensions (vertical and horizontal). These relationships are represented in columns and rows, and the columns and rows must be recognizable in order for the logical relationships to be perceived.

Tagged tables can be created using the Add Tags to Document feature in Adobe Acrobat, using the Object Library in Adobe LiveCycle, or converting tables to PDF from a third-party application, such as Microsoft Word. However, the resulting tables may not be tagged correctly and you should ensure that table tagging issues are resolved.

Within PDF documents, a table uses the following structure types for table elements:

- A table element (Table).
- One or more table row elements(TR) which define each row of table cells as immediate children of the Table element.
- One or more table header elements (TH) or table data elements (TD) as the immediate children of each table row element.
- Cells that span two or more rows or columns should use the RowSpan or ColSpan attribute.
- For tables that contain blank cells, you may need to add empty TD cells so that each row or column has the same number of cells.

Examples

Example 1: Creating tables in Microsoft Word 2007 that have correctly tagged headings when converted to PDF

This example is shown with Microsoft Word. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. Access the table header row's context menu and select Table Properties...
2. Select the Row tab.
3. Check "Repeat as header at the top of each page" as shown in the following image.

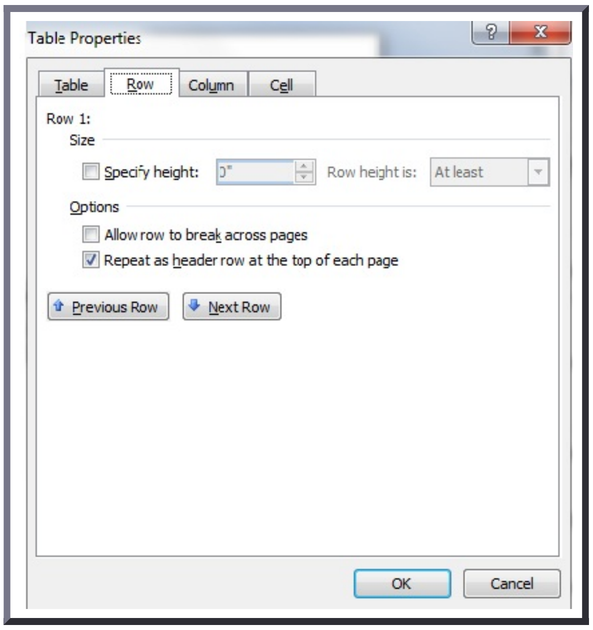


Table Properties

Row 1:

Size

☐ Specify height: 0"   Row height is: At least

Options

☐ Allow row to break across pages

☑ Repeat as header row at the top of each page

This example is shown in operation in the working example of tagged table headings in Word 2007.

Note: Microsoft Word can only mark up cells as column headings, not as row headings. Only the first row can be marked as heading for all table columns. When the table has row headings or a more complex heading structure, this mark-up must be added in a PDF editor such as Acrobat Pro.

Example 2: Creating tables in OpenOffice.org Writer 2.2 that have correctly tagged headings when converted to PDF

This example is shown with OpenOffice.org Writer. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. Access the table's context menu and select Table...
2. Select the Table Format tab.
3. Check Repeat Heading and select "1" in the First Rows listbox as shown in the following image.

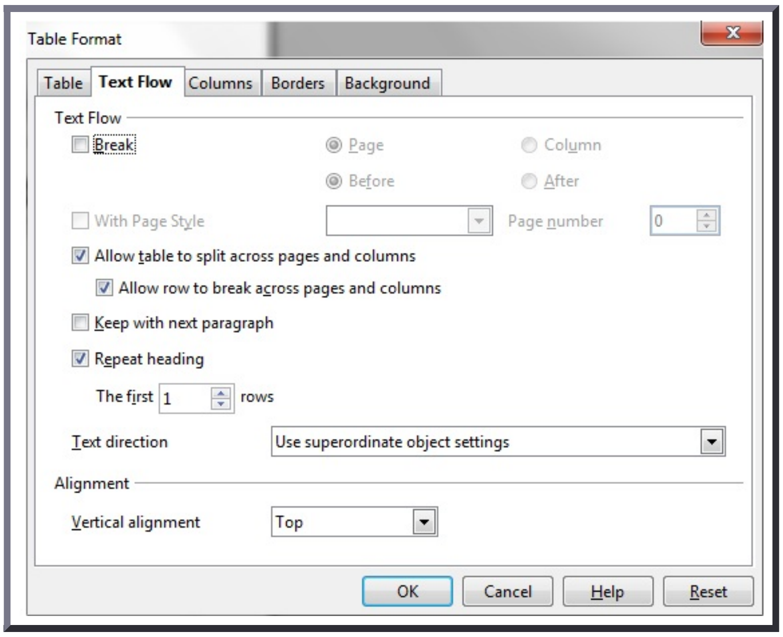


Table Format

Table | Text Flow | Columns | Borders | Background

Text Flow

- [ ] Break — Page / Column; Before / After
- [ ] With Page Style — Page number 0
- [✓] Allow table to split across pages and columns
  - [✓] Allow row to break across pages and columns
- [ ] Keep with next paragraph
- [✓] Repeat heading
  - The first 1 rows

Text direction: Use superordinate object settings

Alignment

Vertical alignment: Top

OK | Cancel | Help | Reset

This example is shown in operation in the working example of tagged table headings in OpenOffice Writer.

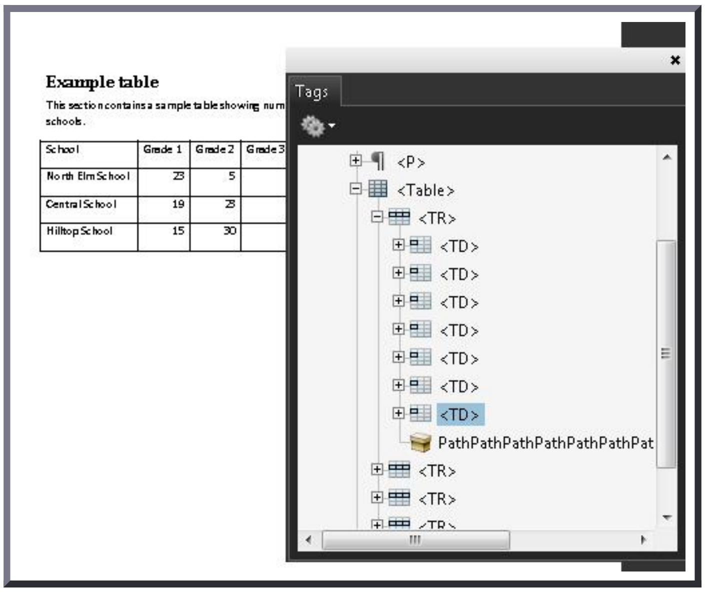
Note: OpenOffice.org Writer can only mark up cells as column headings, not as row headings. Only the first row can be marked as heading for all table columns. When the table has row headings or a more complex heading structure, this mark-up must be added in a PDF editor such as Acrobat Pro.

Example 3: Modifying table tags using the Tags tab in Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

To check that a converted document with tables has correct table tagging:

- In the View menu, select Navigation Panel, then select Tags.


Example table

This section contains a sample table showing num

schools.

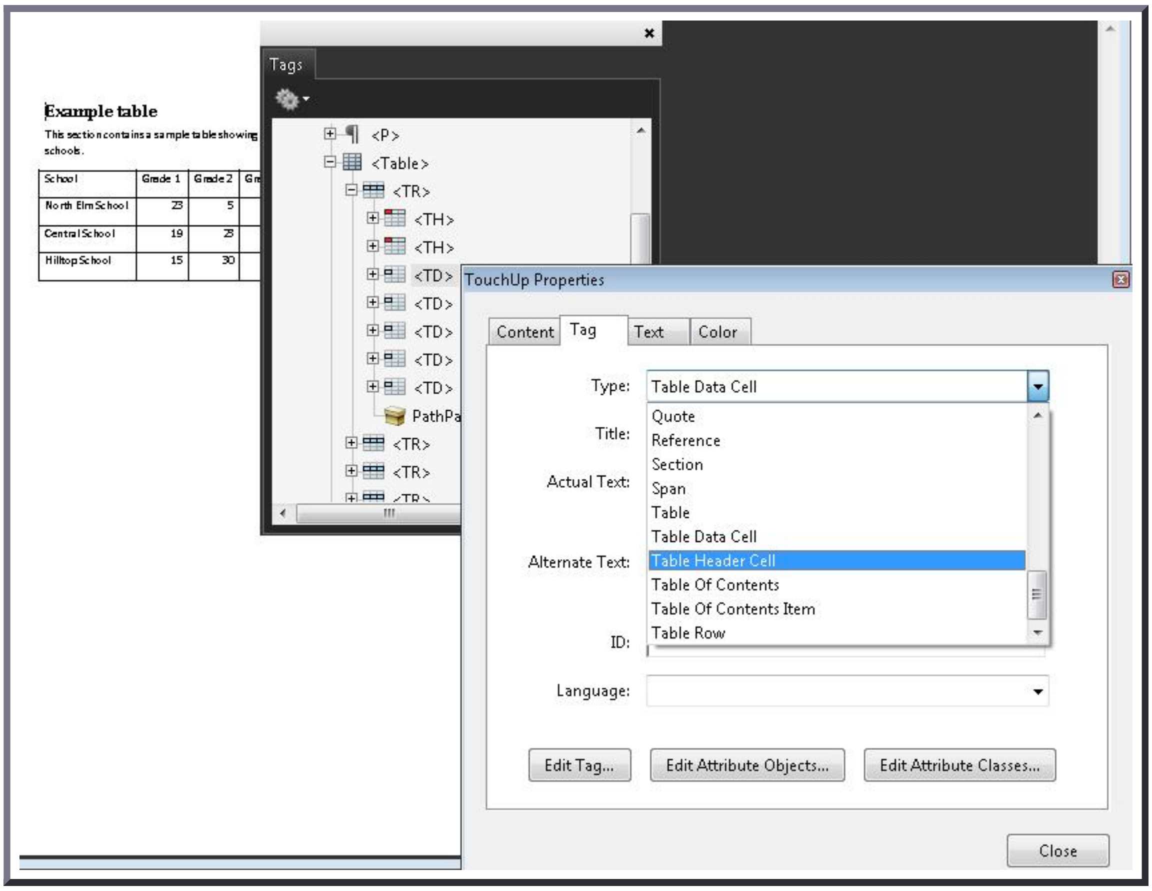
School	Grade 1	Grade 2	Grade 3
North Elm School	23	5	
Central School	19	23	
Hilltop School	15	30	

Tags

```
<P>
<Table>
  <TR>
    <TD>
    <TD>
    <TD>
    <TD>
    <TD>
    <TD>
    <TD>
      PathPathPathPathPathPathPat
  <TR>
  <TR>
  <TR>
```

Note that in this case, the table headers were not formatted as illustrated in Examples 1 and 2, and are marked as data cells (TD). To change these to TH tags:

1. On the Tags tab, open the table row that contains the header cells, as shown on the image above.
2. Select on the first data cell and select Properties...
3. On the Tags tab in the Properties dialog, use the Type dropdown to change Table Data Cell to Table Header Cell.
4. Repeat for all the table header cells in the first table row.



Example table

This section contains a sample table showing ... schools.

School	Grade 1	Grade 2	Gr
North Elm School	23	5	
Central School	19	23	
Hilltop School	15	30	

TouchUp Properties

Type: Table Data Cell

This example is shown in operation in the working example of tagged table headings in Acrobat.

Example 4: Marking up a table using table structure elements

The following code fragment illustrates code that is typical for a simple table (header row and data row) such as shown in Examples 1-3:

```
95 0 obj                %Structure element for a table
 <<
  /A 39 0 R
  /K[96 0 R 101 0 R 106 0 R 111 0 R]
  /P 93 0 R
  /S/Table              %standard structure type is table
 >>
 endobj
96 0 obj                %Structure element for a table row
 <<
  /K[97 0 R 98 0 R 99 0 R 100 0 R]
  /P 95 0 R
  /S/TR                 %standard structure type is table row
 >>
 endobj
97 0 obj                %Structure element for a table header
 <</A[23 0 R 120 0 R]
   /K 1
   /P 96 0 R
   /S/TH                 %standard structure type is table head
   /Pg 8 0 R
 >>
endobj
104 0 obj                %Structure element for table data (cell contents)
 <<
  /A 29 0 R
  /K 7
  /P 101 0 R
  /S/TD                  %standard structure type is table data
  /Pg 8 0 R
 >>
endobj
```

Resources

Resources are for information purposes only, no endorsement implied.

- Section 14.8.4.3.4 (Table Elements) in PDF 1.7 (ISO 32000-1)
- PDF and Accessibility

Related Techniques

- H51: Using table markup to present tabular information
- PDF20: Using Adobe Acrobat Pro's Table Editor to repair mistagged tables

Tests

Procedure

1. For each table, confirm one of the following:
   - Read the PDF document with a screen reader, listening to hear that the tabular information is presented in a way that preserves logical relationships among the table header and data cells.
   - Using a PDF editor, verify that the appropriate TR, TH, and TD tags are in the proper reading order and hierarchy in the table tree.
   - Use a tool which is capable of showing the table elements to open the PDF document, view the table structure, and verify that it contains the appropriate TR, TH, and TD structures.
   - Use a tool that exposes the document through the accessibility API, and verify that the table structure contains the appropriate TR, TH, and TD structures, and that they are in the proper reading order and hierarchy.

Expected Results

- #1 is true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF7: Performing OCR on a scanned PDF document to provide actual text

Applicability

Scanned PDF documents

This technique relates to:

- Success Criterion 1.4.5 (Images of Text)
  - How to Meet 1.4.5 (Images of Text)
  - Understanding Success Criterion 1.4.5 (Images of Text)
- Success Criterion 1.4.9 (Images of Text (No Exception))
  - How to Meet 1.4.9 (Images of Text (No Exception))
  - Understanding Success Criterion 1.4.9 (Images of Text (No Exception))

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF7. Also see PDF Technology Notes.

Description

The intent of this technique is to ensure that visually rendered text is presented in such a manner that it can be perceived without its visual presentation interfering with its readability.

A document that consists of scanned images of text is inherently inaccessible because the content of the document is images, not searchable text. Assistive technologies cannot read or extract the words; users cannot select, edit, resize, or reflow text nor can they change text and background colors; and authors cannot manipulate the PDF for accessibility.

For these reasons, authors should use actual text rather than images of text, using an authoring tool such as Microsoft Word or Oracle Open Office to author and convert content to PDF.

If authors do not have access to the source file and authoring tool, scanned images of text can be converted to PDF using optical character recognition (OCR). Adobe Acrobat Pro can then be used to create accessible text.

Examples

Example 1: Generating actual text rather than images of text using Adobe Acrobat 9 Pro

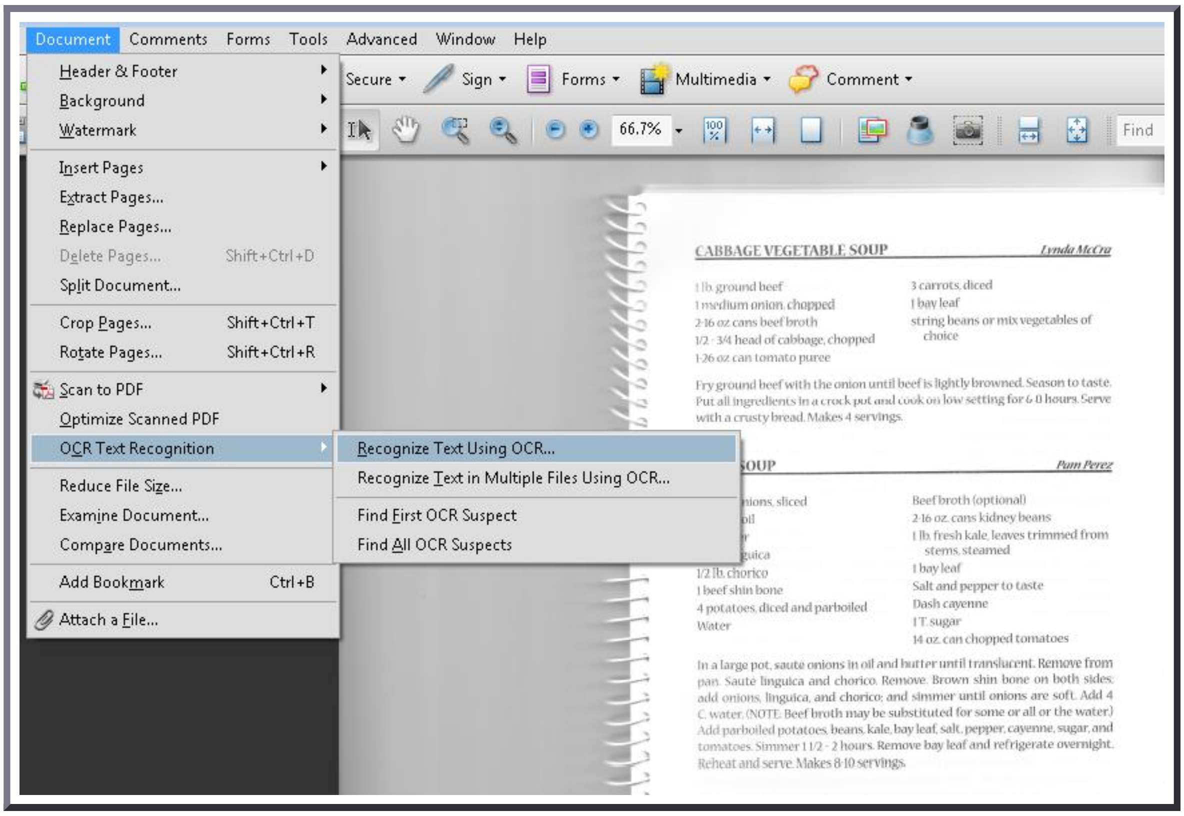
This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

This example uses a simple one-page scanned image of text. To ensure that actual text is stored in the document, perform the following steps:

1. Scan the document using as high a resolution as possible to improve the OCR performance.
2. Load the scanned document in Acrobat Acrobat Pro. Select Document > OCR Text Recognition > Recognize Text Using OCR...
3. In the next dialog, select the All Pages radio button under Pages (or Current Page if you are converting only one page), and then select OK.
4. Under the Settings list, select Edit. In the next dialog, select Formatted Text and Graphics in the PDF Output Style drop-down list. This is important for ensuring accessibility.
5. Depending on the resolution and how clear the text was, OCR converts images of words and characters to actual text. Text that Acrobat Pro does not recognize is listed as an "OCR suspect," or text element that Acrobat suspects was not recognized correctly.
6. To fix the suspects, choose Document > OCR Text Recognition > Find First OCR Suspect. Acrobat Pro presents each suspect one at a time, which can be corrected using Acrobat Pro touchup tools.
7. Run Advanced > Accessibility > Add Tags to Document
8. Test for accessibility: Advanced > Accessibility > Full Check...

Note: Alternatively, you can use Document > OCR Text Recognition > Find All OCR Suspects to display all OCR suspects at the same time for faster editing.

The following image shows a scanned one-page document in Adobe Acrobat Pro.



CABBAGE VEGETABLE SOUP *Lynda McCra*

1 lb. ground beef
1 medium onion, chopped
2-16 oz cans beef broth
1/2 - 3/4 head of cabbage, chopped
1-26 oz can tomato puree

3 carrots, diced
1 bay leaf
string beans or mix vegetables of choice

Fry ground beef with the onion until beef is lightly browned. Season to taste. Put all ingredients in a crock pot and cook on low setting for 6-8 hours. Serve with a crusty bread. Makes 4 servings.

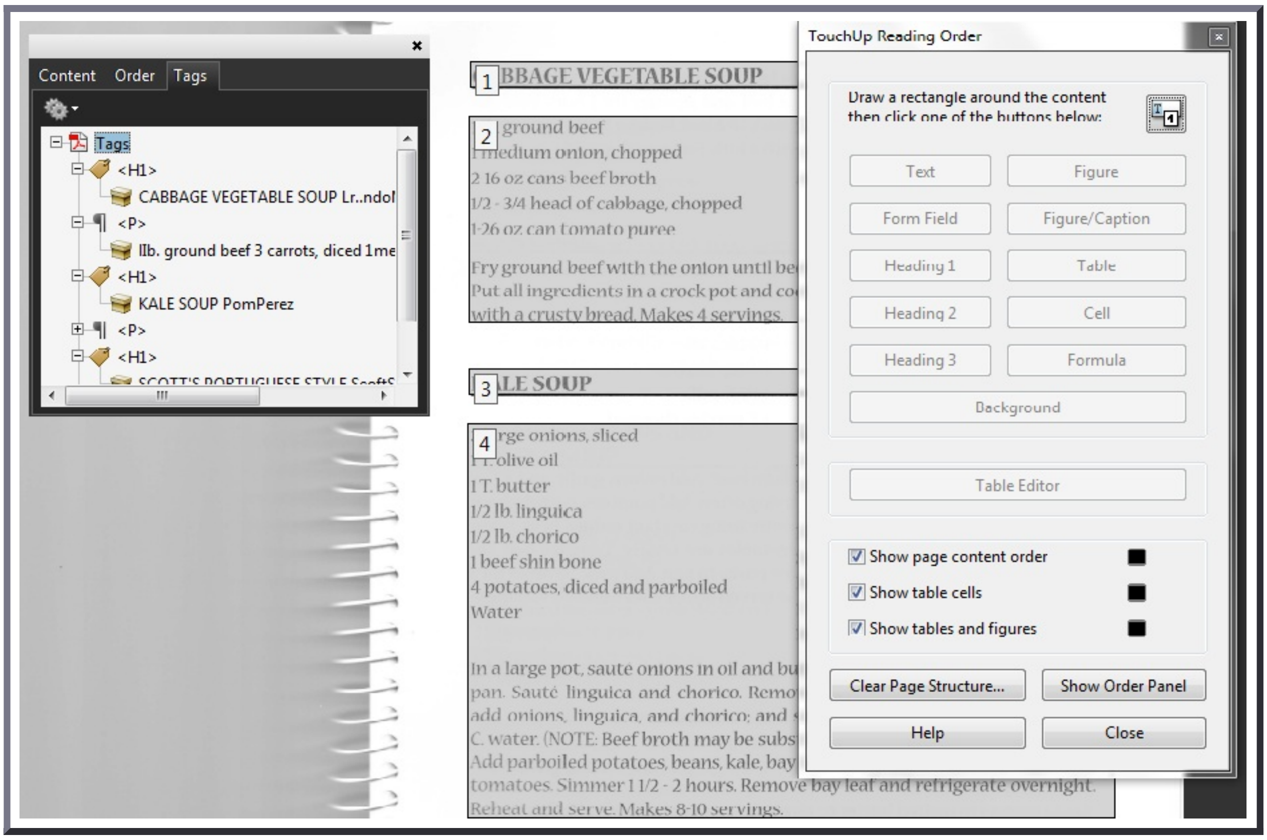
OUP *Pam Perez*

nions, sliced
oil
r
guica
1/2 lb. chorico
1 beef shin bone
4 potatoes, diced and parboiled
Water

Beef broth (optional)
2-16 oz. cans kidney beans
1 lb. fresh kale, leaves trimmed from stems, steamed
1 bay leaf
Salt and pepper to taste
Dash cayenne
1 T. sugar
14 oz. can chopped tomatoes

In a large pot, saute onions in oil and butter until translucent. Remove from pan. Sauté linguica and chorico. Remove. Brown shin bone on both sides; add onions, linguica, and chorico; and simmer until onions are soft. Add 4 C. water. (NOTE: Beef broth may be substituted for some or all or the water.) Add parboiled potatoes, beans, kale, bay leaf, salt, pepper, cayenne, sugar, and tomatoes. Simmer 1 1/2 - 2 hours. Remove bay leaf and refrigerate overnight. Reheat and serve. Makes 8-10 servings.

The next image shows the converted content after adding tags to the document. It will probably be necessary to use the TouchUp Reading Order tool and the Tags panel to tag the content properly for the intended final document. For this example, the image of the spiral book binding was tagged in the conversion. The TouchUp Reading Order tool was used to hide the image as a background (decorative) image (see PDF4: Hiding decorative images with the Artifact tag in PDF documents). The recipe titles were tagged as first level headers.



ground beef
medium onion, chopped
2 16 oz cans beef broth
1/2 - 3/4 head of cabbage, chopped
1-26 oz can tomato puree

Fry ground beef with the onion until be
Put all ingredients in a crock pot and co
with a crusty bread. Makes 4 servings.

rge onions, sliced
T. olive oil
1 T. butter
1/2 lb. linguica
1/2 lb. chorico
1 beef shin bone
4 potatoes, diced and parboiled
Water

In a large pot, saute onions in oil and bu
pan. Sauté linguica and chorico. Remo
add onions, linguica, and chorico; and
C. water. (NOTE: Beef broth may be subs
Add parboiled potatoes, beans, kale, bay
tomatoes. Simmer 1 1/2 - 2 hours. Remove bay leaf and refrigerate overnight.
Reheat and serve. Makes 8-10 servings.

Note: Acrobat Pro may automatically add tags when the file is run through OCR.

This example is shown in operation in the working example of generating actual text and the result of performing OCR.

Resources

Resources are for information purposes only, no endorsement implied.

- PDF and Accessibility

Related Techniques

- G140: Separating information and structure from presentation to enable different presentations

Tests

Procedure

1. For each page converted to text using OCR, ensure that the resulting PDF has been converted correctly, using one of the following ways:
   - Read the PDF document with a screen reader or a tool that reads aloud, listening to hear that all text is read correctly and in the correct reading order.
   - Save the document as text and check that the converted text is complete and in the correct reading order.
   - Use a tool that is capable of showing the converted content to open the PDF document and verify that all text was converted and is in the correct reading order.
   - Use a tool that exposes the document through the accessibility API and verify that all text was converted and is in the correct reading order.

Expected Results

- #1 is true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF8: Providing definitions for abbreviations via an E entry for a structure element

Applicability

Tagged PDF documents containing abbreviations or acronyms

This technique relates to:

- Success Criterion 3.1.4 (Abbreviations)
  - How to Meet 3.1.4 (Abbreviations)
  - Understanding Success Criterion 3.1.4 (Abbreviations)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF8. Also see PDF Technology Notes.

Description

The objective of this technique is to provide an expansion or definition of an abbreviation for the first occurrence of the abbreviation. For example, a reference to an abbreviation, such as "WCAG," should be available as "Web Content Accessibility Guidelines (WCAG)" on its first occurrence in a document.

This is done by setting expansion text using an /E entry for a structure element, and is normally accomplished using a tool for authoring PDF. A Span structure element is typically used to tag the abbreviation, but the /E entry is valid with any structure element.

This technique is applicable for any abbreviation, including acronyms and initialisms. Note that on the first occurrence of the abbreviation, both the abbreviation and the expansion text must be provided. This will aid recognition of later use of the abbreviation.

PDF documents may be enhanced by providing expansions for abbreviations. In fact, such expansions are required for accessibility to ensure understanding by people who have difficulty decoding words; rely on screen magnification (which may obscure context); have limited memory; or who have difficulty using context to aid understanding.

Examples

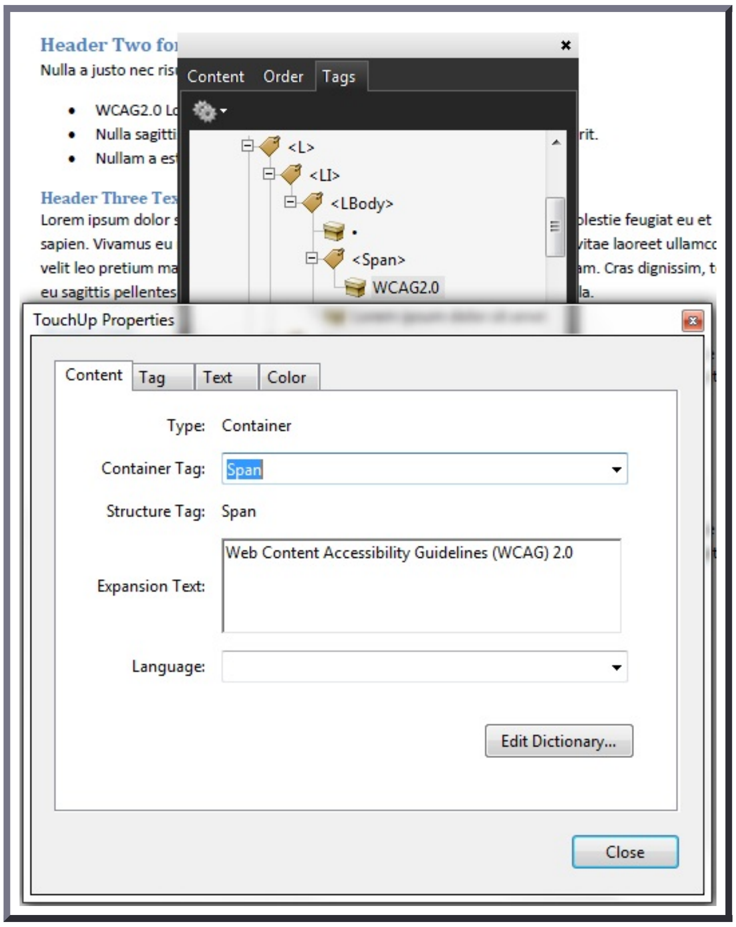
Example 1: Adding an /E entry to an abbreviation using Adobe Acrobat 9 Pro's Tags panel

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

In a tagged PDF document:

1. Select the Tags panel, using Views > Navigation Panes > Tags
2. Select the first instance of the abbreviated text that needs to be expanded. If the selected text is part of a larger tag, access the Tags panel options menu, select Create Tag from Selection, and create a new Span tag. In this example, the text "WCAG2" (within the LBody tag) has been enclosed in a Span tag.
3. In the Tags panel, access the context menu for the spanned text and select Properties... to open the TouchUp Properties dialog for the Span tag.
4. On the Content tab of the TouchUp Properties dialog, enter the expansion text, followed by the originally selected text.

The following image illustrates this technique:



TouchUp Properties

Content | Tag | Text | Color

Type: Container

Container Tag: Span

Structure Tag: Span

Expansion Text: Web Content Accessibility Guidelines (WCAG) 2.0

Language:

Edit Dictionary...

Close

This example is shown in operation in the working example of Providing definitions for Abbreviations (Word document), working example of Providing definitions for Abbreviations (OpenOffice document), and working example of Providing definitions for Abbreviations (PDF document).

Example 2: Using a /Span structure element with an /E entry to define an abbreviation

The following code fragment illustrates code that is typical for using the /Span structure element to define an abbreviation.

This example uses the sentence "Sugar is commonly sold in 5 lb bags." The abbreviation "lb" is tagged as a /Span structure element with an /E entry (typically accomplished by an authoring tool).

```
1 0 obj                                  % structure element
  << /Type /StructElemen
           /S /Span                      % element type
           /P ...                        % Parent in structure hierarchy
           /K << /Type /MCR
                       /Page 2 0 R       % Page containing marked-content sequence
                       /MCID 0           % Marked content identifier for "lb"
              >>
           /E  (pound, lb)
   >>
endobj
```

Example 3: Using a /TH structure element with an /E entry to define an abbreviation

As noted in the Description, the /E entry is valid with any structure element.

The following code fragment illustrates code that is typical for using an /E entry to define an abbreviation.

A table that contains columns for each month uses abbreviations as the values of column headers. The expansion for each abbreviation is provided as the /E entry of the /TH structure element (typically accomplished by an authoring tool).

```
1 0 obj                                  % structure element
  << /Type /StructElemen
           /S /TH                        % element type
           /P ...                        % Parent in structure hierarchy
           /K << /Type /MCR
                       /Page 2 0 R       % Page containing marked-content sequence
                       /MCID 0           % Marked content identifier for "Dec"
              >>
           /E  (December, Dec)
   >>
endobj
```

Resources

Resources are for information purposes only, no endorsement implied.

- Section 14.9.5 (Expansion of Abbreviations and Acronyms) in PDF 1.7 (ISO 32000-1)
- Add alternate text and supplementary information to tags
- Microsoft Inspect.exe tool
- PDF and Accessibility

Related Techniques

- G102: Providing the expansion or explanation of an abbreviation
- G55: Linking to definitions
- G62: Providing a glossary
- G70: Providing a function to search an online dictionary
- G97: Providing the first use of an abbreviation immediately before or after the expanded form

Tests

Procedure

1. Verify that the first occurrence of abbreviations that require expansion text have /E entries on an enclosing tag by one of the following and that both the abbreviation and the expansion text are provided:
   - In Windows, use Microsoft's Inspect.exe tool, or some other tool that allows inspection of the MSAA interface, to locate the text of the abbreviation in the document tree and ensure that the value of the abbreviation is in the expansion text.
   - In a PDF editor, locate the tag for the text that is the abbreviation, and check that an expansion or definition is provided for each abbreviation in the Expansion Text field in the corresponding tag's properties.
   - Read the PDF document with a screen reader, listening to hear that on the first occurrence, the abbreviation and expansion are read when the screen reader reads the content line-by-line.
   - Use a tool that is capable of showing the /E entry value, such as aDesigner to open the PDF document and view the GUI summary to read the text expansions for abbreviations.
   - Use a tool that exposes the document through the accessibility API and verify that the text expansion of the abbreviation is properly implemented.

Expected Results

- Check #1 is true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF9: Providing headings by marking content with heading tags in PDF documents

Applicability

Tagged PDF documents with headings

This technique relates to:

- Success Criterion 1.3.1 (Info and Relationships)
  - How to Meet 1.3.1 (Info and Relationships)
  - Understanding Success Criterion 1.3.1 (Info and Relationships)
- Success Criterion 2.4.1 (Bypass Blocks)
  - How to Meet 2.4.1 (Bypass Blocks)
  - Understanding Success Criterion 2.4.1 (Bypass Blocks)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF9. Also see PDF Technology Notes.

Description

The purpose of this technique is to show how headings in PDF documents can be marked so that they are recognized by assistive technologies. Headings are marked up using the heading elements (H, H1, H2, ... H6) in the structure tree. This is typically accomplished by using a tool for authoring PDF.

Heading markup can be used:

- to indicate start of main content
- to mark up section headings within the main content area
- to demarcate different navigational sections, such as top or main navigation, left or secondary navigation, and footer navigation
- to mark up images (containing text) which have the appearance of headings visually.

Because headings indicate the start of important sections of content, it is possible for assistive technology users to access the list of headings and to jump directly to the appropriate heading and begin reading the content. This ability to "skim" the content through the headings and go directly to content of interest significantly speeds interaction for users who would otherwise access the content slowly.

Examples

Example 1: Adding or modifying tagged headings in PDF documents with Adobe Acrobat 9 Pro

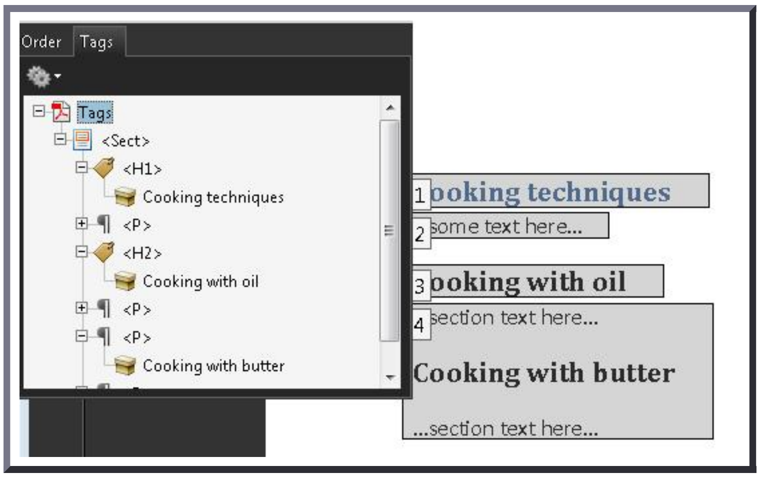
This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

Using the Touchup Reading Order tool

One method of adding headings to PDF documents uses the Touchup Reading Order tool:

1. Open the PDF document in Adobe Acrobat Pro
2. Select Advanced > Accessibility > TouchUp Reading Order...
3. Click the Show Order Panel button on the TouchUp Reading Order panel
4. View the tags in the Show Order panel.

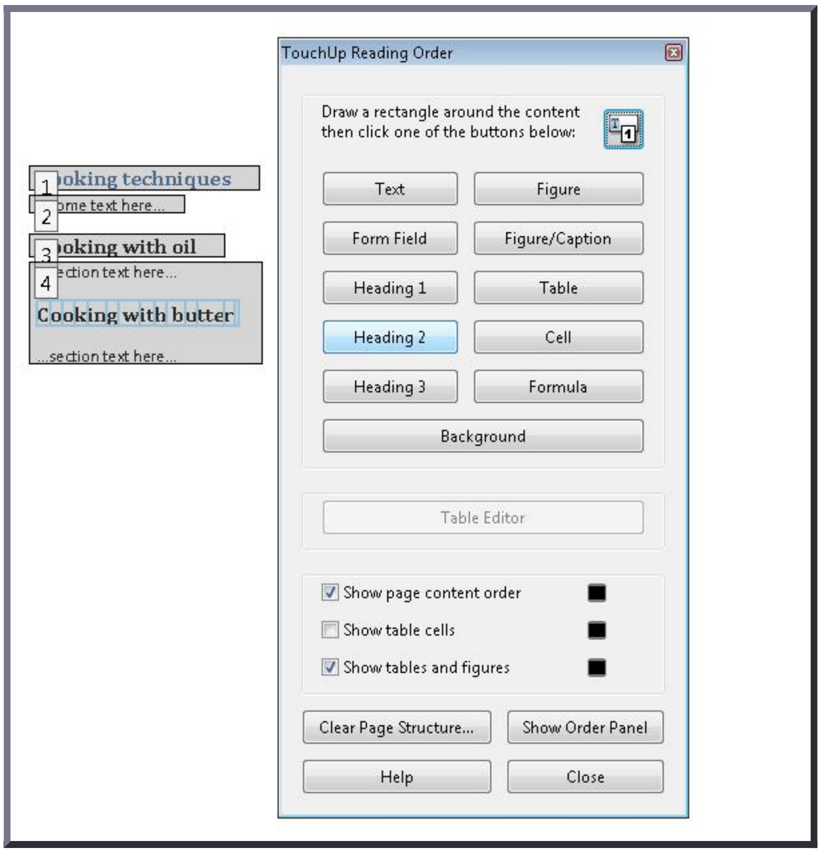
The following image shows a PDF document opened in Adobe Acrobat Pro. The Tags panel is open, showing heading text "Cooking techniques" tagged as H1 and "Cooking with oil" tagged as H2. The text "Cooking with butter" should be tagged as H2 but is not.



To correct the H2 heading, use the TouchUp Reading Order panel as follows:

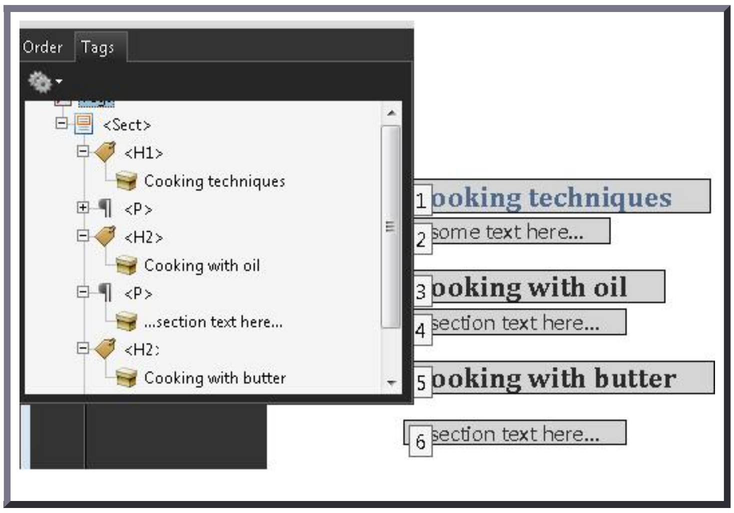
1. Left click and drag a selection box over the content you want to tag.
2. Select the Heading 2 tag from the TouchUp Reading Order panel.

The following image shows the PDF document opened in Adobe Acrobat Pro. The TouchUp Reading Order panel is visible. A selection box appears around the text "Cooking with butter," and Heading 2 on the panel is selected.



Finally, click the Show Order Panel button on the TouchUp Reading Order panel.

The following image shows the PDF document opened in Adobe Acrobat Pro. The Tags panel is visible, showing that the text "Cooking with butter" is now tagged as H2.

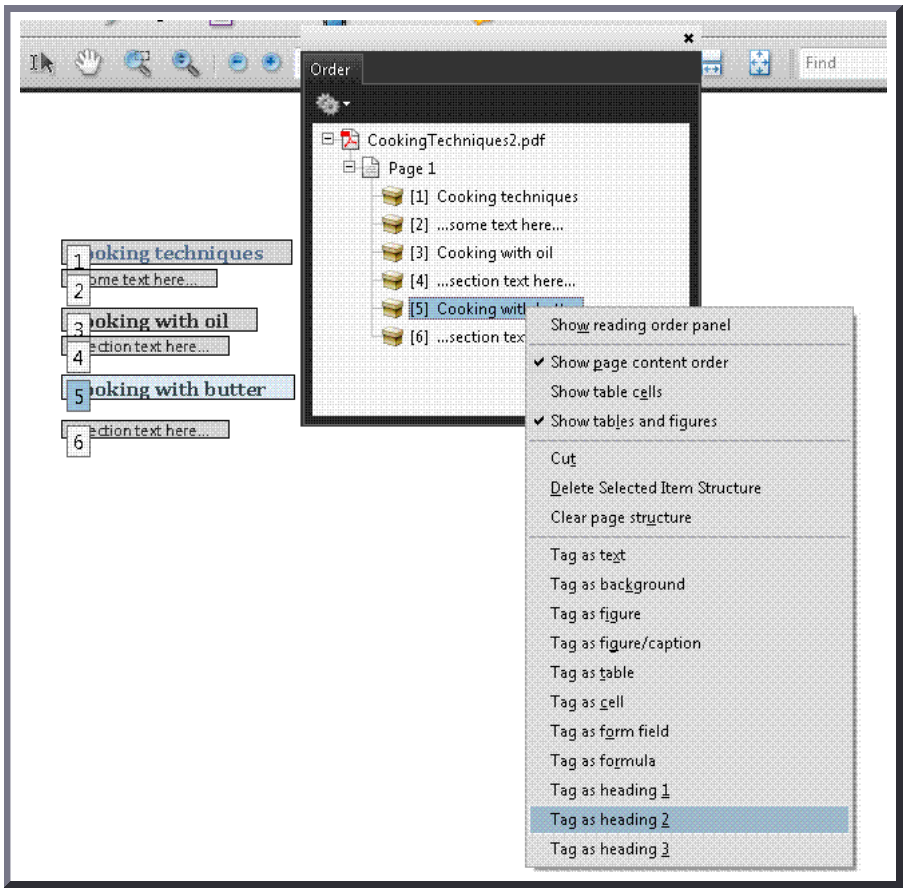


Using the Order and Tags panels

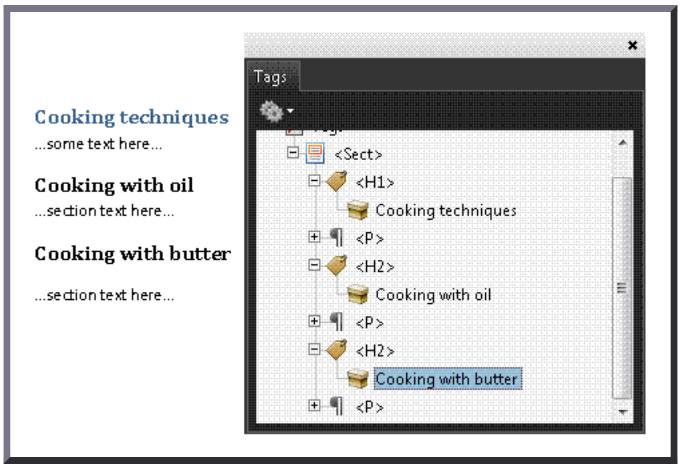
You can also add or change headings as follows:

1. Bring up the Order panel.
2. Access the context menu for the text to be changed or added as a heading.
3. Select the correct heading tag for the text.

The following screenshot shows Order panel and the context menu for the text "Cooking with butter." "Tag as heading 2" is selected in the context menu.



You can then check that the correct heading is applied by opening the Tags panel, as shown in the following screenshot.



Cooking techniques

...some text here...

Cooking with oil

...section text here...

Cooking with butter

...section text here...

```
Tags
<Sect>
  <H1>
    Cooking techniques
  <P>
  <H2>
    Cooking with oil
  <P>
  <H2>
    Cooking with butter
  <P>
```

This example is shown in operation in the working example of adding tagged headings (Word file) and working example of adding tagged headings (PDF file).

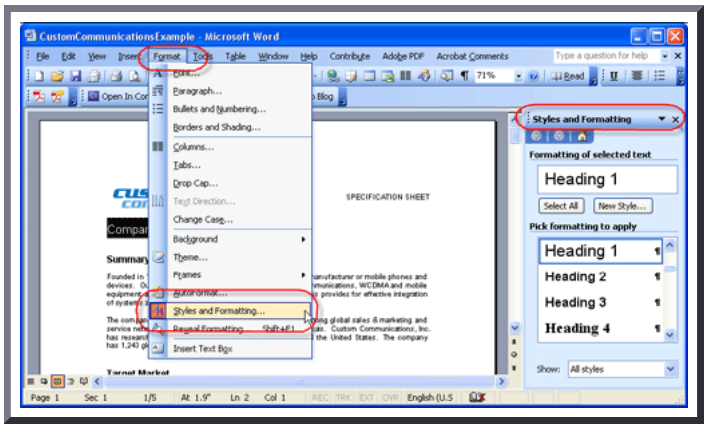
Example 2: Creating documents in Microsoft Word that have correctly tagged headings when converted to PDF

This example is shown with Microsoft Word. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

Use Styles to create heading formats: Heading 1, Heading 2, Heading 3, etc. Make styles progress in a logical manner; e.g., a Heading 2 should come after a Heading 1.

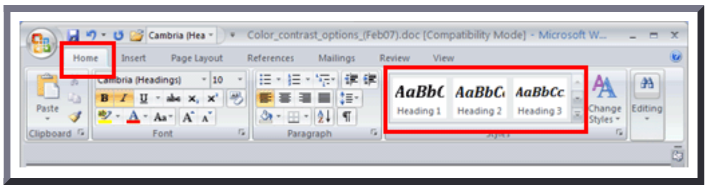
In Microsoft Word 2003

- Select the "Format > Styles and Formatting" menu item to reveal the styles and formatting task pane.
- Use the Heading 1 to Heading 6 styles provided in the "Styles and Formatting" panel.



In Microsoft Word 2007/2010

Select the Home Ribbon in Word 2007/2010 and select the appropriate heading (Heading 1 to Heading 6) from the Styles group.



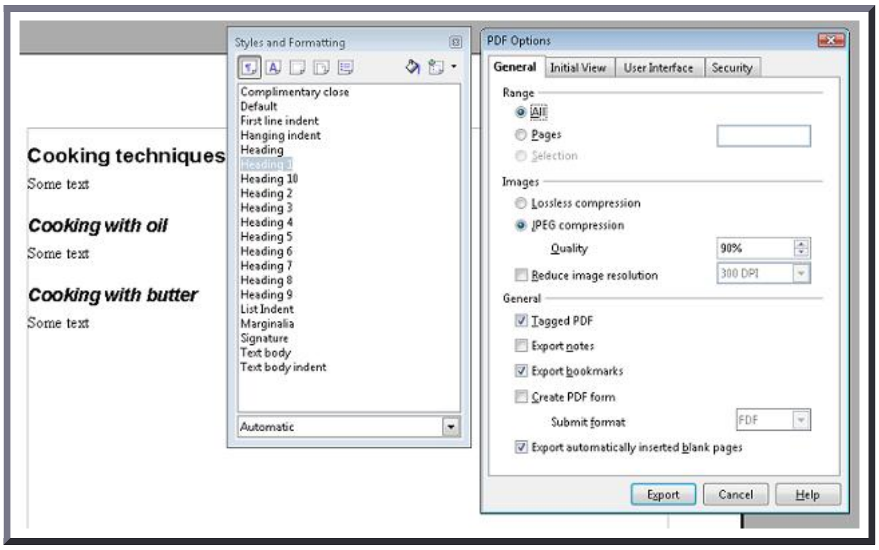
Example 3: Creating documents in OpenOffice.org Writer 2.2 that have correctly tagged headings when converted to PDF

This example is shown with OpenOffice.org Writer. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

Use Styles to create heading formats: Heading 1, Heading 2, Heading 3, etc. Make styles progress in a logical manner; e.g., a Heading 2 should come after a Heading 1.

Export to PDF as follows:

1. From the File menu, select Export as PDF...
2. The first time you export as PDF, an Options Dialog appears.
3. Select Tagged PDF, then select Export.


Cooking techniques

Some text

Cooking with oil

Some text

Cooking with butter

Some text

Styles and Formatting

Complimentary close
Default
First line indent
Hanging indent
Heading
Heading 1
Heading 10
Heading 2
Heading 3
Heading 4
Heading 5
Heading 6
Heading 7
Heading 8
Heading 9
List Indent
Marginalia
Signature
Text body
Text body indent

Automatic

PDF Options

General | Initial View | User Interface | Security

Range
- (●) All
- ( ) Pages
- ( ) Selection

Images
- ( ) Lossless compression
- (●) JPEG compression
  - Quality 90%
- [ ] Reduce image resolution 300 DPI

General
- [✓] Tagged PDF
- [ ] Export notes
- [✓] Export bookmarks
- [ ] Create PDF form
  - Submit format FDF
- [✓] Export automatically inserted blank pages

Export | Cancel | Help

Example 4: Marking up headings using /Hn elements

Headings within PDF documents can be marked up using /Hn elements elements in the structure tree, where n is numeral 1 through 6 (for example /H1, /H2, etc.).

The following code fragment illustrates code that is typical for using the /Hn elements elements to mark content. Note that /H1 has been role-mapped to /Head1 in this example. This is typically accomplished by an authoring tool.

```
0 obj% Document catalog
  << /Type /Catalog
     /Pages 100 0 R                % Page tree
     /StructTreeRoot 300 0 R       % Structure tree root
  >>
endobj
 ...
300 0 obj% Structure tree root
  << /Type /StructTreeRoot
     /K [ 301 0 R                  % Two children: a chapter
        304 0 R                    % and a paragraph
        ]
     /RoleMap << /Chap /Sect       % Mapping to standard structure types
                 /Head1 /H
                 /Para /P
              >>
    /ClassMap << /Normal 305 0 R >>  % Class map containing one attribute class
    /ParentTree 400 0 R            % Number tree for parent elements
    /ParentTreeNextKey 2           % Next key to use in parent tree
    /IDTree 403 0 R                % Name tree for element identifiers
  >>
endobj
301 0 obj                          % Structure element for a chapter
  << /Type /StructElem
     /S /Chap
     /ID (Chap1)                   % Element identifier
     /T (Chapter 1)                % Human-readable title
     /P 300 0 R                    % Parent is the structure tree root
     /K [ 302 0 R                  % Two children: a section head
          303 0 R                  % and a paragraph
        ]
  >>
endobj
302 0 obj                          % Structure element for a section head
  << /Type /StructElem
     /S /Head1
     /ID (Sec1.1)                  % Element identifier
     /T (Section 1.1)              % Human-readable title
     /P 301 0 R                    % Parent is the chapter
     /Pg 101 1 R                   % Page containing content items
     /A << /O /Layout              % Attribute owned by Layout
           /SpaceAfter 25
           /SpaceBefore 0
           /TextIndent 12.5
        >>
    /K 0                           % Marked-content sequence 0
  >>
endobj
...
```

Within marked content containers, headings can be marked up using /Headn elements as follows for a first-level heading in a PDF document:

```
BT                              % Start of text object
  /Head1 <</MCID 0 >>           % Start of marked-content sequence
     BDC
        ...
        (This is a first level heading. Hello world: ) Tj
        ...
     EMC                        % End of marked-content sequence
```

```
   ...
ET                              % End of text object
```

Resources

Resources are for information purposes only, no endorsement implied.

- Section 14.8.4.3.2 (Paragraphlike Elements) in PDF 1.7 (ISO 32000-1)
- PDF Accessibility Documentation:headings
- PDF and Accessibility

Related Techniques

- G141: Organizing a page using headings

Tests

Procedure

1. For all PDF content that is divided into separate sections, use one of the following to verify that headings are tagged correctly:
   - Read the PDF document with a screen reader, listening to hear that the list of headings is announced correctly.
   - Using a PDF editor, make sure the headings are tagged correctly.
   - Use a tool that is capable of showing the /Headn entries to open the PDF document and verify that headings are tagged correctly.
   - Use a tool that exposes the document through the accessibility API and verify that the headings are tagged correctly.

Expected Results

- #1 is true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF10: Providing labels for interactive form controls in PDF documents

Applicability

- Tagged PDF documents with forms.
- PDF forms created using Adobe LiveCycle Designer.

This technique relates to:

- Success Criterion 1.3.1 (Info and Relationships)
  - How to Meet 1.3.1 (Info and Relationships)
  - Understanding Success Criterion 1.3.1 (Info and Relationships)
- Success Criterion 3.3.2 (Labels or Instructions)
  - How to Meet 3.3.2 (Labels or Instructions)
  - Understanding Success Criterion 3.3.2 (Labels or Instructions)
- Success Criterion 4.1.2 (Name, Role, Value)
  - How to Meet 4.1.2 (Name, Role, Value)
  - Understanding Success Criterion 4.1.2 (Name, Role, Value)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF10. Also see PDF Technology Notes.

Description

The objective of this technique is to ensure that users of assistive technology are able to perceive form control labels and understand how form controls are used.

Form controls allow users to interact with a PDF document by filling in information or indicating choices which can then be submitted for processing. Assistive technology users must be able to recognize and understand the form fields, make selections, and provide input to complete the forms, and submit the form, just as sighted users can. Understandable labels that convey the purpose of each form control are essential to form accessibility.

Form inputs generally have labels and instructions to help users understand what information is required and how to fill in the form. Unless these labels are programmatically associated with the relevant fields, assistive technology might not be able to associate them correctly, and thus users might not understand how to complete the form.

Using Adobe Acrobat Pro with documents with interactive forms, you can make sure that the forms are accessible and usable by making sure that programmatically associated labels that convey the purpose of the fields are provided.

The heuristics used by assistive technology will sometimes use the text label if a programmatically associated label cannot be found. The TU entry (which is the tooltip) of the field dictionary is the programmatically associated label (see Example 3 below and Table 220 in PDF 1.7 (ISO 32000-1)). Therefore, add a tooltip to each field to provide a label that assistive technology can interpret.

Placement rules

The table below lists the placement rules governing where Adobe LiveCycle positions labels by default. Note that these rules assume left-to-right text directionality. If your form requires different positioning (e.g., to accommodate PDF documents in languages that use right-to-left text directionality), see Repositioning form labels in Example 2 below. In general, authors should review label positioning to make sure it meets the requirements of their particular form.

Control Type	LiveCycle Placement Rules
Text input (including date/time and password fields)	Default placement for the label is to the left of the control. If this is not possible, LiveCycle will attempt to place it immediately above the control.
Checkbox	Default placement for the label is to the right of the check box.
Radio button group	Default placement for the label for each individual radio button is to the right of the button. Create a visible caption for the radio button group by creating static text and placing it to the left of or above the group. (See Labeling radio buttons below.)
Combo box	Default placement for the label is to the left of the drop-down list. If this is not possible, LiveCycle will attempt to place it immediately above the control..
List box	Default placement for the label is above the list box.
Button	LiveCycle automatically places the label on the button; it does not have to be positioned manually. Ensure that the button's purpose is properly described in the label text.

Examples

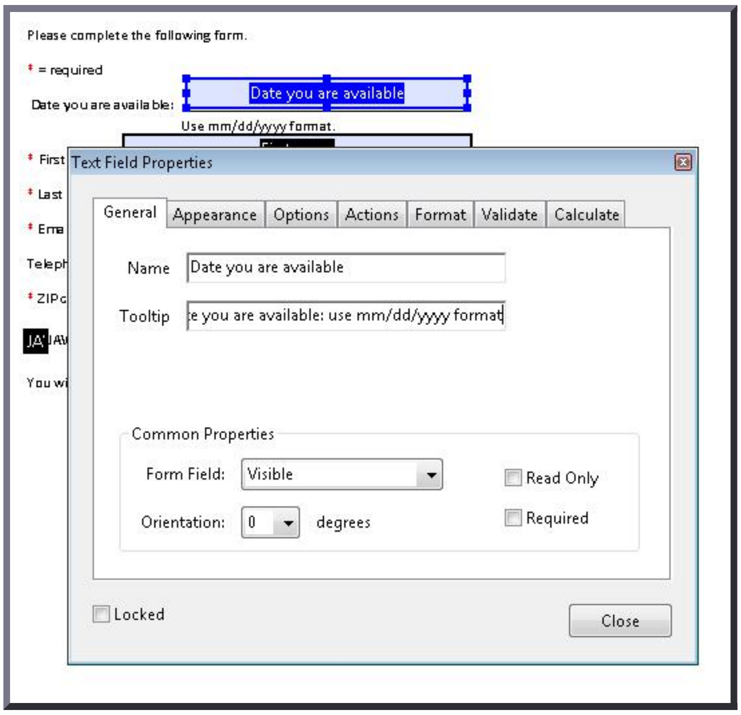
Example 1: Providing labels using the Forms tool in Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

As noted in the Description, text labels added in an authoring tool and then converted to PDF might be visually associated with the fields but are not programmatically associated, and you should provide a tooltip.

1. In the Forms menu, select Add or Edit Fields...
2. For the field you want to edit, access the context menu and select the Properties dialog.
3. In the General tab of the Properties dialog, type a description for the form field in the Tooltip field.
4. Repeat for all form fields.

The following image shows the Properties dialog with a description in the Tooltip field.


Please complete the following form.

* = required

Date you are available: Date you are available

Use mm/dd/yyyy format.

Text Field Properties

Name: Date you are available

Tooltip: e you are available: use mm/dd/yyyy format

Common Properties

Form Field: Visible

Orientation: 0 degrees

Read Only

Required

Locked

This example is shown in operation in the working example of providing labels using the forms tool.

Example 2: Providing labels to form controls in Adobe LiveCycle Designer ES 8.2.1

This example is shown with Adobe LiveCycle Designer. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

LiveCycle Designer provides several options for associating descriptive text and labels with form elements.

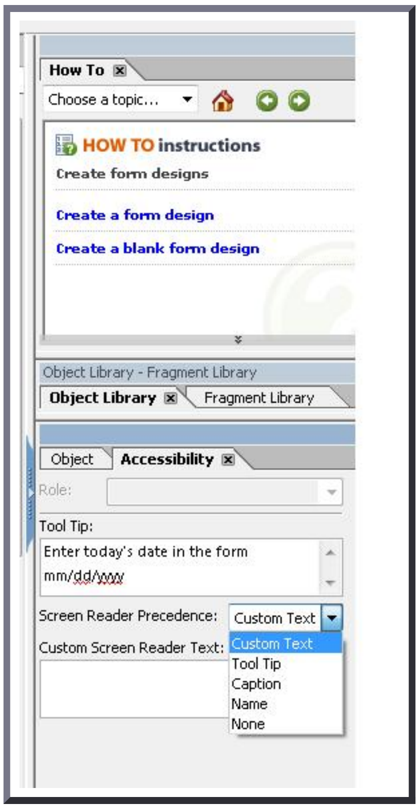
For sighted or low-vision users, it is important to properly position the label adjacent to the control. For screen reader users, you should also ensure that the label is programmatically associated with the form control and that sufficient information is provided so that screen reader users can readily complete and submit the form.

This example is shown in operation in the working example of providing labels in LiveCycle Designer.

Specifying accessible label text using the accessibility palette

In LiveCycle Designer, create or import a form. Then:

1. Enable the palette by selecting Window > Accessibility or by pressing shift + F6.
2. The palette appears in LiveCycle Designer's right-hand panel.
3. Select an object in your form. The palette shows the object's accessibility properties.



Tool Tip: Enter today's date in the form mm/dd/yyyy

Screen Reader Precedence: Custom Text

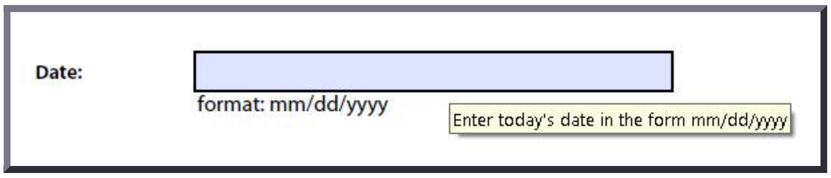
The label that a screen reader uses does not necessarily have to be the same as the visual caption. In some cases, you may want to provide more information about a form element's purpose.

To specify what text should be announced by the screen reader for a particular object, you can use the Accessibility Palette's Screen Reader Precedence drop down list. Text is announced in the order shown in the list: custom text, tool tip, caption, and name.

Depending on the complexity and difficulty of your form, you must decide which option best suits the requirements for your form.

By default, a screen reader searches for an object's text in order shown in the image. Once descriptive text has been found for a control, the search stops.

The image below shows an example of a text field with a visual caption that might be unclear for screen reader users. One of the fields has a caption of "Date" but screen reader users may want to know the preferred date format (shown as screen text). So this text is provided in the tooltip. Because a tooltip has a higher precedence than the visual caption, the screen reader uses the tooltip.



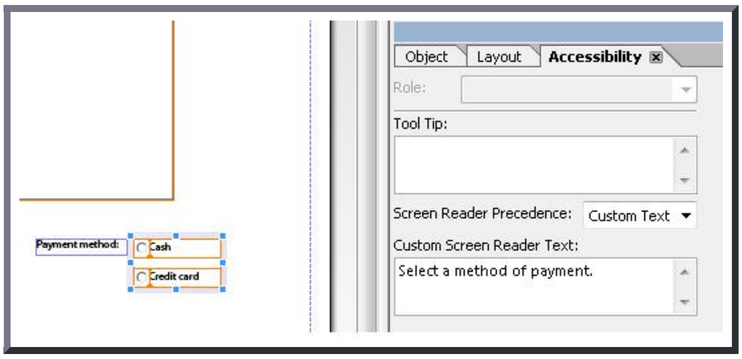
Labeling radio buttons

When a screen reader user tabs into a radio button, the screen reader needs to announce two items:

- A general description of the purpose of the group of buttons
- A meaningful description for the purpose of each radio button

To make radio buttons accessible:

1. In the Hierarchy palette, select the radio button group.
2. Select the Accessibility palette and in the Custom Screen Reader Text box, type the speak text for the group. For example, type "Select a method of payment."
3. In the Hierarchy palette, select the first radio button in the group.
4. In the Object palette, select the Field tab. In the Item area, select the item and type a meaningful value for the selected radio button. For example, type "Cash."
5. Repeat steps 3 and 4 for each radio button in the group.



Tool Tip:

Screen Reader Precedence: Custom Text

Custom Screen Reader Text: Select a method of payment.

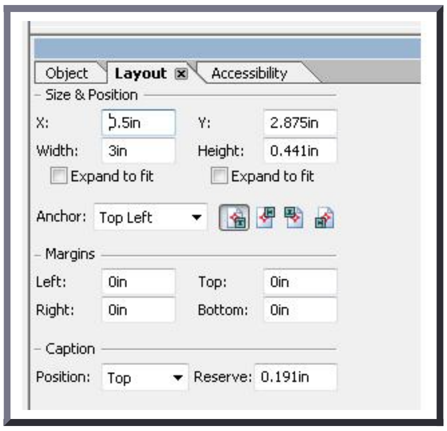
Repositioning form labels

The placement of a caption, or label, is important because users expect them to be found at a particular location adjacent to the control. For screen magnification users this is even more important, as they might not be able to view both the control and the label at the same time.

When you create an object, Adobe LiveCycle Designer automatically positions the label as specified by the control type (see the table in the Description above). For example, for a text field, the label is positioned to the left of the control.

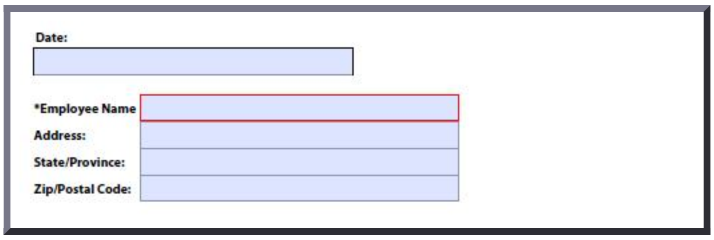
If you need to change the position of the label text (for example, to accommodate right-to-left text directionality):

1. Select the object by moving the focus to it.
2. In the Layout palette, under Caption at the bottom of the palette, select the position of your object from the Position drop-down list.



X: 5.5in Y: 2.875in
Width: 3in Height: 0.441in
Expand to fit Expand to fit
Anchor: Top Left
Left: 0in Top: 0in
Right: 0in Bottom: 0in
Position: Top Reserve: 0.191in

The resulting repositioned label is shown below. The label for the Date text field has been moved from the left of the field to the line above the field.



Date:

*Employee Name

Address:

State/Province:

Zip/Postal Code:

Example 3: Adding a tooltip to interactive form controls

The following code fragment illustrates the use of the TU entry to provide a tooltip (or programmatically associated text label) for a form field. This is typically accomplished by an authoring tool.

```
<< /AP -dict-
   /DA /Helv  0 Tf 0 g
   /DR -dict-
   /F 0x4
   /FT Tx              % FT key set to Tx for Text Field
   /P -dict-
   /Rect -array-
   /StructParent 0x1
   /Subtype Widget
   /T Date you are available   % Partial field name Date
   /TU Date you are available: use MM/DD/YYYY format % TU tool tip value serves as short description
   /Type Annot
   /V Pat Jones
>>
...
<Start Stream>
 BT
  /P <</MCID 0 >>BDC
  /CS0 cs 0  scn
  /TT0 1 Tf
    -0.001 Tc 0.003 Tw 11.04 0 0 11.04 72 709.56 Tm
    [(P)-6(le)-3(as)10(e)-3( )11(P)-6(rin)2(t)-3( Y)8(o)-7(u)2(r N)4(a)11(m)-6(e)]TJ
  0 Tc 0 Tw 9.533 0 Td
  ( )Tj
  -0.004 Tc 0.004 Tw 0.217 0 Td
  [(\()-5(R)-4(e)5(q)-1(u)-1(i)-3(r)-3(e)-6(d)-1(\))]TJ
 EMC
  /P <</MCID 1 >>BDC
  0 Tc 0 Tw 4.283 0 Td
  [( )-2( )]TJ
   EMC
   /ArtifactSpan <</MCID 2 >>BDC
   0.002 Tc -0.002 Tw 0.456 0 Td
  [(__)11(___)11(___)11(___)11(___)11(_)11(____)11(___)11(___)11(__)]TJ
  0 Tc 0 Tw 13.391 0 Td
  ( )Tj
  EMC
 ET
<End Stream>
```

Resources

Resources are for information purposes only, no endorsement implied.

- PDF 1.7 (ISO 32000-1)
- Adobe XML Forms Architecture (XFA)
- PDF and Accessibility

Related Techniques

- G131: Providing descriptive labels
- G162: Positioning labels to maximize predictability of relationships
- PDF23: Providing interactive form controls in PDF documents
- PDF5: Indicating required form controls in PDF forms
- PDF22: Indicating when user input falls outside the required format or values in PDF forms

Tests

Procedure

1. For each form control, verify visually that the label is positioned correctly in relation to the control.
2. For each form control, verify that the name is programmatically associated with the control by one of the following:
   - Open the PDF document with a tool that is capable of showing the name associated with the control and verify that the name is associated correctly with the control.
   - Use a tool that exposes the document through the accessibility API, and verify that the name is associated correctly with the control.

Expected Results

- #1 and #2 are true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF11: Providing links and link text using the Link annotation and the /Link structure element in PDF documents

Applicability

PDF documents that contain links

This technique relates to:

- Success Criterion 1.3.1 (Info and Relationships)
  - How to Meet 1.3.1 (Info and Relationships)
  - Understanding Success Criterion 1.3.1 (Info and Relationships)
- Success Criterion 2.1.1 (Keyboard)
  - How to Meet 2.1.1 (Keyboard)
  - Understanding Success Criterion 2.1.1 (Keyboard)
- Success Criterion 2.1.3 (Keyboard (No Exception))
  - How to Meet 2.1.3 (Keyboard (No Exception))
  - Understanding Success Criterion 2.1.3 (Keyboard (No Exception))
- Success Criterion 2.4.4 (Link Purpose (In Context))
  - How to Meet 2.4.4 (Link Purpose (In Context))
  - Understanding Success Criterion 2.4.4 (Link Purpose (In Context))

  Note: This technique must be combined with other techniques to meet SC 2.4.4. See Understanding SC 2.4.4 for details.
- Success Criterion 2.4.9 (Link Purpose (Link Only))
  - How to Meet 2.4.9 (Link Purpose (Link Only))
  - Understanding Success Criterion 2.4.9 (Link Purpose (Link Only))

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF11. Also see PDF Technology Notes.

Description

The purpose of this technique is to show how link text in PDF documents can be marked up to be recognizable by keyboard and assistive technology users. That is, the link information is programmatically available to user agents so that links are recognizable when presented in a different format. This is typically accomplished by using a tool for authoring PDF.

Links in PDF documents are represented by a Link tag and objects in its sub-tree, consisting of a link object reference (or Link annotation) and one or more text objects. The text object or objects inside the Link tag are used by assistive technologies to provide a name for the link.

The simplest way to provide links that comply with the WCAG success criteria is to create them when authoring the document, before conversion to PDF.

However, in some cases, it may not be possible to create the links using the original authoring tool. In these cases, Adobe Acrobat Pro can be used to create the link. But, because the tooltip created using the Link dialog in Adobe Acrobat Pro is not accessible to screen readers, be sure that the link text or the link context makes the purpose clear.

In all cases, link purpose should be made clear as described in the general techniques:

- G53: Identifying the purpose of a link using link text combined with the text of the enclosing sentence
- G91: Providing link text that describes the purpose of a link

Examples

Example 1: Creating a hyperlink in Microsoft Word 2007 before conversion to PDF

This example is shown with Microsoft Word. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

To create a hyperlink in Microsoft Word, first locate the item (e.g., web page) to link to. Then:

1. Either
   - Select Insert on the ribbon and select Hyperlink in the Links tools
   - Or, use the CTRL+K keyboard shortcut
2. On the Insert Hyperlink dialog, add the link destination and link text.
3. Save the file as tagged PDF. (See the PDF Technology Notes.)

Example 2: Creating a hyperlink in OpenOffice.org Writer 2.2 before conversion to PDF

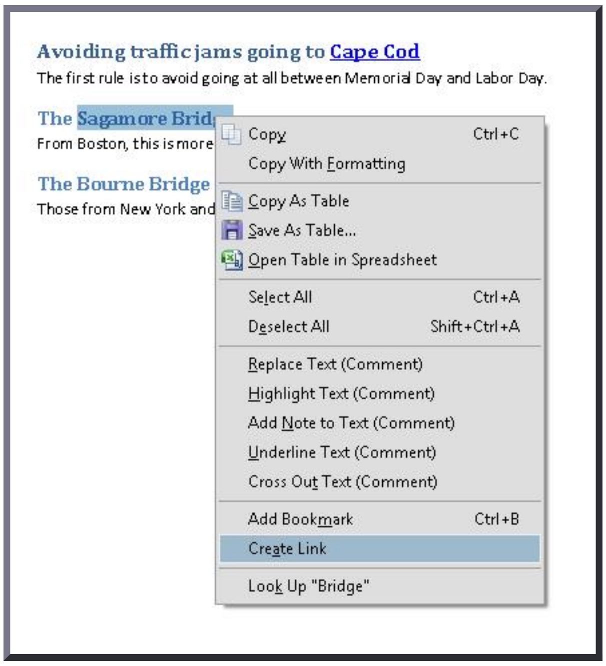
This example is shown with OpenOffice.org Writer. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. On the Insert menu, select Hyperlink.
2. In the Hyperlink dialog, insert the target URI in the Target field under Hyperlink Type.
3. Insert the link text in the Text field under Further Settings. (You can also select the link text from the document text before bringing up the dialog. The Text field will be filled in with the selected text.)
4. Save the file as tagged PDF. (See the PDF Technology Notes.)

Example 3: Creating a hyperlink using the Create Link dialog in Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. Select the text that will become the link text.
2. Access the context menu and select Create Link.


Avoiding traffic jams going to Cape Cod

The first rule is to avoid going at all between Memorial Day and Labor Day.

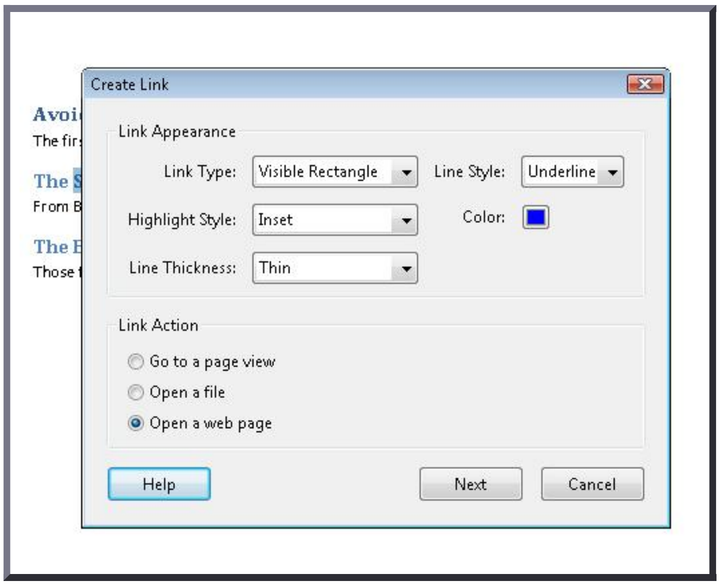
The Sagamore Brid

From Boston, this is more

The Bourne Bridge

Those from New York and

3. Follow the instructions in the Create Link dialog to specify the link appearance, as shown below.



Link Appearance

Link Type: Visible Rectangle

Line Style: Underline

Highlight Style: Inset

Color: 🟦

Line Thickness: Thin

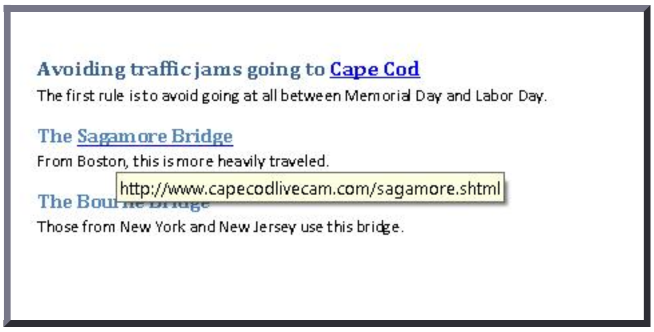
Link Action

○ Go to a page view

○ Open a file

● Open a web page

Then select Next and specify the URI. The image below shows the resulting hyperlink and tooltip.


Avoiding traffic jams going to Cape Cod

The first rule is to avoid going at all between Memorial Day and Labor Day.

The Sagamore Bridge

From Boston, this is more heavily traveled.

http://www.capecodlivecam.com/sagamore.shtml

The Bourne Bridge

Those from New York and New Jersey use this bridge.

This example is shown in operation in the working example of creating a hyperlink in PDF.

Example 4: Marking up link text using a /Link structure element

Link annotations in PDF documents are associated with a geometric region of a page rather than a particular object in a content stream. For this reason, link annotations alone are not useful for users with visual impairments, or to applications that must determine which content can be activated to invoke a hypertext link.

Tagged PDF /Link elements use PDF's logical structure to establish the association between content items and link annotations, providing functionality comparable to HTML hypertext links.

In HTML, the following example produces text containing a hypertext link:

```
Here is some text <a href="http://www.w3.org/WAI/"> with a link </a> inside.
```

In PDF the page must be painted first and then a link annotation placed over the area where the object action will occur.

The following code fragment shows PDF equivalent to the HTML above; it uses link text displayed in blue and underlined. A second code fragment follows, indicating the associated logical structure hierarchy. This is typically accomplished by an authoring tool.

```
/P <</MCID 0>>                                              %Marked Content Sequence 0 (paragraph)
 BDC                                                        %Begin marked content sequence
  BT                                                        %Begin text object
   /F1 11.04 Tf                                             %set text font and size
   1 0 0 1 72.024 709.54 Tm                                 %set text matrix
   0 g                                                      %set non stroking color to black
   0 G                                                      %set stroke color to black
  [(H)3(ere )-4(is s)10(o)5(m)-4(e)9( t)-3(e)9(xt)-3( )] TJ %Show text preceding the link" Here is some text"
  ET                                                        %end text object
 EMC                                                        %end marked content sequence

/Span <</MCID 1>>                                           %Marked Content Sequence 1 (underlined link text)
 BDC                                                        %Begin marked content sequence
  BT                                                        %Begin text object
   1 0 0 1 152.42 709.54 Tm                                 %set text matrix
   0 0 1 rg                                                 %set non-stroking color to blue
   0 0 1 RG                                                 %set stroke color to blue
   [(with a )-2(li)3(n)14(k)] TJ                            %Show link text " with a link"
  ET                                                        %end text object
   0 0 1 rg                                                 %set stroke color to blue
   152.42 707.62 45.984 0.72 re                             %rectangle operator - target area for the link
   f*                                                       %fill the path using the even-odd rule
 EMC                                                        %end marked content sequence

/P <</MCID 2>>                                              %Marked Content Sequence 2 (paragraph)
 BDC                                                        %Begin marked content sequence
  BT                                                        %begin text object
   1 0 0 1 198.41 709.54 Tm                                 %set text matrix
   0 g                                                      %set non stroking color to black
   0 G                                                      %set stroke color to black
   [( )] TJ                                                 %empty text string showing white space
  ET                                                        %end text object
  BT                                                        %begin text object
   1 0 0 1 200.93 709.54 Tm                                 %set text matrix
   [(in)5(sid)5(e.)] TJ                                     %show text following the link "inside."
  ET                                                        %end text
  BT                                                        %begin text object
   1 0 0 1 229.97 709.54 Tm                                 %set text matrix
   [( )] TJ                                                 %empty text string showing white space
  ET                                                        %end text object
 EMC                                                        %end marked content sequence
```

The following code fragment is an excerpt from the logical structure that establishes the association between the content items and the link annotation:

```
11 0 obj                                          %Object ID 11, generation   0, obj keyword
 <</K[1                                           %immediate child of the structure tree root
  <<
   /Obj 26 0 R                                    %reference to Object 26
   /Type/OBJR                                     %this object describes an indirect object reference
  >>]
   /P 12 0 R
   /Pg 17 0 R
   /S/Link
 >>
endobj

26 0 obj                                          %object ID 26 which is referenced by the OBJR in Object 11
 <</A 31 0 R
  /BS
  <</S/S
    /Type/Border
    /W 0
  >>
  /Border[0 0 0]                                  %a colorless border
  /H/I
  /Rect[150.128 694.558 200.551 720.0]            %the boundaries defining target area where link annotation is active
  /StructParent 1
  /Type/Annot                                     %Structure element is an annotation
  /Subtype/Link
 >>                                               %It is a link annotation
endobj
31 0 obj                                          %Object 31, gen 0, obj
 <</S/URI                                         %Object type is URI action
   /URI(http://www.w3.org/WAI)                    %The Uniform resource identifier to resolve
 >>
endobj
```

Resources

Resources are for information purposes only, no endorsement implied.

- Section 14.8.4.4.2 (Link Elements) in PDF 1.7 (ISO 32000-1)
- PDF and Accessibility

Related Techniques

- G53: Identifying the purpose of a link using link text combined with the text of the enclosing sentence
- G91: Providing link text that describes the purpose of a link
- PDF13: Providing replacement text using the /Alt entry for links in PDF documents

Tests

Procedure

For each hyperlink, verify that the link is correctly tagged and the link text is properly exposed:

1. Read the PDF document with a screen reader, listening to hear that the link is read correctly and that it describes the purpose of the link (i.e., its destination).
2. Visually scan the tag tree to verify that the link is tagged correctly and the link text is exposed (for screen magnifier users and sighted users with cognitive disabilities).
3. Use a tool that is capable of showing the /Link entry value to open the PDF document and view the hyperlink and link text.
4. Use a tool that exposes the document through the accessibility API and verify that the link has the correct link text.
5. Tab to each link and check that it can be followed to its target by pressing Enter.

Expected Results

- #1 or #2 or #3 or #4 is true.
- #5 is true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF12: Providing name, role, value information for form fields in PDF documents

Applicability

Tagged PDF documents with interactive form fields.

This technique relates to:

- Success Criterion 1.3.1 (Info and Relationships)
  - How to Meet 1.3.1 (Info and Relationships)
  - Understanding Success Criterion 1.3.1 (Info and Relationships)
- Success Criterion 4.1.2 (Name, Role, Value)
  - How to Meet 4.1.2 (Name, Role, Value)
  - Understanding Success Criterion 4.1.2 (Name, Role, Value)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF12. Also see PDF Technology Notes.

Description

The objective of this technique is to ensure that assistive technologies can gather information about and interact with form controls in PDF content.

The types of PDF form controls are: text input field, check box, radio button, combo box, list box, and button.

Providing name, role, state, and value information for all form components enables compatibility with assistive technology, such as screen readers, screen magnifiers, and speech recognition software used by people with disabilities.

The PDF specification defines how name, role, and value are set for form controls in Section 12.7.4 (Field Types) of PDF 1.7 (ISO 32000-1), as shown in the following table. The Comments column explains how Adobe Acrobat Pro displays the corresponding information.

Interactive Form Dictionary Entries	Used to Define	Comments
FT	Role	Controls that share field type also use field flags to set the appropriate role. In Adobe Acrobat the role for form controls is set automatically.
TU	Name	In Adobe Acrobat the TU entry value is provided via the Tooltip field in the form control's Properties dialog. This should not be confused with the T entry which is defined as the Name in Acrobat's form control properties dialog - the name field in the Properties dialog is not used to provide the name for a control when read by assistive technologies.
CA	Name (Pushbuttons only)	In Adobe Acrobat the CA entry value is provided via the label field in the form control's Properties dialog.
V	Value	The Value entry is set by the user interacting with the control, where a value is needed.
DV	Default Value	In Adobe Acrobat the DV entry value can be set in the form control's Properties dialog.

The following table describes how the role, name, value, and state are defined for PDF form controls created using Adobe Acrobat Pro. Adobe LiveCycle Designer provides the same controls as well as several additional ones: see Example 2 below.

PDF form	Role (FT entry)	Name (TU entry)	Value (V entry)	Configurable

element				States
Text field	Text /Tx	Tooltip	Default value (DV entry in field dictionary) can be set in the Properties dialog. Value is entered by user.	Read Only, Required, Multiline, Password
Check box	Check box /Btn	Tooltip	V entry is set to 'Yes' or 'No' depending on Checked state.	Read Only, Required, Checked
Radio button	Radio button /Btn (Field Flag set to 'Radio')	Tooltip	V entry is set to 'Yes' or 'No' depending on Checked state.	Read Only, Required, Checked
Combo box	Combo box /Ch (Field Flag set to 'Combo')	Tooltip	Default value (/DV) can be set in the Properties dialog. Value is determined by user selection.	Read Only, Required
List box	Drop-down list /Ch	Tooltip	Default value (/DV) can be set in the Properties dialog. Value is determined by user selection.	Read Only, Required
Button	Push button /Btn (Field Flag set to 'Pushbutton')	Label (CA entry instead of TU entry)	Push buttons do not have or require a value.	Read Only, Required
Signature field	Text /Sig	Tooltip	Default value (DV entry in field dictionary) can be set in the Properties dialog. Value is entered by user.	Read Only, Required

Examples

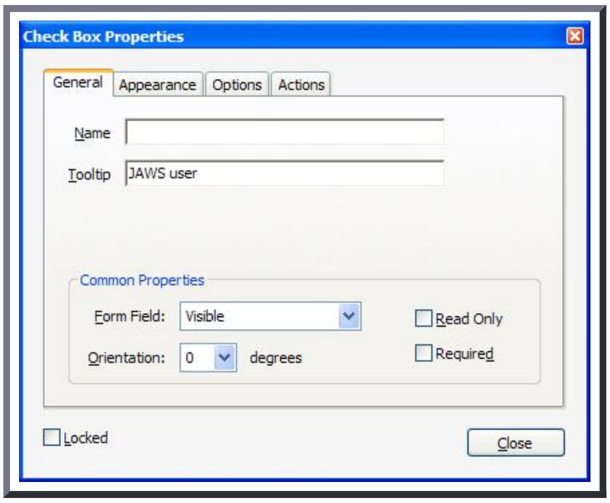
Example 1: Specifying name, role, value and/or state for a form field using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

This example uses a check box for illustration; the procedure is the same for other form controls. In Form Editing mode:

1. Access the context menu for the form field you are creating or modifying.
2. Select the Properties... dialog for the form field.
3. Specify the name by adding a value to the tool tip field. This will used by the accessibility API as the Name for the control and should usually be set to match the text used as a visual label for the control.
4. Select the Options tab.
5. Specify the default value and the default state, if appropriate.

The image below shows the Check Box Properties dialog, open in the General tab. (The Name field in the dialog is not needed for accessibility.)


Check Box Properties

General | Appearance | Options | Actions

Name:

Tooltip: JAWS user

Common Properties

Form Field: Visible

Orientation: 0 degrees

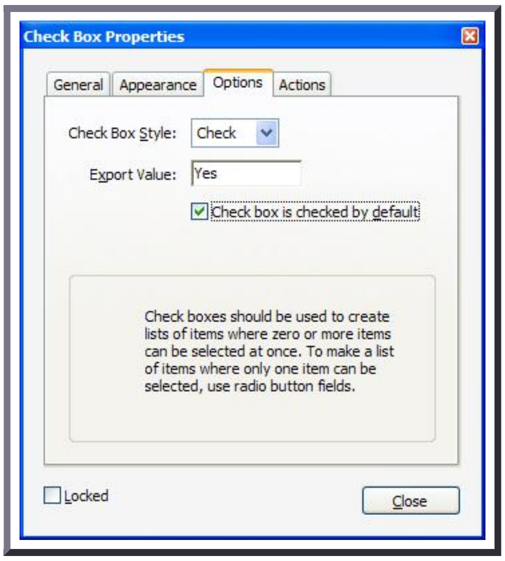
Read Only

Required

Locked

Close

The image below shows the Check Box Properties dialog, open in the Options tab.


Check Box Properties

General | Appearance | Options | Actions

Check Box Style: Check

Export Value: Yes

✓ Check box is checked by default

Check boxes should be used to create lists of items where zero or more items can be selected at once. To make a list of items where only one item can be selected, use radio button fields.

Locked

Close

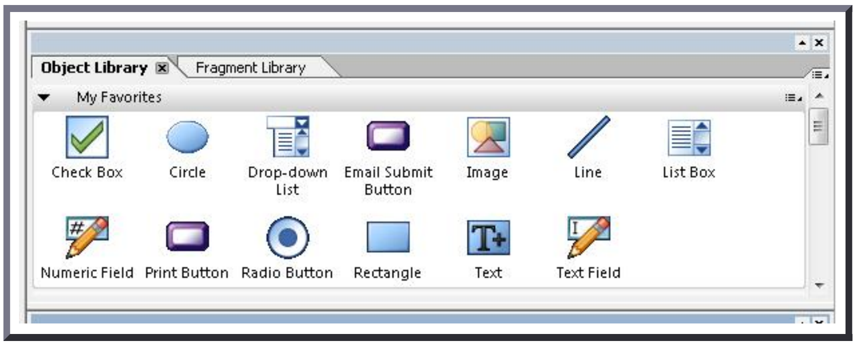
This example is shown in operation in the working example of specifying name, role, value using Acrobat Pro.

Example 2: Specifying name, value, and state for a form field using Adobe LiveCycle Designer ES 8.2.1

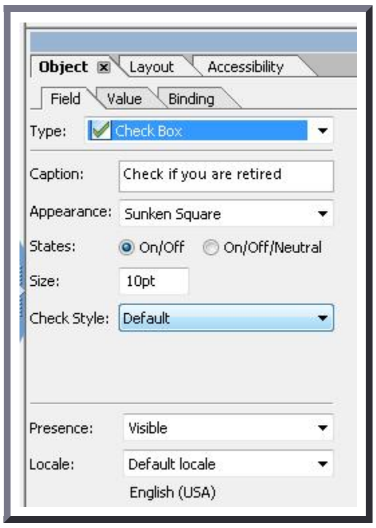
This example is shown with Adobe LiveCycle Designer. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

In Adobe LiveCycle Designer, you use the Object Library to create form objects and the Object Palette to specify name, role, state or value for the object.

The following image shows the Object Palette.

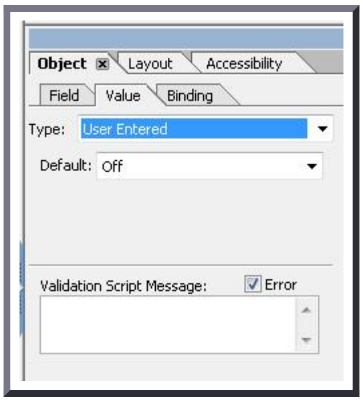


The following three images show the tabs in the Object palette. In the first the Field tab is open for specifying the type (or role) of the field.



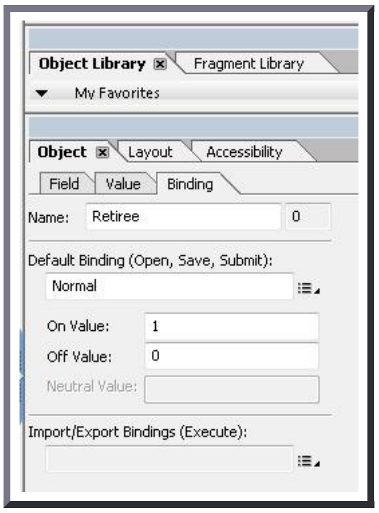
Type: Check Box
Caption: Check if you are retired
Appearance: Sunken Square
States: [✓ On/Off] On/Off/Neutral
Size: 10pt
Check Style: Default
Presence: Visible
Locale: Default locale
English (USA)

The next image shows the Value tab, with options that can be applied to the field.



Type: User Entered
Default: Off
Validation Script Message: ✓ Error

The third images shows the Binding tab, specifying the name of the field.



Name: Retiree 0

Default Binding (Open, Save, Submit): Normal

On Value: 1

Off Value: 0

Neutral Value:

Import/Export Bindings (Execute):

This example is shown in operation in the working example of specifying name, role, value using LiveCycle Designer.

Example 3: Adding a checkbox in a PDF document using the /Btn field type

The following code fragment illustrates code that is typical for a simple check box field such as shown in Examples 1 and 2. This is typically accomplished by an authoring tool.

```
1 0 obj
  << /FT /Btn     % Role
     /TU Retiree  % Name
     /V /Yes      % Value
     /AS /Yes
     /AP << /N << /Yes 2 0 R /Off 3 0 R>>
  >>
endobj
```

Resources

Resources are for information purposes only, no endorsement implied.

- Section 12.7.4 (Field Types) of PDF 1.7 (ISO 32000-1)
- Adobe XML Forms Architecture (XFA)
- PDF and Accessibility

Related Techniques

- PDF23: Providing interactive form controls in PDF documents
- PDF5: Indicating required form controls in PDF forms
- PDF22: Indicating when user input falls outside the required format or values in PDF forms

Tests

Procedure

1. For the form control, verify that name, role, and value/state are specified by one of the following:
   - Use a screen reader to navigate to the form control and check that it can be activated or that its value can be changed. Verify that the name (tooltip) and role are announced.
   - Use a tool capable of showing the form field information to open the PDF document and verify that the form control has the correct name, role, value, and state (if appropriate) information.
   - Use a tool that exposes the document through the accessibility API, and verify that the form control has the correct name, role, value, and state (if appropriate) information.

Expected Results

- #1 is true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF13: Providing replacement text using the /Alt entry for links in PDF documents

Applicability

Tagged PDF documents that contain links.

This technique relates to:

- Success Criterion 2.4.4 (Link Purpose (In Context))
  - How to Meet 2.4.4 (Link Purpose (In Context))
  - Understanding Success Criterion 2.4.4 (Link Purpose (In Context))

Note: This technique must be combined with other techniques to meet SC 2.4.4. See Understanding SC 2.4.4 for details.

- Success Criterion 2.4.9 (Link Purpose (Link Only))
  - How to Meet 2.4.9 (Link Purpose (Link Only))
  - Understanding Success Criterion 2.4.9 (Link Purpose (Link Only))

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF13. Also see PDF Technology Notes.

Description

The objective of this technique is to provide replacement link text via the /Alt entry in the property list for a tag. This is usually not necessary, but in some situations, additional information beyond the visible link text is needed, particularly for screen reader users. Screen readers can read visible link text, but replacing the screen text with meaningful alternate text for links in a PDF document can make links more accessible.

Links in PDF documents are represented by a Link tag and objects in its sub-tree, consisting of a link object reference (or Link annotation) and one or more text objects. The text object or objects inside the Link tag are used by assistive technologies to provide a name for the link.

Authors can replace the default link text by providing an /Alt entry for the Link tag. When the Link tag has an /Alt entry, screen readers ignore the value of any visible text objects in the Link tag and use the /Alt entry value for the link text.

The simplest way to provide context-independent link text that complies with the WCAG 2.0 success criteria is to create them when authoring the document, before conversion to PDF. In some cases, it may not be possible to create the links using the original authoring tool. When editing PDF documents with Adobe Acrobat Pro, the best way to create accessible links is to use the Create Link command.

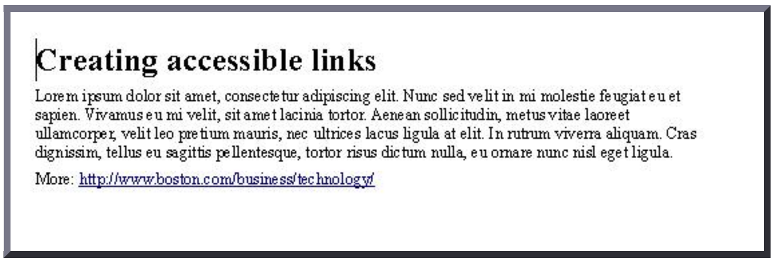
Authors should make sure that the alternate text makes sense in context of the screen text before and after the link.

Examples

Example 1: Adding alternate link text using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

The image below shows a document converted to PDF from Oracle Open Office. Note that the visible link text is the URL for the link target. A screen reader will read the entire URI as the link text.



Creating accessible links

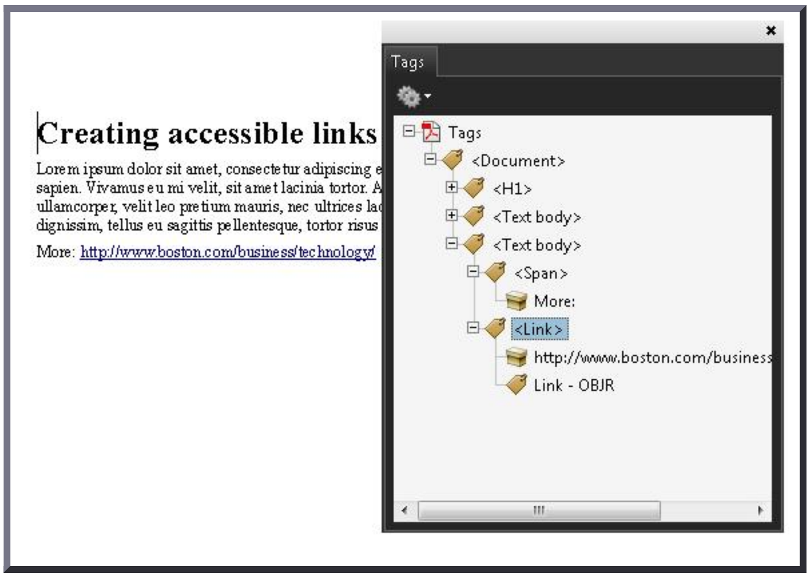
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc sed velit in mi molestie feugiat eu et sapien. Vivamus eu mi velit, sit amet lacinia tortor. Aenean sollicitudin, metus vitae laoreet ullamcorper, velit leo pretium mauris, nec ultrices lacus ligula at elit. In rutrum viverra aliquam. Cras dignissim, tellus eu sagittis pellentesque, tortor risus dictum nulla, eu ornare nunc nisl eget ligula.

More: http://www.boston.com/business/technology/

To create more accessible link text for assistive technology:

1. In the View menu, open the Tag panel by selecting Navigation Panels > Tags.
2. Locate the Link tag in the tag tree, access the context menu for the link, and select Properties.
3. In the TouchUp Properties dialog, in the Tags tab, enter replacement text in the Alternate Text field. Screen readers will read this text instead of the entire URI.

The next image shows the Link tag structure in the Tag panel.



Creating accessible links

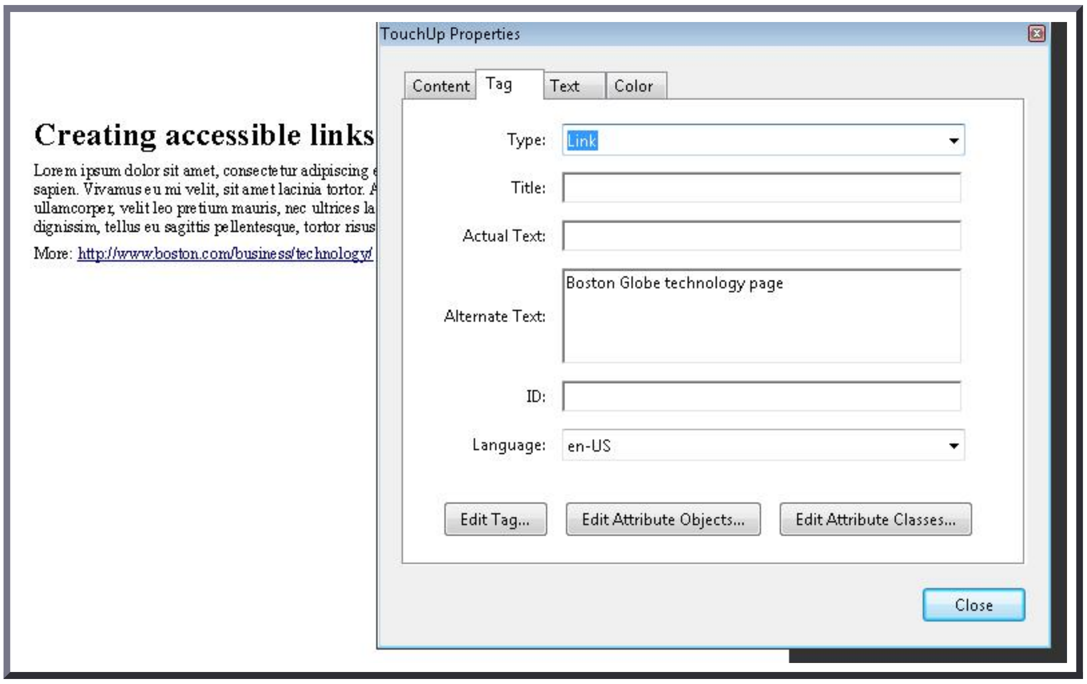
Lorem ipsum dolor sit amet, consectetur adipiscing e
sapien. Vivamus eu mi velit, sit amet lacinia tortor. A
ullamcorper, velit leo pretium mauris, nec ultrices la
dignissim, tellus eu sagittis pellentesque, tortor risus

More: http://www.boston.com/business/technology/

```
Tags
  <Document>
    <H1>
    <Text body>
    <Text body>
      <Span>
        More:
      <Link>
        http://www.boston.com/business
        Link - OBJR
```

The last image shows the Alternate Text specified in the Link tag's TouchUp Properties dialog. A screen reader will read the

Alternate Text as the link text.



Creating accessible links

Lorem ipsum dolor sit amet, consectetur adipiscing e sapien. Vivamus eu mi velit, sit amet lacinia tortor. A ullamcorper, velit leo pretium mauris, nec ultrices la dignissim, tellus eu sagittis pellentesque, tortor risus

More: http://www.boston.com/business/technology/

TouchUp Properties

Type: Link

Title:

Actual Text:

Alternate Text: Boston Globe technology page

ID:

Language: en-US

This example is shown in operation in the working example of adding alternate link text (OpenOffice file) and working example of adding alternate link text (PDF file).

Example 2: Adding alternate link text in a PDF document using the /Alt entry

The following code fragment illustrates code that is typical for alternative text for a link. This is typically accomplished by an authoring tool.

```
32 0 obj
<<
  /S/URI                                       %Action type (required), must be URI for a URI action
  /URI(http://www.boston.com/business/technology/)  %Uniform resource identifier(required), the URI to be resolved
>>
endobj
```

The following illustrates how to specify alternate text for the URL in the above link:

```
11 0 obj
<<
  /Alt(Boston Globe technology page)    %Alternate text entry
  /K [ 1
       <<
         /Obj 27 0 R
         /Type /OBJR            %Object reference to the link
       >>
       ]
  /P 12 0 R
  /Pg 18 0 R
  /S
  /Link
>>
endobj
```

Resources

Resources are for information purposes only, no endorsement implied.

- Section 14.9.4 (Replacement Text) in PDF 1.7 (ISO 32000-1)
- PDF and Accessibility

Related Techniques

- G53: Identifying the purpose of a link using link text combined with the text of the enclosing sentence
- G91: Providing link text that describes the purpose of a link
- G149: Using user interface components that are highlighted by the user agent when they receive focus
- PDF11: Providing links and link text using the Link annotation and the /Link structure element in PDF documents

Tests

Procedure

1. For the hyperlink, verify that the alternate link text is properly coded by one of the following:
   - Read the PDF document with a screen reader, listening to hear that the alternate link text is read correctly.
   - Use a tool that is capable of showing the /Alt entry to open the PDF document and view the hyperlink and alternate link text.

- Use a tool that exposes the document through the accessibility API and verify that the alternate link text is the text for the link.

Expected Results

- #1 is true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF14: Providing running headers and footers in PDF documents

Applicability

Tagged PDF documents

This technique relates to:

- Success Criterion 2.4.8 (Location)
  - How to Meet 2.4.8 (Location)
  - Understanding Success Criterion 2.4.8 (Location)
- Success Criterion 3.2.3 (Consistent Navigation)
  - How to Meet 3.2.3 (Consistent Navigation)
  - Understanding Success Criterion 3.2.3 (Consistent Navigation)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF14. Also see PDF Technology Notes.

Description

The objective of this technique is to help users locate themselves in a document by providing running headers and footers via pagination artifacts. This is normally accomplished using a tool for authoring PDF.

Running headers and footers help make content easier to use and understandable by providing repeated information in a consistent and predictable way. The content of headers and footers will vary widely depending on the document scope and content, the audience, and design decisions. Some examples of location information that may be used in headers and footers are listed below. Whether the information appears in a header or a footer is often a design decision; page numbers often appear in footers but they may alternatively appear in headers.

- Document title
- Current chapter and/or section in the document
- Page numbers with location information such as, "Page 3-4" or "Page 9 of 15."
- Author and/or date information.

Consistency helps users with cognitive limitations, screen-reader users and low-vision magnifier users, and users with intellectual disabilities understand content more readily.

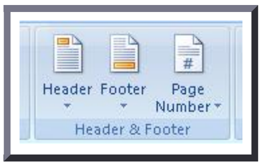
The easiest way to provide page headers and footers is in the authoring tool for the document. Authoring tools typically provide features for creating header and footer text and information (such as page numbers). However, if after converting your document to PDF, you need to add or modify page headers and footers, authoring or repair tools like Adobe Acrobat Pro's Header & Footer tools can be used. In all cases, the tools generate page headers and footers in consistent and predictable layout, format, and text.

Examples

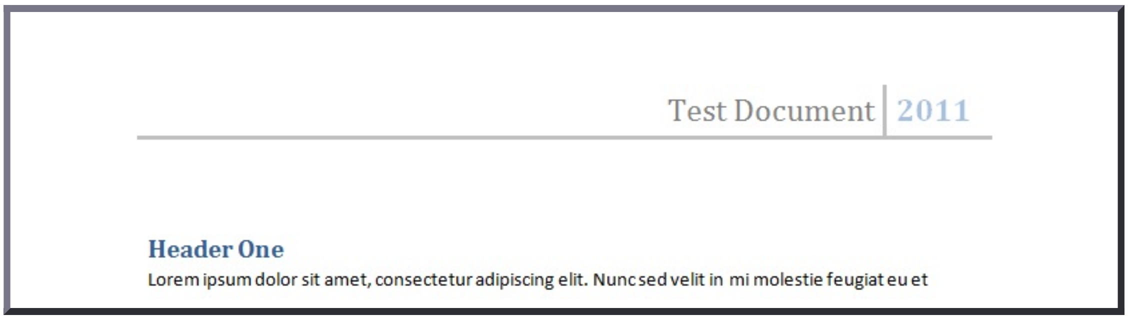
Example 1: Adding running headers and footers using Microsoft Word 2007

This example is shown with Microsoft Word. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

In Microsoft Word, use the Insert ribbon, which allows you to specify header, footer, and page number information and layout, as shown in the following images.

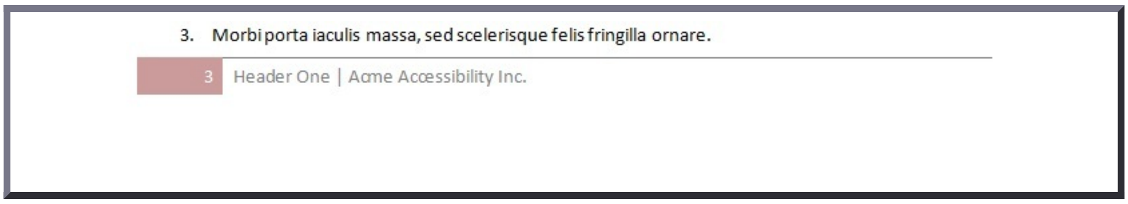


You can use these tools to specify headers and footers as shown in the following images:


Test Document | 2011

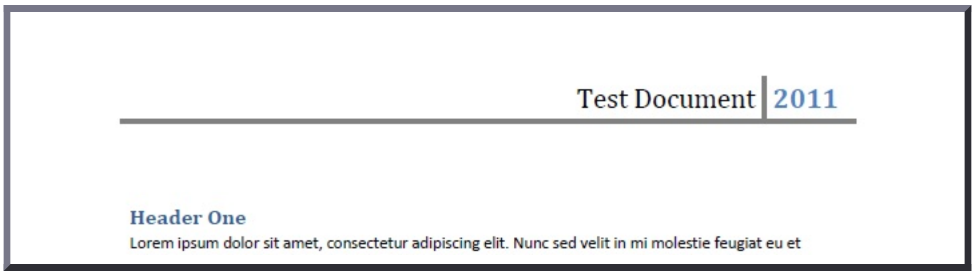
Header One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc sed velit in mi molestie feugiat eu et


3. Morbi porta iaculis massa, sed scelerisque felis fringilla ornare.

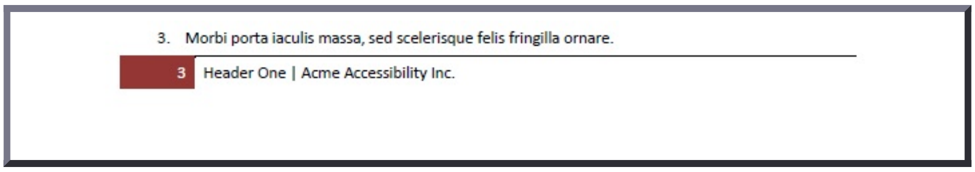
3 Header One | Acme Accessibility Inc.

When converted to PDF, the page headers and footers appear in the document.


Test Document | 2011

Header One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc sed velit in mi molestie feugiat eu et


3. Morbi porta iaculis massa, sed scelerisque felis fringilla ornare.

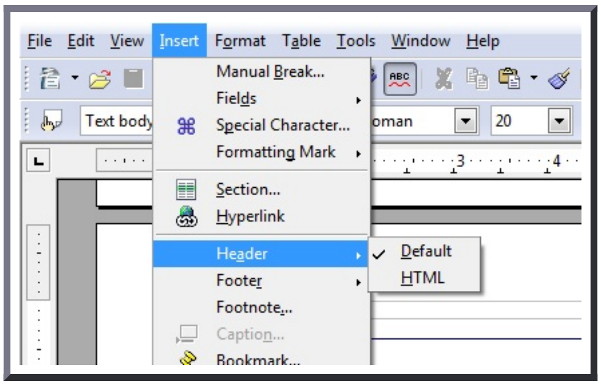
3 Header One | Acme Accessibility Inc.

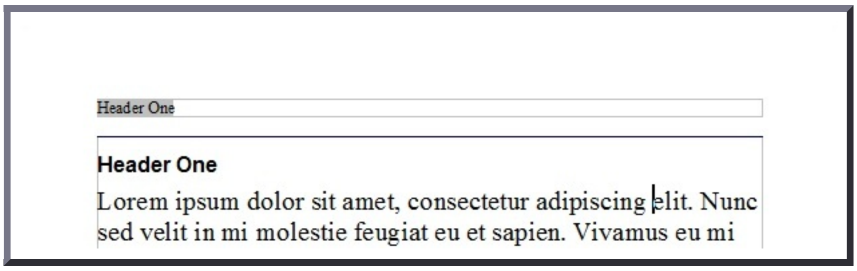
This example is shown in operation in the working example of adding running headers using Word (Word file) and working example of adding running headers using Word (PDF file).

Example 2: Adding running headers and footers using OpenOffice.org Writer 2.2

This example is shown with OpenOffice.org Writer. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

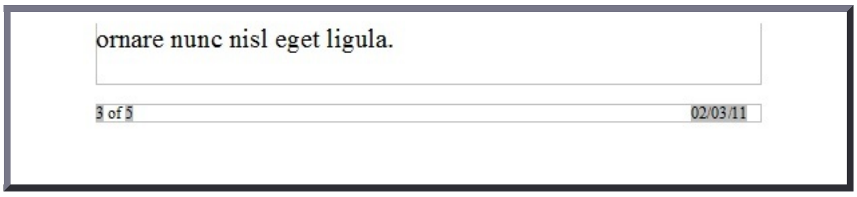
In OpenOffice.org Writer, use the Insert > Header and Insert > Footer tools, which allow you to specify header and footer information and layout, as shown in the following images.




Header One

Header One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc sed velit in mi molestie feugiat eu et sapien. Vivamus eu mi



ornare nunc nisl eget ligula.

3 of 5 02/03/11

When converted to PDF, the page headers and footers appear in the document as they do in the converted Word document in Example 1.

This example is shown in operation in the working example of adding running headers using OpenOffice Writer (OpenOffice file) and working example of adding running headers using OpenOffice Writer (PDF file).

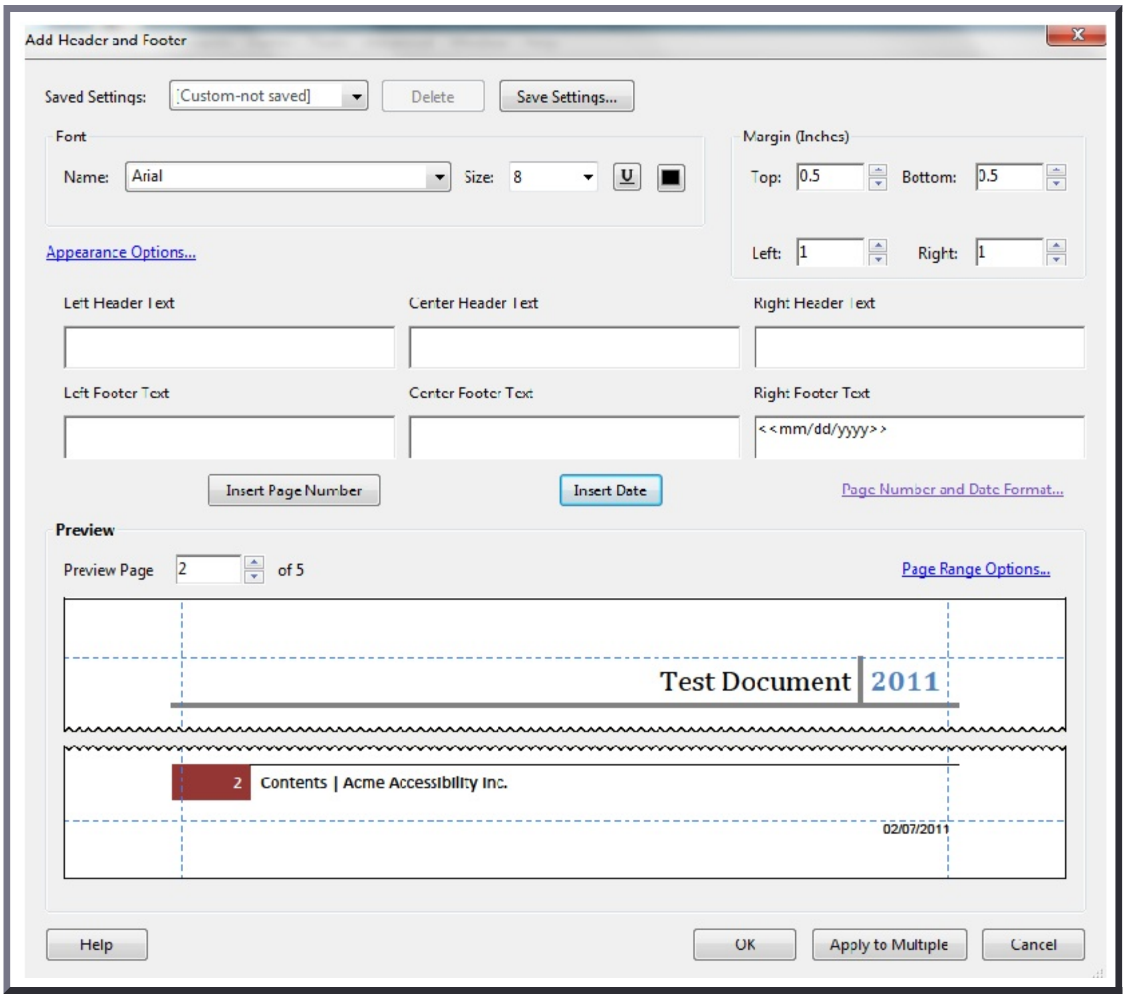
Example 3: Adding running headers and footers to PDF documents using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

In Adobe Acrobat Pro, you can add or modify headers and footers:

1. Select Document > Header & Footer > Add...
2. In the Add Header and Footer tool, specify text and formats for headers and footers in your document.
3. Use the Previews to make sure the text, fonts, and layout are as you want them for your document.

The image below shows Acrobat Pro's Add Header and Footer tool.



Add Header and Footer

Saved Settings: [Custom-not saved]

Font: Name: Arial, Size: 8

Margin (Inches): Top: 0.5, Bottom: 0.5, Left: 1, Right: 1

Right Footer Text: <<mm/dd/yyyy>>

Preview Page 2 of 5

Test Document | 2011

2 Contents | Acme Accessibility Inc.

02/07/2011

Example 4: Marking a running header or footer as a pagination artifact in a PDF document using an /Artifact tag or property list

The PDF specification allows running headers and footers to be marked as "pagination artifacts" as defined in section 14.8.2.2 "Real Content and Artifacts," of PDF 1.7 (ISO 32000-1).

An artifact is explicitly distinguished from real content by enclosing it in a marked-content sequence with the /Artifact tag.

```
/Artifact
BMC
...
EMC
```

or

```
/Artifact propertyList
BDC
```

```
...
EMC
```

The first is used to identify a generic artifact; the second is used for artifacts that have an associated property list. Note: to aid in text reflow, artifacts should be defined with property lists whenever possible. Artifacts lacking a specified bounding box are likely to be discarded during reflow.

Property list entries for artifacts include Type, BBox, Attached, and Subtype.

Resources

Resources are for information purposes only, no endorsement implied.

- Section 14.8.2.2 (Real Content and Artifacts) in PDF 1.7 (ISO 32000-1)
- PDF and Accessibility

Related Techniques

- G61: Presenting repeated components in the same relative order each time they appear
- PDF9: Providing headings by marking content with heading tags in PDF documents
- PDF2: Creating bookmarks in PDF documents

Tests

Procedure

1. Check that running headers and/or footers are provided and contain information to help users locate themselves within the document (such as page numbers or chapter numbers).
2. If section headers are used in the running header or footer, check that the section header and the running header or footer are consistent.

Expected Results

- #1 and #2 are true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF15: Providing submit buttons with the submit-form action in PDF forms

Applicability

Tagged PDF documents with forms.

This technique relates to:

- Success Criterion 3.2.2 (On Input)
  - How to Meet 3.2.2 (On Input)
  - Understanding Success Criterion 3.2.2 (On Input)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF15. Also see PDF Technology Notes.

Description

The objective of this technique is to provide a mechanism that allows users to explicitly request a change of context using the submit-form action in a PDF form. The intended use of a submit button is to generate an HTTP request that submits data entered in a form, so it is an appropriate control to use for causing a change of context. In PDF documents, submit buttons are normally implemented using a tool for authoring PDF.

Examples 1 and 2 demonstrate how to add a submit button using specific authoring tools. There are other PDF tools that perform similar functions. Check the functionality provided by PDF Authoring Tools that Provide Accessibility Support.

Examples

Example 1: Adding a submit button using Adobe Acrobat 9 Pro

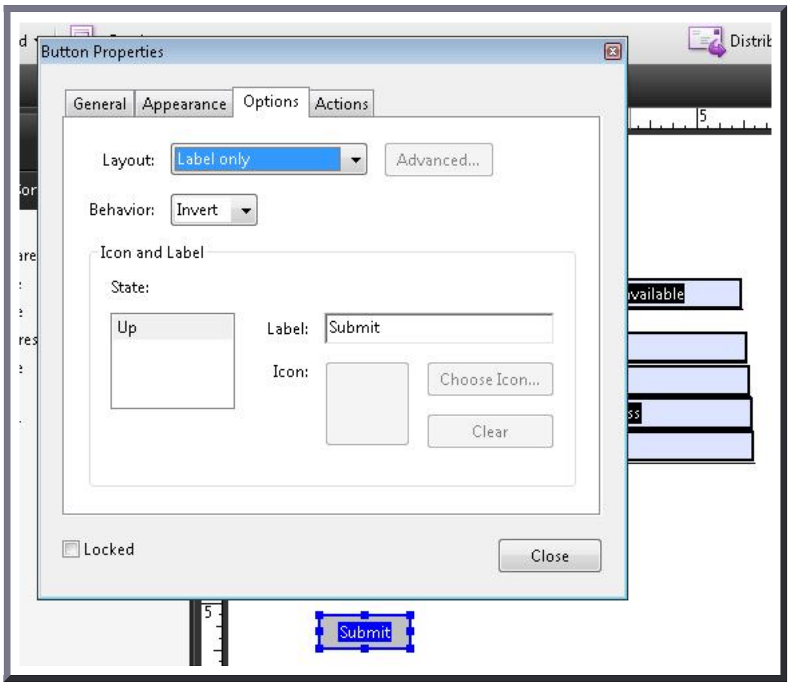
This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. From the toolbar, select Forms > Form Tools > Button and create a button on the form.
2. Access the context menu for the button and select Properties... to open the Button Properties dialog.
3. In the General tab, provide a tooltip for the button.
4. In the Options tab, choose an option in the Layout menu for the button label, icon image, or both. Then, type text in the Label box to identify the button as a submit button and/or click Choose Icon and locate the image file you want to use.
5. In the Actions tab:
   - For Select Trigger, choose Mouse Up. (The Mouse Up event is keyboard accessible and, in addition, ensures that the button will not change context unexpectedly, as it might with, e.g., a Mouse Enter event.)
   - For Select Action, choose Submit A Form.

- Click Add.

6. In the Add dialog, enter a URL to collect data on a server or collect form data as e-mail attachments.

The following image shows the Options tab on the Button Properties dialog.



Layout: Label only

Behavior: Invert

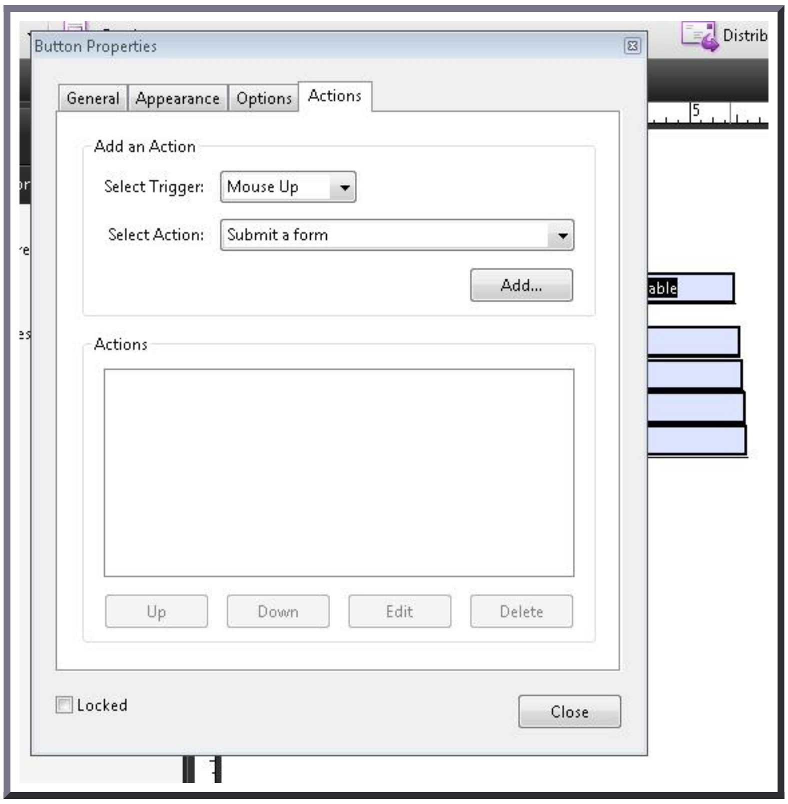
State: Up

Label: Submit

Icon:

Locked

The following image shows the Actions tab on the Button Properties dialog.



Select Trigger: Mouse Up

Select Action: Submit a form

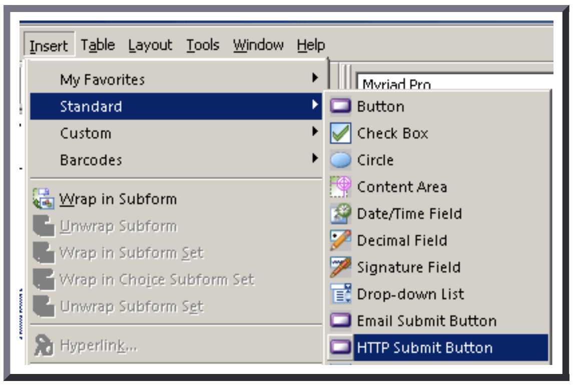
Locked

Example 2: Adding a submit button using Adobe LiveCycle Designer ES 8.2.1

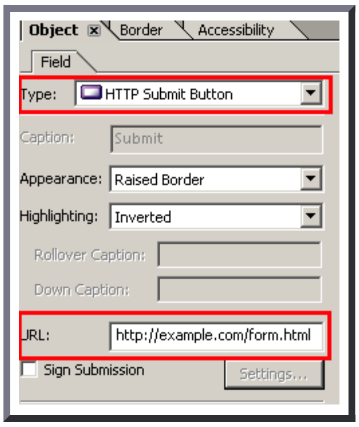
This example is shown with Adobe LiveCycle Designer. There are other software tools that perform similar functions. See the list of other software tools in [PDF Authoring Tools that Provide Accessibility Support](#).

1. On the Insert > Standard menu, select the HTTP Submit Button item.
2. On the Object panel for the HTTP Submit Button, insert the URL for form-submission processing.

The following image shows the Standard menu with the list of form controls.



The following image shows the Object panel with the URL and other fields for button appearance.



Type: HTTP Submit Button

Caption: Submit

Appearance: Raised Border

Highlighting: Inverted

Rollover Caption:

Down Caption:

URL: http://example.com/form.html

Sign Submission

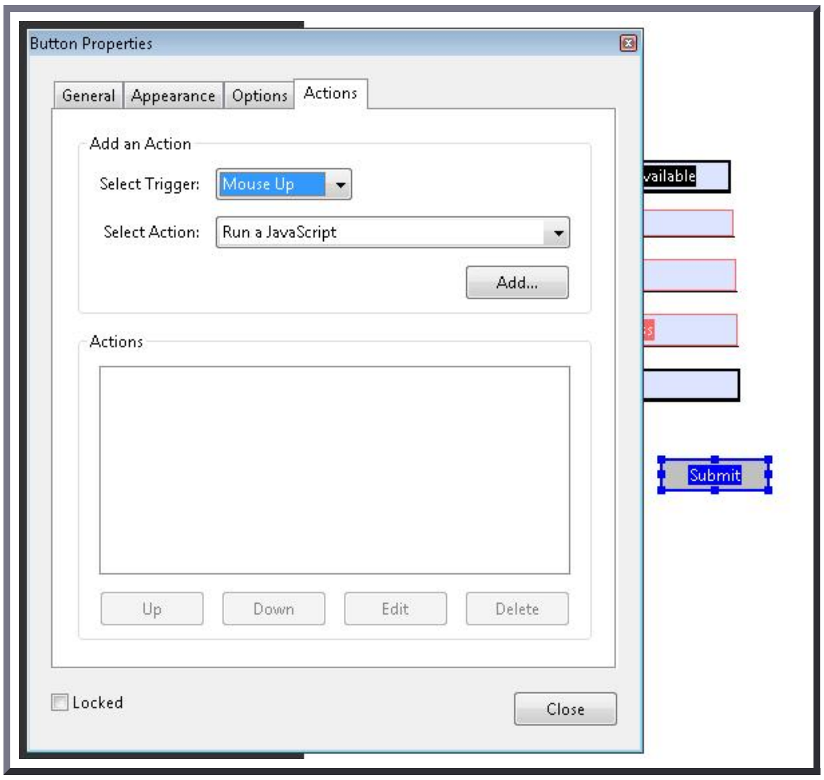
Example 3: Adding a script action to a submit button in a PDF document using JavaScript

The following JavaScript code illustrates the use of a script to specify the submit-form action. To add this script to the form field:

1. Open the Button Properties dialog, as shown in Example 1, and select the Actions tab
2. Select Run a JavaScript from the drop-down list, and select the Add button
3. Enter JavaScript code in the JavaScript Editor dialog, for example:

```
var aSubmitFields = new Array( "name", "id", "juser" );
this.submitForm({
  cURL: "http://www.example.com/cgi-bin/myscript.cgi#FDF",
  aFields: aSubmitFields,
  cSubmitAs: "FDF" // the default, not needed here
});
```

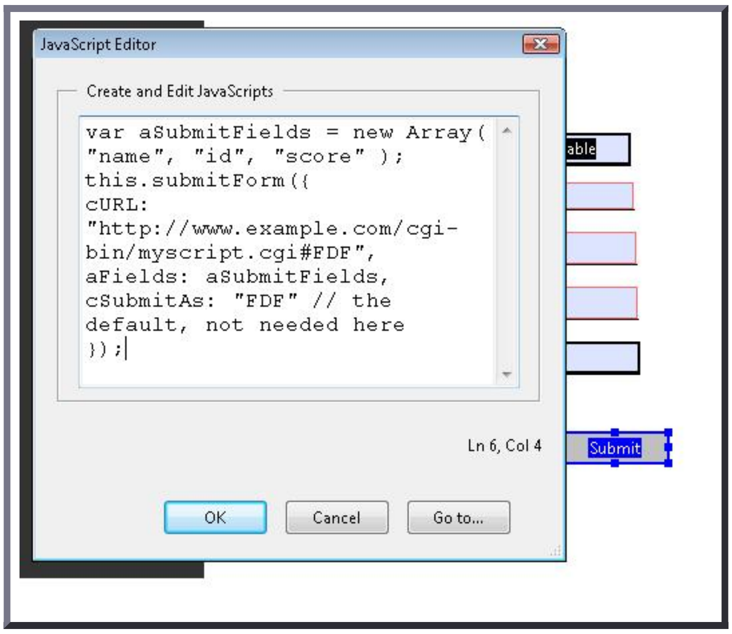
The following images illustrate this process:



Select Trigger: Mouse Up

Select Action: Run a JavaScript

Locked



```
var aSubmitFields = new Array(
"name", "id", "score" );
this.submitForm({
cURL:
"http://www.example.com/cgi-
bin/myscript.cgi#FDF",
aFields: aSubmitFields,
cSubmitAs: "FDF" // the
default, not needed here
});
```

This example is shown in operation in the working example of adding a script action to a submit button.

Resources

Resources are for information purposes only, no endorsement implied.

- Section 12.7.5.2 (Submit-Form Action) in PDF 1.7 (ISO 32000-1)
- Create submission forms in LiveCycle Designer
- XML Forms Architecture (XFA) Specification Version 2.5
- PDF and Accessibility

Related Techniques

- G80: Providing a submit button to initiate a change of context
- PDF23: Providing interactive form controls in PDF documents
- PDF12: Providing name, role, value information for form fields in PDF documents

Tests

Procedure

1. For each page that submits a form, visually verify that the form contains a submit button and check one of the following:
   - Tab to the button and check that it submits the form in response to user action to select the button.
   - Open the PDF document with a tool that is capable of showing the submit-form action and check that the button action is to submit the form.

Expected Results

- #1 is true for each page that contains a form.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF16: Setting the default language using the /Lang entry in the document catalog of a PDF document

Applicability

Tagged PDF documents

This technique relates to:

- Success Criterion 3.1.1 (Language of Page)
  - How to Meet 3.1.1 (Language of Page)
  - Understanding Success Criterion 3.1.1 (Language of Page)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF16. Also see PDF Technology Notes.

Description

The objective of this technique is to specify a document's default language by setting the /Lang entry in the document catalog. This is normally accomplished using a tool for authoring PDF.

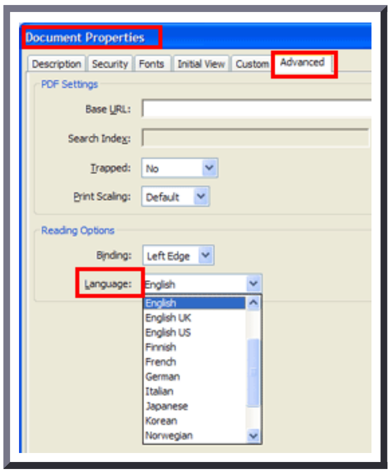
Both assistive technologies and conventional user agents can render text more accurately when the language of the document is identified. Screen readers can load the correct pronunciation rules. Visual browsers can display characters and scripts correctly. Media players can show captions correctly. As a result, users with disabilities are better able to understand the content.

Examples

Example 1: Adding a /Lang entry to specify the default document language using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. Open the document in Adobe Acrobat Pro
2. From the File menu, select "Properties..."
3. In the "Properties" dialog, select the "Advanced" tab
4. In the "Reading Options" field, select the default language from the "Language" combo box



Trapped: No
Print Scaling: Default
Binding: Left Edge
Language: English

Note: Acrobat includes 16 preset language selections. If you need to specify a language that is not on the list, such as Russian, you must type the ISO 639 code for the language, not its name.

This example is shown in operation in the working example of adding a /Lang entry using Acrobat Pro.

Example 2: Specifying the default document language in a PDF document using Microsoft Word 2007

This example is shown with Microsoft Word. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

Documents authored in Microsoft Word: "In some instances, even if the document language has been specified in the source file, the information about document language is not conveyed to the PDFMaker. Setting the language for an entire document in the Document Properties dialog box [see Example 1] corrects all errors related to this option."(Adobe® Acrobat® 9 Pro Accessibility Guide: Creating Accessible PDF from Microsoft® Word)

Example 3: Specifying the default document language in a PDF document using a /Lang entry

The natural language used for text in a document is determined in a hierarchical fashion, based on whether an optional /Lang entry is present in any of several possible locations. At the highest level, the document's default language may be specified by a /Lang entry in the document catalog.

The following code fragment illustrates code that is typical for using the /Lang entry in the document catalog for a document's default language (in this case, US English). (This is typically accomplished by an authoring tool.)

```
1 0 obj
  << /Type /Catalog
     ...
     /Lang (en-US)
     ...
  >>
endobj
```

Resources

Resources are for information purposes only, no endorsement implied.

- Section 14.9.2 (Natural Language Specification) in PDF 1.7 (ISO 32000-1)
- ISO 639-2 Codes
- PDF Reference 1.6, 10.8.1 Natural Language Specification (PDF 8.7 Mb)
- PDF Standards: Natural Language Specification
- Adobe® Acrobat® 9 Pro Accessibility Guide: Creating Accessible PDF from Microsoft® Word
- PDF and Accessibility

Related Techniques

- PDF19: Specifying the language for a passage or phrase with the Lang entry in PDF documents

Tests

Procedure

1. Verify that the default language for the document is correctly specified by applying one of the following:
   - Read the PDF document with a screen reader, listening to hear that the text is read in the correct natural language.
   - Using a PDF editor, check that the language is set to the default document language.
   - Use a tool which is capable of showing the /Lang entry value in the document catalog to open the PDF document and view the language settings.
   - Use a tool that exposes the document through the accessibility API and verify that the language is set to the default language.

Expected Results

- #1 is true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF17: Specifying consistent page numbering for PDF documents

Applicability

Tagged PDF documents

This technique relates to:

- Success Criterion 1.3.1 (Info and Relationships)
  - How to Meet 1.3.1 (Info and Relationships)
  - Understanding Success Criterion 1.3.1 (Info and Relationships)
- Success Criterion 2.4.8 (Location)
  - How to Meet 2.4.8 (Location)
  - Understanding Success Criterion 2.4.8 (Location)
- Success Criterion 3.2.3 (Consistent Navigation)
  - How to Meet 3.2.3 (Consistent Navigation)
  - Understanding Success Criterion 3.2.3 (Consistent Navigation)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF17. Also see PDF Technology Notes.

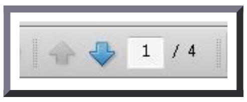
Description

The objective of this technique is to help users locate themselves in a document by ensuring that the page numbering displayed in the PDF viewer page controls has the same page numbering as the document. For example, Adobe Acrobat Pro and Reader display page numbers in the Page Navigation toolbar. The page number format is specified by the /PageLabels entry in the Document Catalog.

Many documents use specific page number formats within a document. Commonly, front matter is numbered with lowercase Roman numerals. The main content, starting on the page numbered 1, may actually be the fifth or sixth page in the document. Similarly, appendices may begin with page number 1 plus a prefix of the appendix letter (e.g., "A-1").

Authors should make sure that the page numbering of their converted documents is reflected in any page number displays in their user agent. Consistency in presenting the document's page numbers will help make navigating the document more predictable and understandable.

As an example, if /PageLabels has not been provided to describe the page number formatting, the page numbering scheme will not be reflected in the Page Navigation toolbar in Adobe Acrobat Pro or Reader. This toolbar displays the page number in a text box, which users can change to move to another page. In addition, users can select the arrows to move one page up or down in the document. The toolbar also displays the relative page number location. In the image below, the default display indicates the user is on page 1 of 4 pages.



A more direct way of going to a page is to use the shortcut for the View > Page Navigation > Page menu item. On Windows, this shortcut is "Ctrl + Shift + N"; on Mac OS, it is "Cmd + Shift + N". This brings up a dialog box to go to a specific page number.

Examples

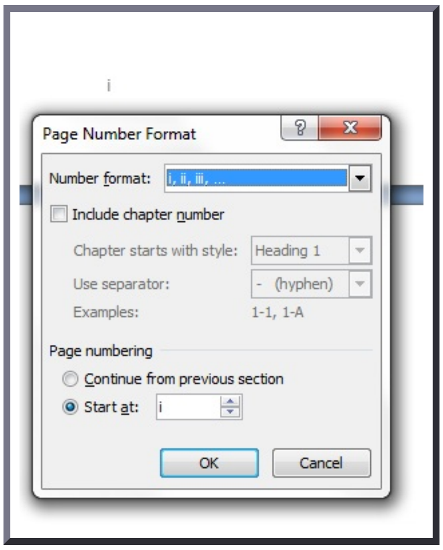
Example 1: Editing PDF page number formatting specifications using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

The example document converted from Microsoft Word 2007 has 4 pages, numbered i, ii, iii, 1. The image below shows the Word document with lowercase Roman numeral page numbering specified In Word using:

- Insert ribbon > Page number > Page Number Format

In this document, a new section has been created with page numbering beginning with Arabic numeral 1 on the fourth page of the document. The document was then converted to PDF from Word.



Page Number Format

Number format: i, ii, iii, ...

[ ] Include chapter number

Chapter starts with style: Heading 1

Use separator: - (hyphen)

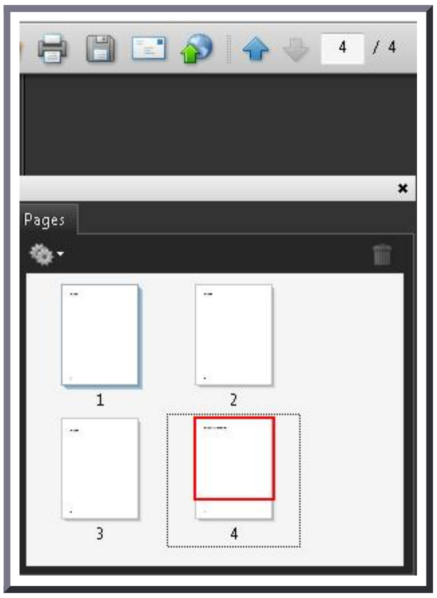
Examples: 1-1, 1-A

Page numbering

( ) Continue from previous section

(●) Start at: i

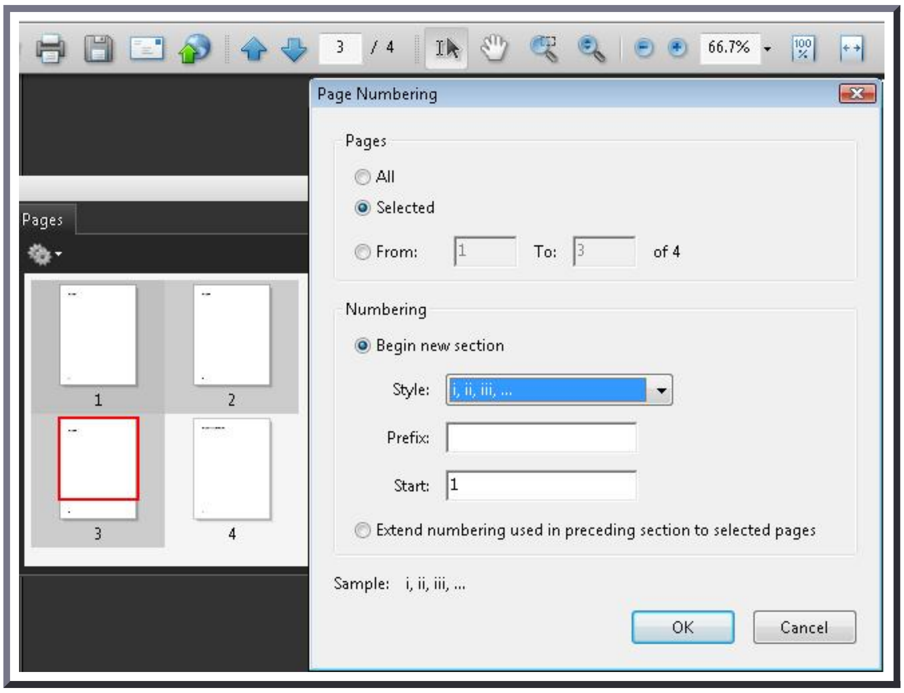
In Adobe Acrobat Pro, Select View > Navigation Panels > Pages. The following image shows the page thumbnails in the Pages panel and the Page Navigation toolbar. Both the thumbnails and the toolbar use Arabic page numbers.



To correct the page numbers:

1. Select the pages to be renumbered
2. Access the context menu for the selected pages and select Number Pages
3. In the Page Numbering dialog, select the lowercase Roman numeral style and the starting page (1 by default, which is correct in this case)
4. Select OK

The following image shows the Page Numbering dialog and selections.



Page Numbering

Pages
- All
- [✓ Selected]
- From: 1 To: 3 of 4

Numbering
- [✓ Begin new section]

Style: i, ii, iii, ...
Prefix:
Start: 1

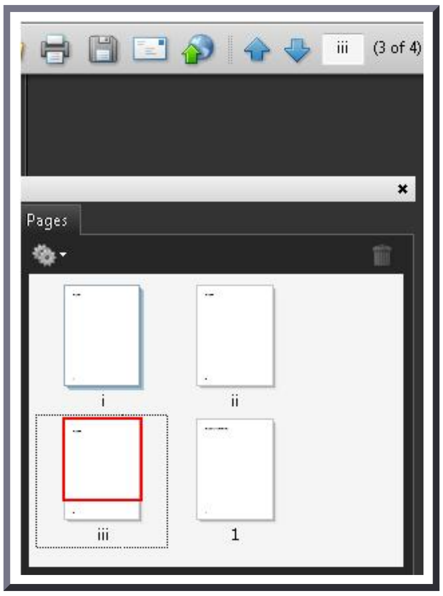
- Extend numbering used in preceding section to selected pages

Sample: i, ii, iii, ...

OK Cancel

Follow the same process to change the fourth page number to Arabic numeral 1.

The following image shows the correct page numbers for the 4 pages. Note that page iii is selected in the Pages panel and the Page Navigation toolbar shows iii in the text area. In addition, the relative location in the document is shown at the right of the toolbar: "(3 of 4)."



This example is shown in operation in the working example of specifying page numbers in a document converted from Word (Word file) and working example of specifying page numbers in a document converted from Word (PDF file).

Example 2: Specifying page numbers using the /PageLabels entry

The following code fragment illustrates code that is typical for specifying multiple page numbering schemes in a document.

The example below is for a document with pages labeled:

Example: i, ii, iii, iv, 1, 2, 3, A-8, A-9, …

This numbering scheme requires 3 page-label dictionaries (for lowercase Roman, Arabic, and prefixed numbers)

```
1 0 obj
    << /Type /Catalog
       /PageLabels << /Nums [ 0 << /S /r >>  % lowercase Roman numerals
                              4 << /S /D >>  % Arabic numerals
                              7 << /S /D     % Arabic numerals with ...
                      /P (A-)                % the prefix "A-"...
                      /St 8                  % starting at page 8
                                >>
                          ]
                     >>
          …
   >>
  endobj
```

Page labels are specified as follows:

- /S specifies the numbering style for page numbers:
  - /D - Arabic numerals (1,2,3...)
  - /r - lowercase Roman numerals (i, ii, iii,...)
  - /R - uppercase Roman numerals (I, II, III,...)
  - /A - uppercase letters (A-Z)
  - /a - lowercase letters (a-z)
- /P (optional) - page number prefix
- /St (optional) - the value of the first page number in the range (default: 1)

Resources

Resources are for information purposes only, no endorsement implied.

- Section 12.4.2 (Page Labels) PDF 1.7 (ISO 32000-1)
- PDF and Accessibility

Related Techniques

- PDF14: Providing running headers and footers in PDF documents

Tests

Procedure

1. For every section in the document that uses a different pagination format, check that the page navigation feature uses the same format used on the document pages:
   - Select the pages that begin a new pagination format and visually verify that the same format and page number is shown in the page navigation feature.
   - Using a screen reader, check that the page number announced in the page navigation feature is the same as the page number announced on the document page.
   - Using a tool that is capable of showing the /PageLabels entries, open the PDF document and view the entries.
   - Use a tool that exposes the document through the accessibility API, and verify that the /PageLabels entries are specified correctly.

Expected Results

- #1 is true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF18: Specifying the document title using the Title entry in the document information dictionary of a PDF document

Applicability

Tagged PDF documents

This technique relates to:

- Success Criterion 2.4.2 (Page Titled)
  - How to Meet 2.4.2 (Page Titled)
  - Understanding Success Criterion 2.4.2 (Page Titled)

  Note: This technique must be combined with other techniques to meet SC 2.4.2. See Understanding SC 2.4.2 for details.

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF18. Also see PDF Technology Notes.

Description

The intent of this technique is to show how a descriptive title for a PDF document can be specified for assistive technology by using the /Title entry in the document information dictionary and by setting the DisplayDocTitle flag to True in a viewer preferences dictionary. This is typically accomplished by using a tool for authoring PDF.

Document titles identify the current location without requiring users to read or interpret page content. User agents make the title of the page easily available to the user for identifying the page. For instance, a user agent may display the page title in the window title bar or as the name of the tab containing the page.

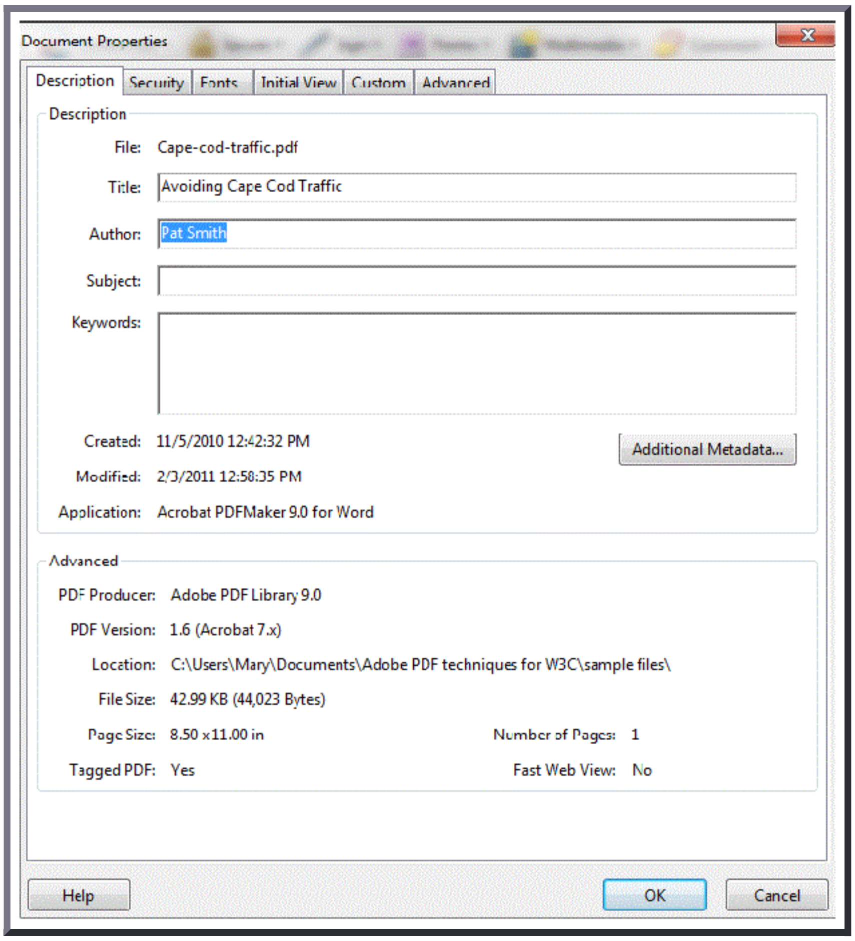
Examples

Example 1: Setting the document title in the metadata and specifying that the title be displayed in the title bar using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

Open the PDF document in Adobe Acrobat Pro:

1. Select File > Properties
2. Select the Description tab to view the metadata in the document, including the document information dictionary
3. Modify the Title field to add or change the document's Title entry



Description

File: Cape-cod-traffic.pdf

Title: Avoiding Cape Cod Traffic

Author: Pat Smith

Subject:

Keywords:

Created: 11/5/2010 12:42:32 PM

Modified: 2/3/2011 12:58:35 PM

Application: Acrobat PDFMaker 9.0 for Word

Advanced

PDF Producer: Adobe PDF Library 9.0

PDF Version: 1.6 (Acrobat 7.x)

Location: C:\Users\Mary\Documents\Adobe PDF techniques for W3C\sample files\

File Size: 42.99 KB (44,023 Bytes)

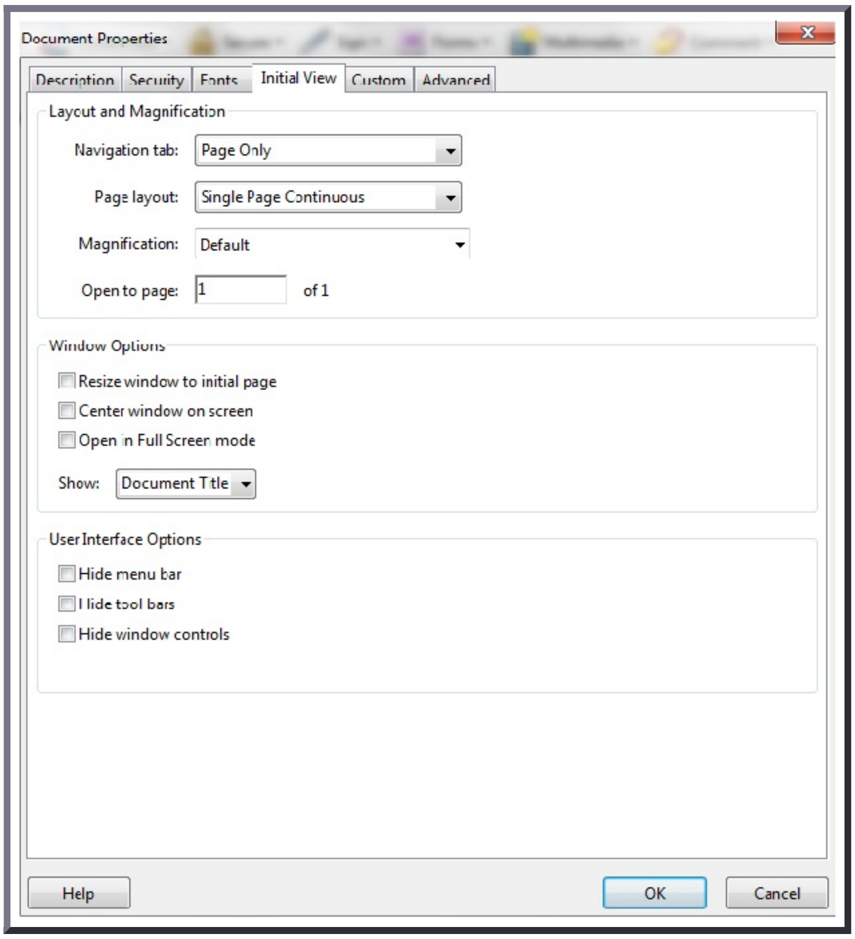
Page Size: 8.50 x 11.00 in Number of Pages: 1

Tagged PDF: Yes Fast Web View: No

Note that, with Adobe Acrobat installed, you can also enter and read the data properties information from the desktop. Access the file's context menu, choose Properties, and select the PDF tab. Any information you type or edit in this dialog box also appears in the Document Properties Description when you open the file.

To display the document title in the title bar of a user agent:

1. Select File > Properties
2. Select the Initial View tab
3. In the Window Options section, select Document Title in the Show pull-down list.


Document Properties

Description | Security | Fonts | Initial View | Custom | Advanced

Layout and Magnification

Navigation tab: Page Only

Page layout: Single Page Continuous

Magnification: Default

Open to page: 1 of 1

Window Options

- [ ] Resize window to initial page
- [ ] Center window on screen
- [ ] Open in Full Screen mode

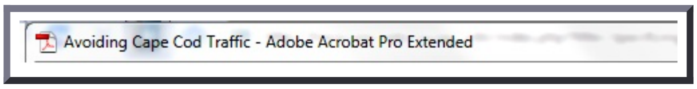
Show: Document Title

User Interface Options

- [ ] Hide menu bar
- [ ] Hide tool bars
- [ ] Hide window controls

Help | OK | Cancel

The title is displayed in the title bar, as shown in the image below.


Avoiding Cape Cod Traffic - Adobe Acrobat Pro Extended

This example is shown in operation in the working example of displaying document title in the title bar.

Example 2: A /Title entry in the document information dictionary of a PDF document

The following code fragment illustrates code that is typical for providing a /Title entry in a document information dictionary that contains a document title.

```
1 0 obj
   << /Title (Applying Guerrilla Tactics to Usability Testing by People with Disabilities)
      /Author (Mary Smith)
      /CreationDate (D:19970915110347-08'00')
   >>
endobj
```

Resources

Resources are for information purposes only, no endorsement implied.

- PDF and Accessibility
- Section 14.3.3 (Document Information Dictionary) in PDF 1.7 (ISO 32000-1)
- PDF Reference 1.6, TITLE entry of the document information dictionary

Related Techniques

- G88: Providing descriptive titles for Web pages

Tests

Procedure

1. Verify that the title for the document is correctly specified and displayed in the user agent title bar by applying one of the following:
   - Open the PDF document with a screen reader, listening to hear that the document title is read correctly.
   - Using a PDF editor, check that the document title is specified. Select the Initial View tab to check that the title will be displayed.
   - Use a tool which is capable of showing the /Title entry value in the document catalog to open the PDF document and view the /Title entry and /DisplayDocTitle flag settings.

Expected Results

- #1 is true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF19: Specifying the language for a passage or phrase with the Lang entry in PDF documents

Applicability

Tagged PDF documents

This technique relates to:

- Success Criterion 3.1.1 (Language of Page)
    - How to Meet 3.1.1 (Language of Page)
    - Understanding Success Criterion 3.1.1 (Language of Page)
- Success Criterion 3.1.2 (Language of Parts)
    - How to Meet 3.1.2 (Language of Parts)
    - Understanding Success Criterion 3.1.2 (Language of Parts)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF19. Also see PDF Technology Notes.

Description

The objective of this technique is to specify the language of a passage, phrase, or word using the /Lang entry to provide information in the PDF document that user agents need to present text and other linguistic content correctly. This is normally accomplished using a tool for authoring PDF.

Both assistive technologies and conventional user agents can render text more accurately when the language is identified. Screen readers can load the correct pronunciation rules. As a result, users with disabilities are better able to understand the content.

Note: This technique can be used to set the default language for the entire document if the entire document is contained in the container or tag. In this case, this technique would apply to Success Criterion 3.1.1.

Examples

Example 1: Adding a /Lang entry to specify the language for a paragraph using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. In the Tools menu, select Advanced Editing.
2. Select the TouchUp Reading Order Tool.
3. Click the Show Order Panel button in the TouchUp Reading Order Tool
4. Select the Tags tab in the Show Order Panel and select the paragraph that is in the different language. You can also use the Options menu in the Tags tab: select Find Tag from Selection.
5. Right-click the selection and select Properties in the context menu.
6. In the Tags tab in the Properties dialog, select the language from the drop-down list.

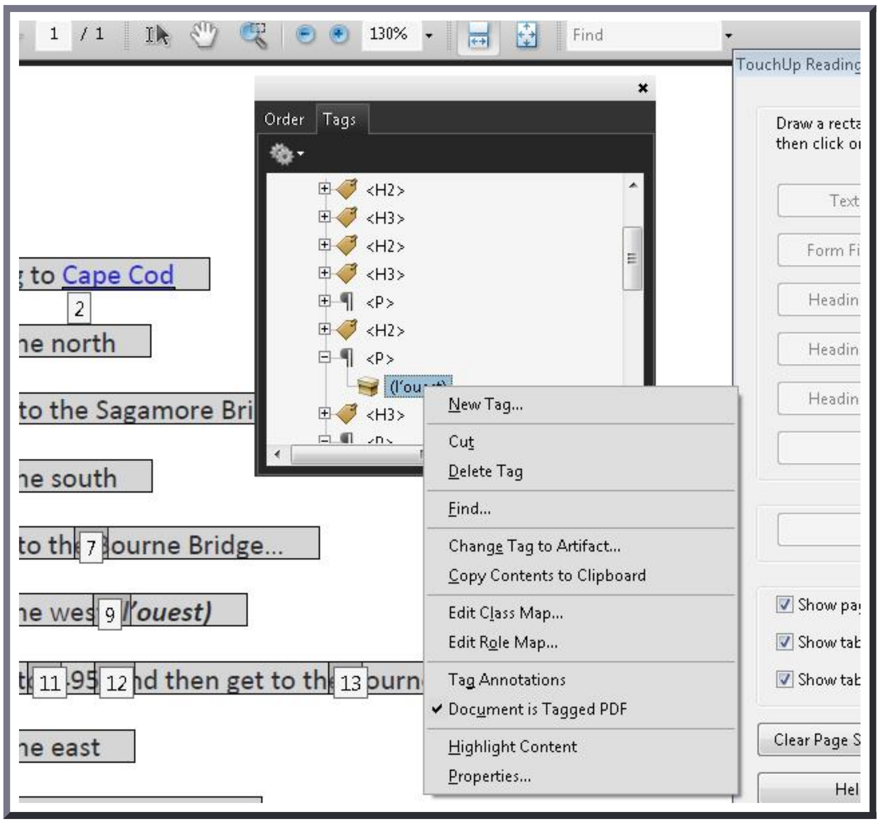
Note: Acrobat includes 16 preset language selections. If you need to specify a language that is not on the list, such as Russian, you must type the ISO 639 code for the language, not its name.

Example 2: Adding a /Lang entry to specify the language for a specific word or phrase using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. Select the word or phrase that is in a different language and create a tag for it in the Reading Order Panel (e.g., Text).
2. Open the Tags tab in the Show Order Panel and select the tagged word or phrase that is in the different language. You can also use the Options menu in the Tags tab: select Create Tag from Selection.
3. Right-click the selection and select Properties in the context menu.
4. In the Tags tab in the Properties dialog, select the language from the drop-down list.

When you tag a word or phrase, Acrobat splits the original content into three document content tags: one for the text that precedes your selection, one for the selection, and one for the text that follows the selection. As needed, drag the document content tag for the selected text into position between the other two tags, so that the text reads in the proper order. All three tags must also be at the same level beneath their parent tag. Drag them into place if they are not.



This example is shown in operation in the working example of marking a specific word or phrase in Acrobat Pro.

Example 3: Specifying the language for a word or phrase in a PDF document using a /Lang entry

Below the level of the default document language, the language for a passage may be specified for the following items:

- Marked-content sequences that are not in the structure hierarchy, through a /Lang entry in a property list attached to the marked-content sequence with a Span tag.
- Structure elements of any type, through a /Lang entry in the structure element dictionary.

The following code fragment illustrates code that is typical for using the /Lang entry to override the default document language by specifying a marked-content sequence within a page's content stream:

```
/P % Start of marked-content sequence
BDC
   (See you later, or in Spanish you would say, ) Tj
   /Span << /Lang (es-MX) >>% Start of nested marked-content sequence
  BDC
   (Hasta la vista.) Tj
  EMC% End of nested marked-content sequence
EMC% End of marked-content sequence
```

The following code fragment illustrates code that is typical for using the /Lang entry in the structure element dictionary. In this case, the /Lang entry applies to the marked-content sequence having an MCID (marked-content identifier) value of 0 within the indicated page's content stream.

```
1 0 obj% Structure element
  << /Type /StructElem
    /S /Span% Structure type
    /P /P% Parent in structure hierarchy
    /K<< /Type /MCR
      /Pg 2 0 R% Page containing marked-content sequence
      /MCID 0% Marked-content identifier
     >>
   /Lang (es-MX)% Language specification for this element
   >>
endobj
2 0 obj% Page object
  << /Type /Page
     /Contents 3 0 R% Content stream
   …
   >>
   endobj
3 0 obj% Page's content stream
  << /Length … >>
    stream
     BT
      /P % Start of marked-content sequence
      BDC
     (See you later, or in Spanish you would say, ) Tj
     /Span << /MCID 0 >>% Start of nested marked-content sequence
    BDC
     (Hasta la vista.) Tj
    EMC% End of nested marked-content sequence
```

```
 EMC% End of marked-content sequence
ET
endstream
endobj
```

Resources

Resources are for information purposes only, no endorsement implied.

- Section 14.9.2 (Natural Language Specification) in PDF 1.7 (ISO 32000-1)
- ISO 639-2 Codes
- PDF Reference 1.6, 10.8.1 Natural Language Specification (PDF 8.7 Mb)
- PDF Standards: Natural Language Specification
- Adobe® Acrobat® 9 Pro Accessibility Guide: Creating Accessible PDF from Microsoft® Word
- PDF and Accessibility

Related Techniques

- PDF16: Setting the default language using the /Lang entry in the document catalog of a PDF document

Tests

Procedure

1. Verify that the language of a passage, phrase, or word that differs from the language of the surrounding text is correctly specified by a /Lang entry on an enclosing tag or container:
   - Read the PDF document with a screen reader that supports the language of the phrase and the language of the surrounding text, listening to hear that the text is read in the correct natural language.
   - Using a PDF editor, select the word or phrase that is in the different language and check that the language is set correctly.
   - Use a tool which is capable of showing the /Lang entry value to open the PDF document and view the language settings.
   - Use a tool that exposes the document through the accessibility API and verify that the language for the passage or phrase is set correctly.
2. Verify that if the container or tag contains the entire document, the language setting is the language intended as the default for the document.

Expected Results

- #1 and #2 are true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF20: Using Adobe Acrobat Pro's Table Editor to repair mistagged tables

Applicability

Tagged PDF documents with tables.

This technique relates to:

- Success Criterion 1.3.1 (Info and Relationships)
  - How to Meet 1.3.1 (Info and Relationships)
  - Understanding Success Criterion 1.3.1 (Info and Relationships)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF20. Also see PDF Technology Notes.

Description

The purpose of this technique is to show how table cells in PDF documents can be marked up so that the logical relationships among rows and columns are preserved and recognized by assistive technology. This is typically accomplished by using a tool for authoring PDF.

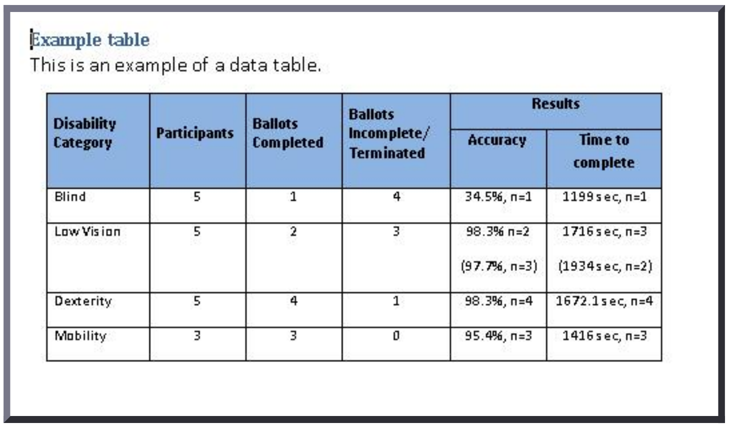
However, tables converted to PDF may have incorrectly merged or split table cells, even if they were marked up correctly in the authoring tool. Authors can ensure that table cells are structured properly by using the Table Editor in Adobe Acrobat Pro's TouchUp Reading Order tool.

Examples

Example 1: Repairing table cells using the Table Editor in the TouchUp Reading Order tool in Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

This example uses a table that was marked up correctly when it was created in Microsoft Word. Some table headers span two rows in the header row; one table header spans two columns.



Example table

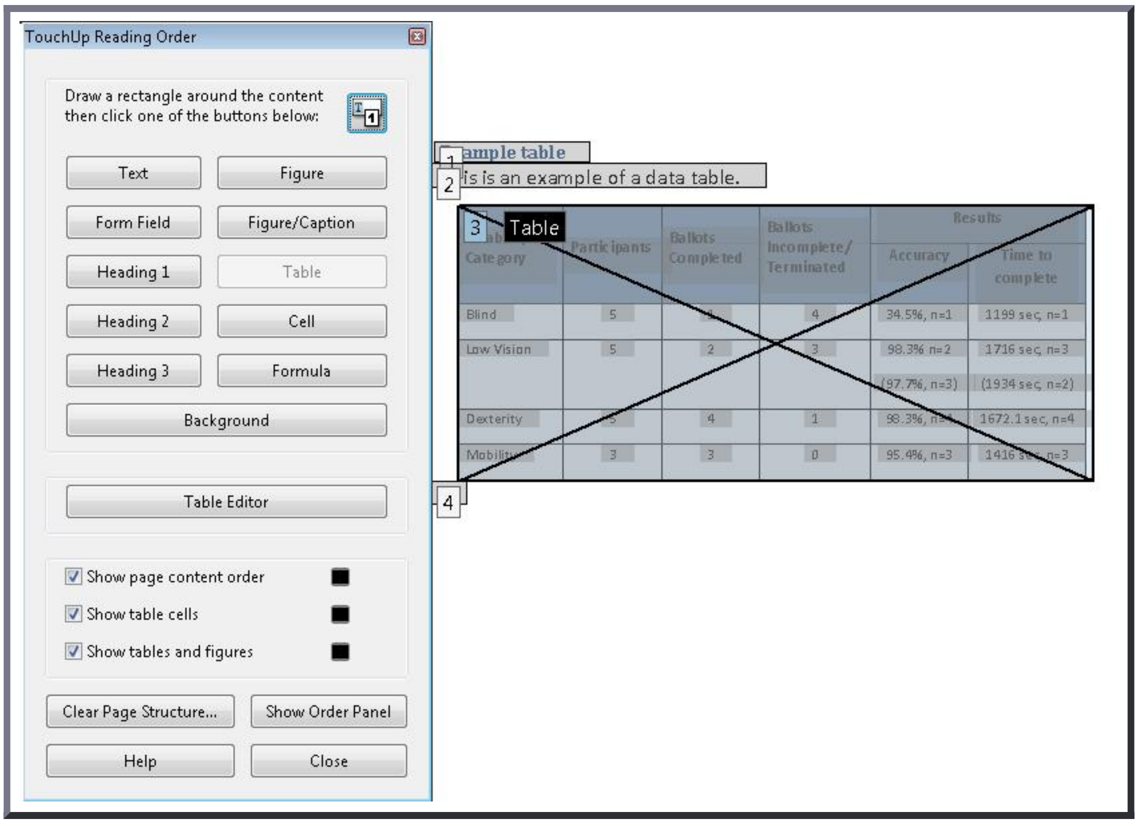
This is an example of a data table.

Disability Category	Participants	Ballots Completed	Ballots Incomplete/ Terminated	Results	
				Accuracy	Time to complete
Blind	5	1	4	34.5%, n=1	1199 sec, n=1
Low Vision	5	2	3	98.3% n=2 (97.7%, n=3)	1716 sec, n=3 (1934 sec, n=2)
Dexterity	5	4	1	98.3%, n=4	1672.1 sec, n=4
Mobility	3	3	0	95.4%, n=3	1416 sec, n=3

To check the table in the PDF document:

1. Advanced > Accessibility > TouchUp Reading Order...
2. Select the table by clicking the number in the top left hand corner of the table (3 in the reading order in the image below).
3. Select the Table Editor button on the TouchUp Reading Order panel. The table cells will be outlined in red and labeled with their tags. The red outlines may not exactly match up to the table cells but you should be able to determine if the cells are tagged correctly.

The following image shows the example table in the TouchUp Reading Order tool. Note that the Results header appears to span two sub-headers and the other headers to the left span the two rows in the Results header.



TouchUp Reading Order

Draw a rectangle around the content then click one of the buttons below:

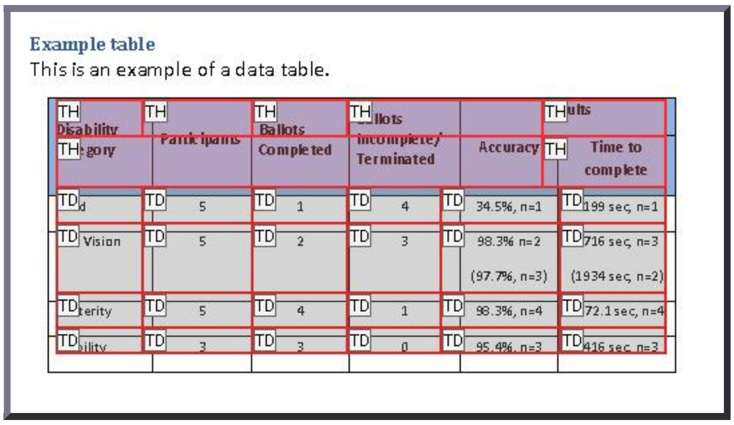
- [✓] Show page content order
- [✓] Show table cells
- [✓] Show tables and figures

ample table

is is an example of a data table.

Disability Category	Participants	Ballots Completed	Ballots Incomplete/ Terminated	Results	
				Accuracy	Time to complete
Blind	5	1	4	34.5%, n=1	1199 sec, n=1
Low Vision	5	2	3	98.3% n=2 (97.7%, n=3)	1716 sec, n=3 (1934 sec, n=2)
Dexterity	5	4	1	98.3%, n=4	1672.1 sec, n=4
Mobility	3	3	0	95.4%, n=3	1416 sec, n=3

The following images shows the example table in the Table Editor. The cells are outlined in red, and the tab for each cell is displayed. Upon conversion, the Results header was incorrectly split and does not span its two sub-headers. The headers to the right were incorrectly split into 2 cells each and do not span the Results headers. In addition, the incorrectly split cells were merged into one cell.



Example table

This is an example of a data table.

TH Disability	TH Participants	TH Ballots	TH llots		TH ults
TH gory		Completed	Incomplete/ Terminated	Accuracy	TH Time to complete
TD d	TD 5	TD 1	TD 4	TD 34.5%, n=1	TD 199 sec, n=1
TD Vision	TD 5	TD 2	TD 3	TD 98.3% n=2 (97.7%, n=3)	TD 716 sec, n=3 (1934 sec, n=2)
TD terity	TD 5	TD 4	TD 1	TD 98.3%, n=4	TD 72.1 sec, n=4
TD bility	TD 3	TD 3	TD 0	TD 95.4%, n=3	TD 416 sec, n=3

To repair the Results header:

1. Select the header in the table (it will be outlined in blue when selected)
2. Access the context menu
3. Select Table Cell Properties...
4. In the Table Cell Properties dialog, change the Column Span from 1 to 2
5. Press OK. You'll get a warning that the change might result in a malformed table structure. In this case, the change is correct. The cell you changed should change color to show the new span, as shown in the following image.

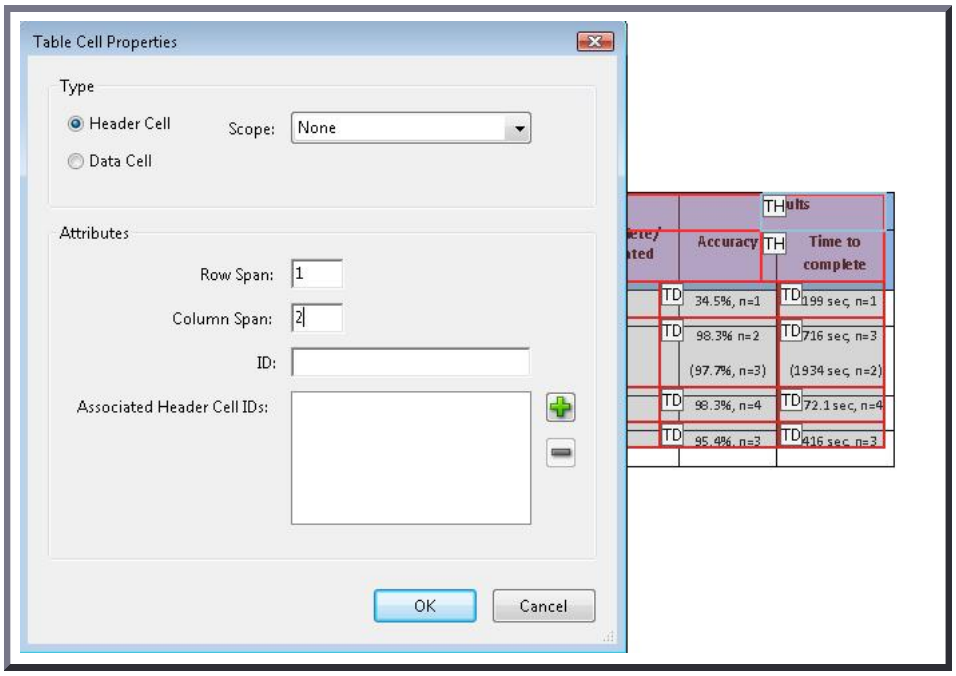


Table Cell Properties

Type: Header Cell [✓] / Data Cell — Scope: None

Attributes

Row Span: 1

Column Span: 2

ID:

Associated Header Cell IDs:

Similarly, to repair the incorrectly split header cells to the left of Results header:

1. Select the top cell in the column (it will be outlined in blue when selected)
2. Access the context menu
3. Select Table Cell Properties...
4. In the Table Cell Properties dialog, change the Row Span from 1 to 2
5. Press OK. The following image shows the correction being made to the last header cell, with the corrected header cells to its left.

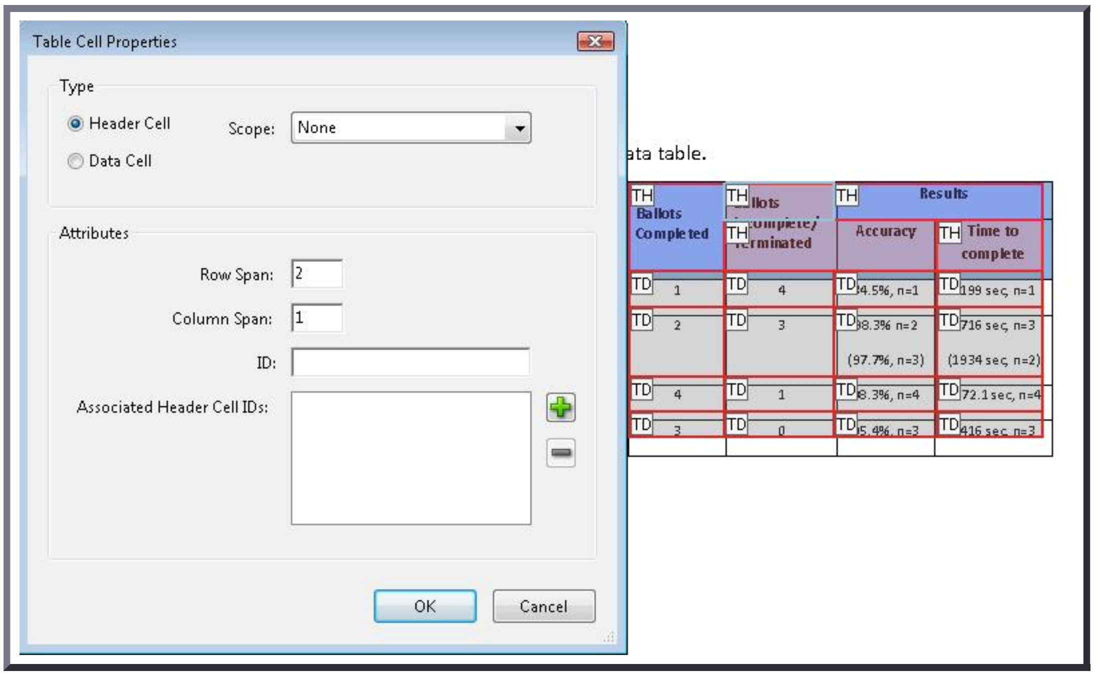


Table Cell Properties

Type: Header Cell [✓] / Data Cell — Scope: None

Attributes

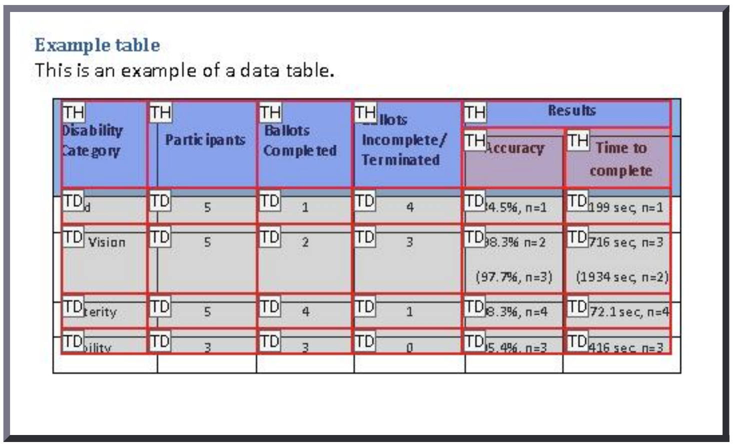
Row Span: 2

Column Span: 1

ID:

Associated Header Cell IDs:

The following image shows the repaired example table.



Example table

This is an example of a data table.

TH Disability Category	TH Participants	TH Ballots Completed	TH Ballots Incomplete/ Terminated	TH Results	
				TH Accuracy	TH Time to complete
TD d	TD 5	TD 1	TD 4	TD 4.5%, n=1	TD 199 sec, n=1
TD Vision	TD 5	TD 2	TD 3	TD 98.3% n=2 (97.7%, n=3)	TD 716 sec, n=3 (1934 sec, n=2)
TD terity	TD 5	TD 4	TD 1	TD 8.3%, n=4	TD 72.1 sec, n=4
TD bility	TD 3	TD 3	TD 0	TD 5.4%, n=3	TD 416 sec, n=3

This example is shown in operation in the working example of repairing table structure (Word file) and working example of repairing table structure (PDF file).

Example 2: Marking up a table using table structure elements

The following code fragment illustrates code that is typical for a simple table (header row and data row) such as shown in Examples 1-3:

```
95 0 obj                 %Structure element for a table
 <<
  /A 39 0 R
  /K[96 0 R 101 0 R 106 0 R 111 0 R]
  /P 93 0 R
  /S/Table               %standard structure type is table
 >>
 endobj
96 0 obj                 %Structure element for a table row
 <<
  /K[97 0 R 98 0 R 99 0 R 100 0 R]
  /P 95 0 R
  /S/TR                  %standard structure type is table row
 >>
 endobj
97 0 obj                 %Structure element for a table header
 <</A[23 0 R 120 0 R]
   /K 1
   /P 96 0 R
   /S/TH                  %standard structure type is table head
   /Pg 8 0 R
 >>
endobj
104 0 obj                 %Structure element for table data (cell contents)
 <<
  /A 29 0 R
  /K 7
  /P 101 0 R
  /S/TD                   %standard structure type is table data
  /Pg 8 0 R
 >>
endobj
```

Resources

Resources are for information purposes only, no endorsement implied.

- PDF and Accessibility
- 14.8.4.3.4 (Table Elements) in PDF 1.7 (ISO 32000-1)

Related Techniques

- H51: Using table markup to present tabular information
- PDF6: Using table elements for table markup in PDF Documents

Tests

Procedure

1. For a table that has been repaired with the Table Editor, confirm one of the following:
   - Read the PDF document with a screen reader, listening to hear that the tabular information is presented in a way that preserves logical relationships among the table header and data cells. (Configure the screen reader to not use heuristics to read table header cells.)
   - Using a PDF editor, verify that the appropriate TR, TH, and TD tags are in the proper reading order and hierarchy in the table tree.
   - Use a tool which is capable of showing the table elements to open the PDF document, view the table structure, and verify that it contains the appropriate TR, TH, and TD structures.
   - Use a tool that exposes the document through the accessibility API, and verify that the table structure contains the appropriate TR, TH, and TD structures, and that they are in the proper reading order and hierarchy.

Expected Results

- #1 is true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF21: Using List tags for lists in PDF documents

Applicability

Tagged PDF documents with lists.

This technique relates to:

- Success Criterion 1.3.1 (Info and Relationships)
  - How to Meet 1.3.1 (Info and Relationships)
  - Understanding Success Criterion 1.3.1 (Info and Relationships)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF21. Also see PDF Technology Notes.

Description

The intent of this technique is to create lists of related items using list elements appropriate for their purposes. PDF files containing lists are normally created or repaired using a tool for authoring PDF.

When markup is used that visually formats items as a list but does not indicate the list relationship, users may have difficulty navigating the information. An example of such visual formatting is simply using line-breaks to separate list items.

Some assistive technologies allow users to navigate from list to list or item to item. If the lists are not correctly formatted with list tags, these users will have difficulty understanding the list content.

The easiest way to create lists in PDF content is to format them properly using list markup in the authoring tool, for example, Microsoft Word or OpenOffice.org Writer. However, if you do not have access to the source file and authoring tool, you can use Acrobat Pro's TouchUp Reading Order tool and the Tags panel.

The PDF specification defines list structure in section 14.8.4.3.3 (List Elements). The structure types for lists in PDF documents are:

- L - the List tag, which contains one or more LI tags.
- LI - the List Item tag. List item tags can contain Lbl and LBody tags.
- Lbl - the list item label. Contains distinguishing information such as a item number or bullet character.
- LBody - the list item body. Contains list item content, or in the case of a nested list, it may contain additional List tag trees.

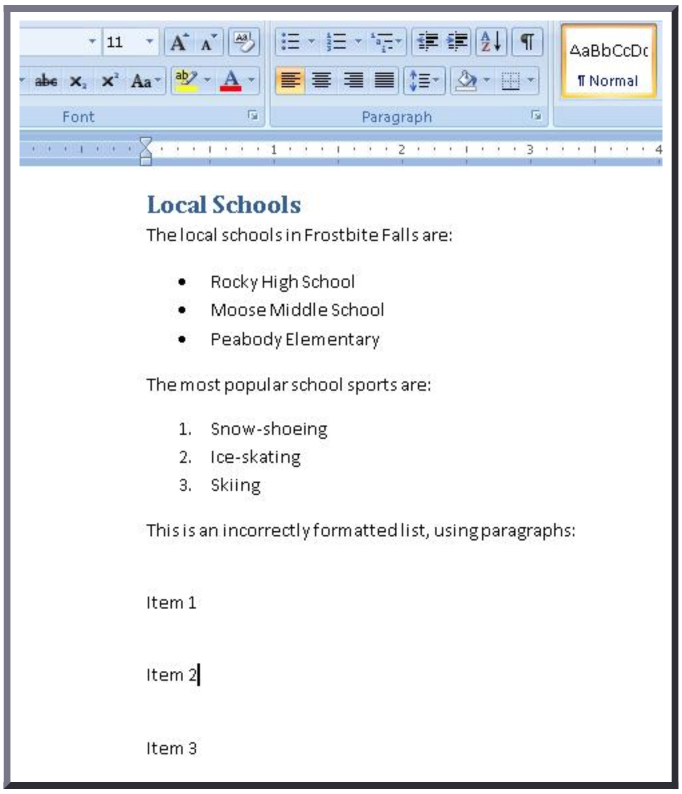
Examples

Example 1: Adding lists to Microsoft Word 2007 documents

This example is shown with Microsoft Word. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

On the Home ribbon, use the lists tools to create or repair lists in Word documents. This is the easiest way to ensure that lists are formatted correctly when they are converted to PDF.

In the image below, the numbered and bullet lists were created using the list tools. The third list did not use the list tool (see the ribbon) and the list will not be tagged as list elements when converted to PDF.



Local Schools

The local schools in Frostbite Falls are:

- Rocky High School
- Moose Middle School
- Peabody Elementary

The most popular school sports are:

1. Snow-shoeing
2. Ice-skating
3. Skiing

This is an incorrectly formatted list, using paragraphs:

Item 1

Item 2

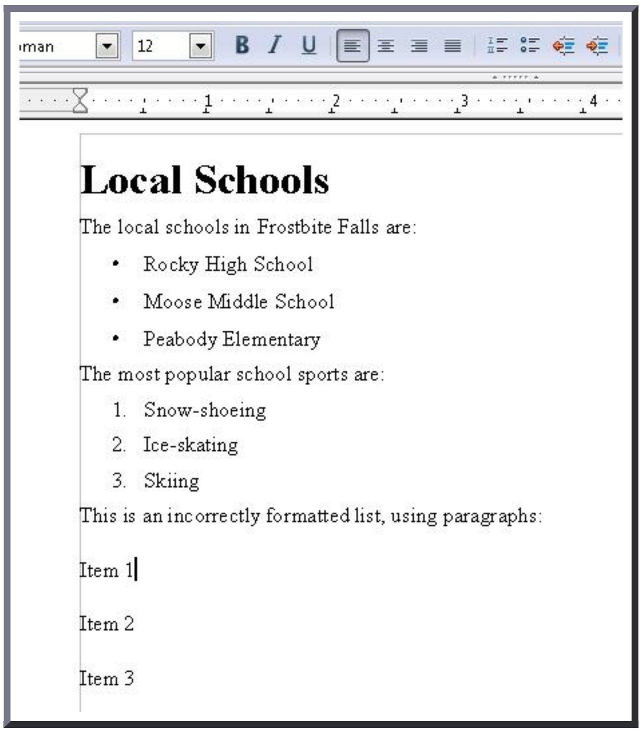
Item 3

Example 2: Adding lists to OpenOffice.org Writer 2.2 documents

This example is shown with OpenOffice.org Writer. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

Use the Bullets and Numbering tool to create or repair lists in OpenOffice.org Writer documents. This is the easiest way to ensure that lists are formatted correctly when they are converted to PDF.

In the image below, the numbered and bullet lists were created using the list tools. The third list did not use the list tool (see the toolbar) and the list will not be tagged as list elements when converted to PDF.



Local Schools

The local schools in Frostbite Falls are:

- Rocky High School
- Moose Middle School
- Peabody Elementary

The most popular school sports are:

1. Snow-shoeing
2. Ice-skating
3. Skiing

This is an incorrectly formatted list, using paragraphs:

Item 1

Item 2

Item 3

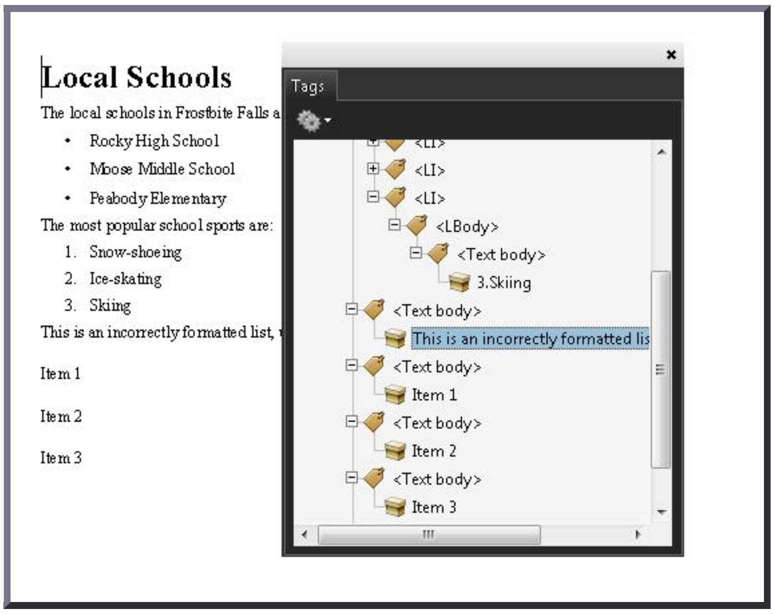
This example is shown in operation in the working example of adding lists to OpenOffice Writer documents.

Example 3: Ensuring that lists are correctly formatted using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. View > Navigation Panels... > Tags
2. Inspect the lists in the document to determine which, if any, are not formatted properly.

In the following image, the third list is formatted as text. The list items are separated only by line-breaks. Assistive technology may not be able to render the list intelligibly for users.



Local Schools

The local schools in Frostbite Falls a

- Rocky High School
- Moose Middle School
- Peabody Elementary

The most popular school sports are:

1. Snow-shoeing
2. Ice-skating
3. Skiing

This is an incorrectly formatted list,

Item 1

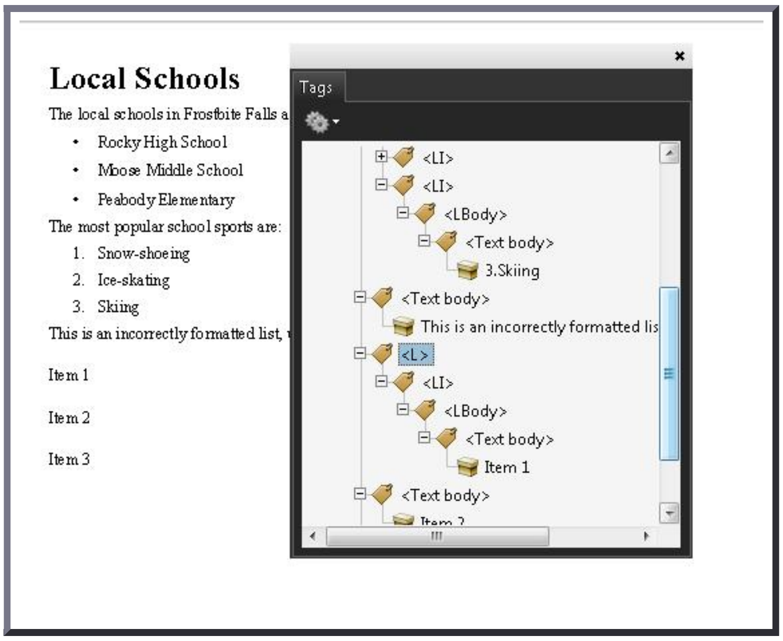
Item 2

Item 3

```
Tags
<LI>
<LI>
<LI>
  <LBody>
    <Text body>
      3.Skiing
<Text body>
  This is an incorrectly formatted lis
<Text body>
  Item 1
<Text body>
  Item 2
<Text body>
  Item 3
```

To repair the list, use the Tags panel to create list tags in the content.

The following image shows the resulting first list item correctly formatted.



Local Schools

The local schools in Frostbite Falls a

- Rocky High School
- Moose Middle School
- Peabody Elementary

The most popular school sports are:

1. Snow-shoeing
2. Ice-skating
3. Skiing

This is an incorrectly formatted list,

Item 1

Item 2

Item 3

```
Tags
<LI>
<LI>
  <LBody>
    <Text body>
      3.Skiing
<Text body>
  This is an incorrectly formatted lis
<L>
  <LI>
    <LBody>
      <Text body>
        Item 1
<Text body>
  Item 2
```

This example is shown in operation in the working example of ensuring lists are properly formatted in Acrobat Pro.

Example 4: Marking up lists using List structure elements

The following code fragment illustrates code that is typical marking up a list hierarchy in PDF documents. It uses the simple numbered list in the previous examples. This is typically accomplished by an authoring tool.

```
4 0 obj
  <</Type /Page
    /Contents 5 0 R
  >>

endobj
5 0 obj
  << /Length 3 0 R >>
  stream
   /P <</MCID 1>> BDC
      BT T* (The most popular sports are:) Tj ET EMC
   /Lbl <</MCID 11>> BDC
      BT T* (1. ) Tj ET EMC
   /LBody <</MCID 12>> /BDC
      BT (Snow-shoeing ) Tj ET EMC
   /Lbl <</MCID 21>> BDC
      BT T* (2. ) Tj ET EMC
   /LBody <</MCID 22>> /BDC
      BT (Ice-skating ) Tj ET EMC
   /Lbl <</MCID 31>> BDC
      BT T* (3. ) Tj ET EMC
   /LBody <</MCID 32>> /BDC
      BT (Skiing ) Tj ET EMC
```

```
endstream
endobj

101 0 obj                 % Structure element for intro paragraph to list ("The most popular sports are:")
  << /Type /StructElem
     /S /P
     /P 201 0 R
     /Pg 4 0 R
     /K 1
  >>
endobj

111 0 obj                  % Structure element for first item, list label (Lbl): "1."
  << /Type /StructElem
     /S /Lbl
     /P 211 0 R
     /Pg 4 0 R
     /K 11
  >>
endobj

112 0 obj
  << /Type /StructElem     % Structure element for first item, list text (LBody): ("Snow-shoeing")
     /S /LBody
     /P 211 0 R
     /Pg 4 0 R
     /K 12
  >>
endobj

... [ objects 121-122 and 131-132, referencing MCIDs 21-22 and 31-32 are omitted in the interest of space. ]

201 0 obj
  << /Type /StructElem
     /S /Caption            % Intro paragraph
     /P 400 0 R
     /K [101 0 R]
  >>
endobj

211 0 obj
  << /Type /StructElem
     /S /LI                 % List item for "1. Snow-shoeing"
     /P 400 0 R
     /K [111 0 R 112 0 R]
  >>
endobj

212 0 obj
  << /Type /StructElem
     /S /LI                 % List item for "2. Ice-skating"
     /P 301 0 R
     /K [121 0 R 122 0 R]
  >>
endobj

213 0 obj
  << /Type /StructElem
     /S /LI                 % List item for "3. Skiing"
     /P 301 0 R
     /K [131 0 R 132 0 R]
  >>
endobj

400 0 obj
  << /Type /StructElem
     /S /L                   % Top-level structure element in the list hierarchy
     /K [201 0 R 211 0 R 212 0 R 213 0 R]
  >>
endobj
```

Resources

Resources are for information purposes only, no endorsement implied.

- Section 14.8.4.3.3 (List Elements) in PDF 1.7 (ISO 32000-1)
- Edit Tags in the Tag Panel documentation
- PDF and Accessibility

Related Techniques

- G115: Using semantic elements to mark up structure

Tests

Procedure

1. For a list in a PDF document, verify in one of the following ways:
   - Read the PDF document with a screen reader, listening to hear that list is read correctly when reading the content line-by-line.
   - Use a tool that is capable of showing lists to open the PDF document and view the list to check that it is correctly structured.

- Inspect the tag tree to verify that the list is structured according to the PDF specification.
- Use a tool that exposes the document through the accessibility API and verify that the list is correctly structured.

Expected Results

- #1 is true

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF22: Indicating when user input falls outside the required format or values in PDF forms

Applicability

Tagged PDF documents

This technique relates to:

- Success Criterion 3.3.1 (Error Identification)
  - How to Meet 3.3.1 (Error Identification)
  - Understanding Success Criterion 3.3.1 (Error Identification)
- Success Criterion 3.3.3 (Error Suggestion)
  - How to Meet 3.3.3 (Error Suggestion)
  - Understanding Success Criterion 3.3.3 (Error Suggestion)

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF22. Also see PDF Technology Notes.

Description

The objective of this technique is to notify the user when user input to a field that requires a specific, required format (e.g., date fields) is not submitted in that format.

If the required format is not used, an alert dialog describes the nature of the error in text. This may be accomplished through scripting created by the author (see, for example, SCR18: Providing client-side validation and alert). User agents, such as Adobe LiveCycle can provide automatic alerts (as described in the examples below).

Note: Once the user dismisses the alert dialog, it may be helpful if the script positions the keyboard focus on the field where the error occurred, although some users may expect the focus to remain on the last control focused prior to the alert appearing. Authors should exercise care to ensure that any movement of the focus will be expected. For example, if the alert announces an error in a phone number format, positioning the focus on the phone number field when the alert is dismissed can be regarded as helpful and expected. In some cases, however, this may not be possible. If multiple input errors occur on the page, an alternative approach to error notification should be implemented.

Ensuring that users are aware an error has occurred, can determine what is wrong, and can correct it are key to software usability and accessibility. Meeting this objective helps ensure that all users can complete for-based transactions with ease and confidence.

Labels for required formats in form controls

It is also important that users are aware that an error may occur. You can incorporate this information in labels; for example, "Date (MM/DD/YYYY)." See PDF10: Providing labels for interactive form controls in PDF documents.

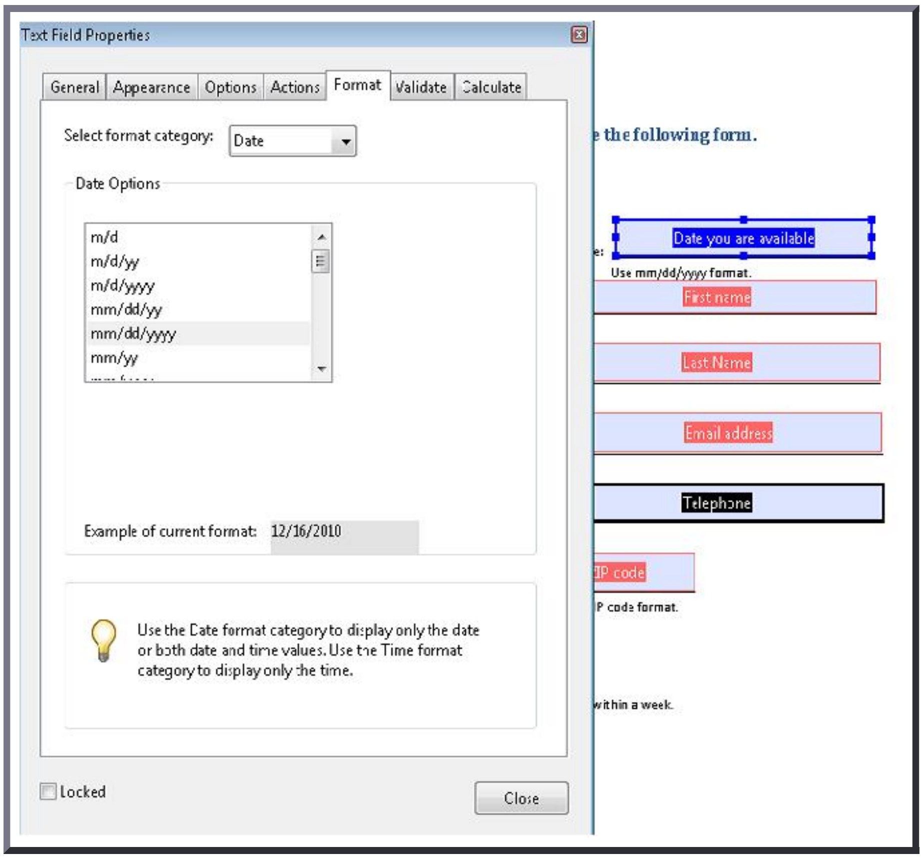
Examples

Example 1: Providing validation for an input field format using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

Many fields -- telephone number, postal code, date -- must have data entered in a specific format or pattern.

1. Access the context menu for the form control that requires a specific format.
2. Select Properties...
3. In the Format tab, select the Format Category (in this case, Date). The Date Options appear.
4. Select the format for this form control (in this case, mm/dd/yyyy).
5. In the General tab, specify "Date (mm/dd/yyyy)" for the Name and Tooltip for the control.


Text Field Properties

General | Appearance | Options | Actions | Format | Validate | Calculate

Select format category: Date

Date Options

m/d
m/d/yy
m/d/yyyy
mm/dd/yy
mm/dd/yyyy
mm/yy

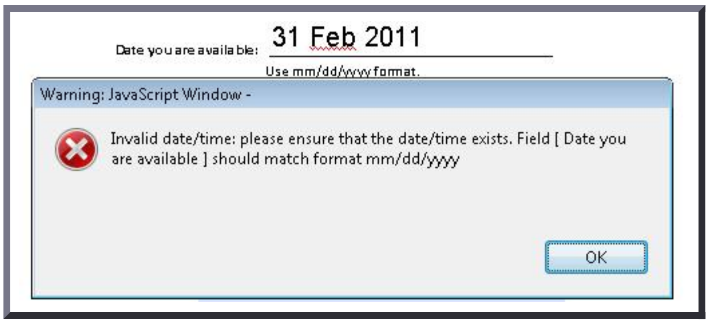
Example of current format: 12/16/2010

Use the Date format category to display only the date or both date and time values. Use the Time format category to display only the time.

Locked

Close

When a user types a recognized date format, it is converted automatically to the specified format. If the date format or value is not recognized, an error alert appears and provides further information, as shown in the image below.


Date you are available: 31 Feb 2011

Use mm/dd/yyyy format.

Warning: JavaScript Window -

Invalid date/time: please ensure that the date/time exists. Field [Date you are available] should match format mm/dd/yyyy

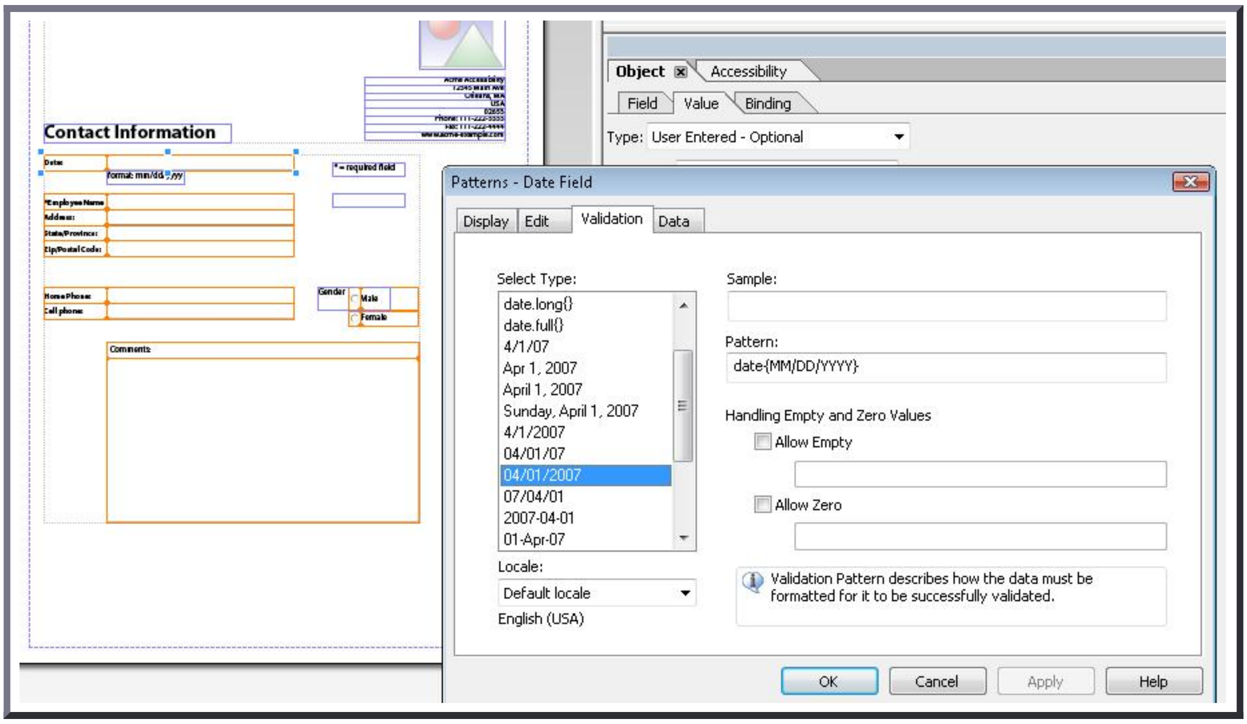
OK

This example is shown in operation in the working example of Required Fields in Acrobat.

Example 2: Providing validation for an input field format using Adobe LiveCycle Designer ES 8.2.1

This example is shown with Adobe LiveCycle Designer. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

1. Select the form control that has a required format.
2. In the Object palette, click the Validation Pattern... button.
3. Because this is a date field the Patterns-Date Field dialog appears. Select the pattern or format you want users to enter. Then click OK.



Type: User Entered - Optional

Patterns - Date Field

Select Type:
date.long{}
date.full{}
4/1/07
Apr 1, 2007
April 1, 2007
Sunday, April 1, 2007
4/1/2007
04/01/07
04/01/2007
07/04/01
2007-04-01
01-Apr-07

Pattern: date{MM/DD/YYYY}

Locale: Default locale
English (USA)

Validation Pattern describes how the data must be formatted for it to be successfully validated.

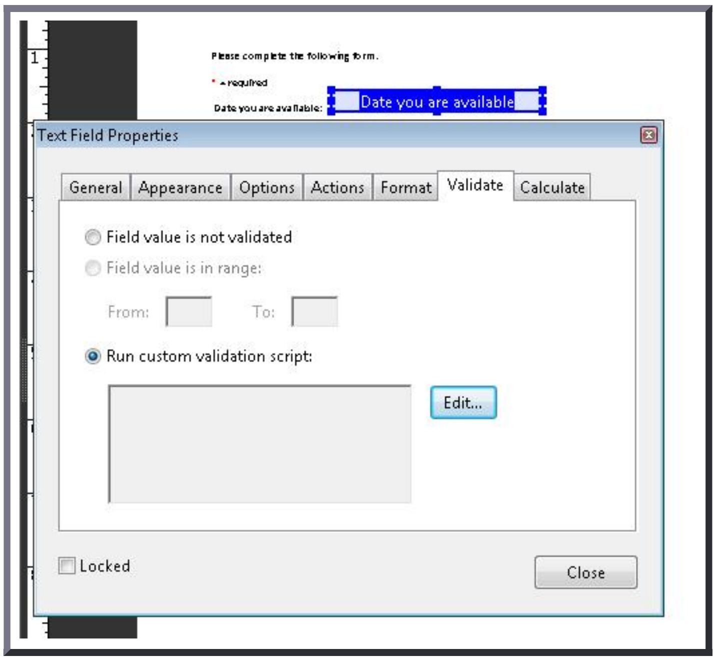
4. In the Object palette, use the Validation Pattern Message box to type a warning message. Be sure to include the required pattern. This message appears when a user tries to submit the form using an invalid date format.

This example is shown in operation in the working example of Required Fields in LiveCycle Designer.

Example 3: Validating a required date format in a PDF form using JavaScript using Adobe Acrobat Pro 9

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

The following JavaScript code illustrates the use of a script to validate form fields, in this case, a date field. To add this script to the form field, open the Text Field Properties dialog, as shown in Example 1, and select Edit in the Validate tab:



Please complete the following form.
* = required
Date you are available: Date you are available

Field value is not validated
Field value is in range:
[✓ Run custom validation script:]

```
// JavaScript code for date mask format MM/DD/YYYY
var re = /^[mdy0-9]{2}\/[mdy0-9]{2}\/[mdy0-9]{4}$/
//Allow blank space in field
if (event.value !="") {
  if (re.test(event.value) == false) {
    app.alert ({
       cTitle: "Incorrect Format",
       cMsg: "Please enter date using mm/dd/yyyy format"
    });
  }
}
```

Resources

Resources are for information purposes only, no endorsement implied.

- JavaScript for Acrobat

Related Techniques

- G89: Providing expected data format and example
- SCR18: Providing client-side validation and alert
- PDF23: Providing interactive form controls in PDF documents
- PDF10: Providing labels for interactive form controls in PDF documents
- PDF5: Indicating required form controls in PDF forms

Tests

Procedure

For each form field that requires specific input, verify that validation information and instructions are provided by applying the following:

1. Check that the format or value that is required is indicated in the form control's label.
2. Use an erroneous format or value and move off the field: make sure that an alert describing the error is provided.

Expected Results

- #1 and #2 are true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

PDF23: Providing interactive form controls in PDF documents

Applicability

- Tagged PDF documents with forms.
- PDF forms created using Adobe LiveCycle Designer.

This technique relates to:

- Success Criterion 2.1.1 (Keyboard)
  - How to Meet 2.1.1 (Keyboard)
  - Understanding Success Criterion 2.1.1 (Keyboard)
- Success Criterion 2.1.3 (Keyboard (No Exception))
  - How to Meet 2.1.3 (Keyboard (No Exception))
  - Understanding Success Criterion 2.1.3 (Keyboard (No Exception))

User Agent and Assistive Technology Support Notes

See User Agent Support Notes for PDF23. Also see PDF Technology Notes.

Description

The objective of this technique is to ensure that interactive form controls in PDF documents allow keyboard operation. Interactive PDF forms are generally created using a tool for authoring PDF. Form controls are implemented in PDF documents either as described in Section 12.7 (Interactive Forms) of PDF 1.7 (ISO 32000-1) or as described in the Adobe XML Forms Architecture (XFA).

The types of PDF form controls are: text input field, check box, radio button, combo box, list box, and button.

Form controls allow users to interact with a PDF document by filling in information or indicating choices, which can then be submitted for processing. Users who rely on keyboard access must be able to recognize and understand the form fields, make selections, and provide input to complete the forms, and submit the form, just as sighted users can.

Interactive form controls can be provided for forms created by converting a scanned paper form to tagged PDF or by creating a form in an authoring application such as Microsoft Word or Open Office and converting it to tagged PDF.

However, documents created by authoring applications that provide form design features might not fully retain their fillable form fields on conversion to PDF. Complex forms in particular may not have properly converted form fields and labels when tagged in conversion.

Using Adobe Acrobat Pro with forms in converted documents, you can ensure that form fields are keyboard accessible and usable by:

- Opening tagged PDF documents with form fields and creating interactive PDF form elements with the Run Form Fields Recognition tool.
- Modifying fillable form fields, or adding form fields, using Adobe Acrobat Pro or Adobe LiveCycle Designer.

Using Adobe LiveCycle Designer, you can create forms from scratch.

Examples

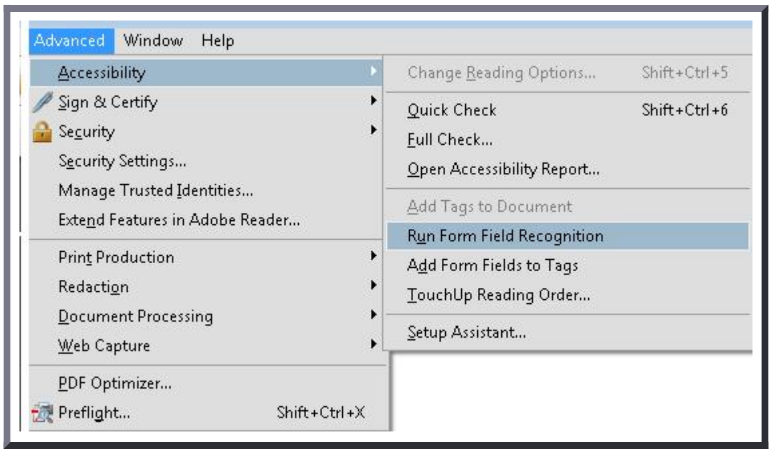
Example 1: Adding interactive controls to existing forms in PDF documents using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

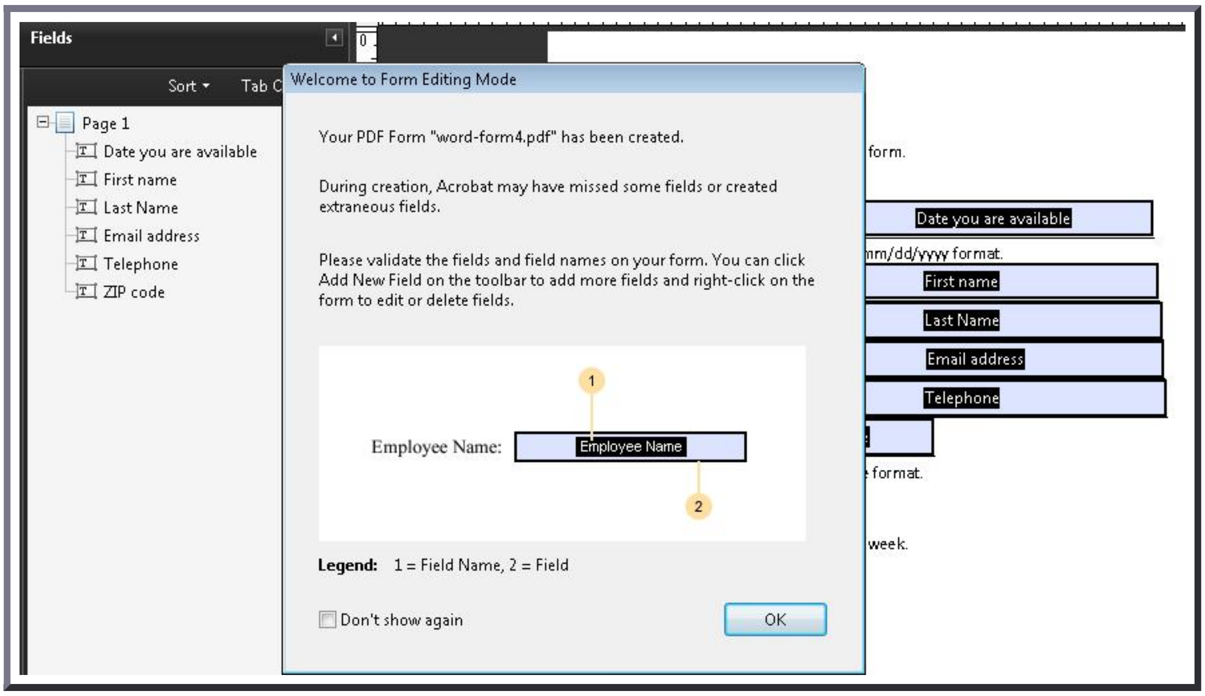
If you have a form in a tagged PDF document (created by scanning a paper form or using an authoring tool to generate tagged PDF), you can use Adobe Acrobat Pro to make the form elements keyboard accessible in the same page locations as the static form.

1. Use Advanced > Accessibility > Run Form Field Recognition to automatically detect form fields and make them fillable.

The following image shows the Run Form Field Recognition tool is selected to detect form fields in a document converted to tagged PDF.



The following image shows the resulting form fields after the Run Form Recognition tool is run.



Welcome to Form Editing Mode

Your PDF Form "word-form4.pdf" has been created.

During creation, Acrobat may have missed some fields or created extraneous fields.

Please validate the fields and field names on your form. You can click Add New Field on the toolbar to add more fields and right-click on the form to edit or delete fields.

Legend: 1 = Field Name, 2 = Field

Don't show again

OK

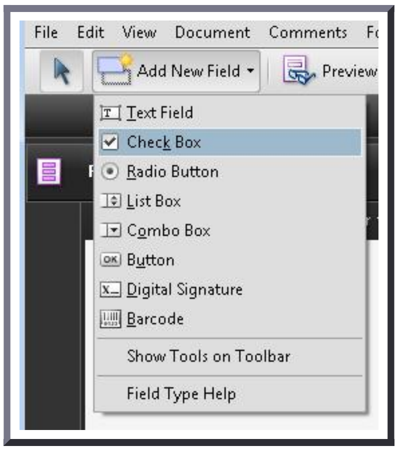
This example is shown in operation in the working example of Interactive Controls in Acrobat.

Example 2: Adding form controls in PDF documents using Adobe Acrobat 9 Pro

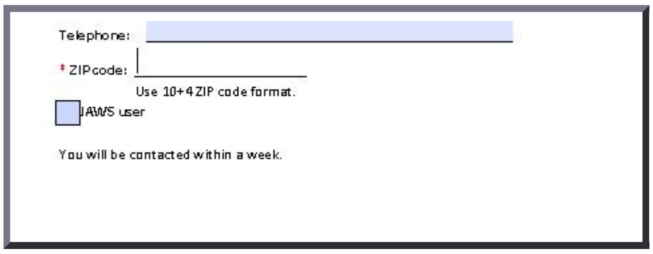
This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

You can add keyboard accessible form controls to your form as follows:

1. Forms > Add or Edit Fields... This puts the form in Form Editing mode.
2. Open the Add New Field menu on the upper left, and select a form field to add. The image below shows the menu of fields.



The following image shows a checkbox added to the form in Example 1.


Telephone:

* ZIPcode:

Use 10+4 ZIP code format.

JAWS user

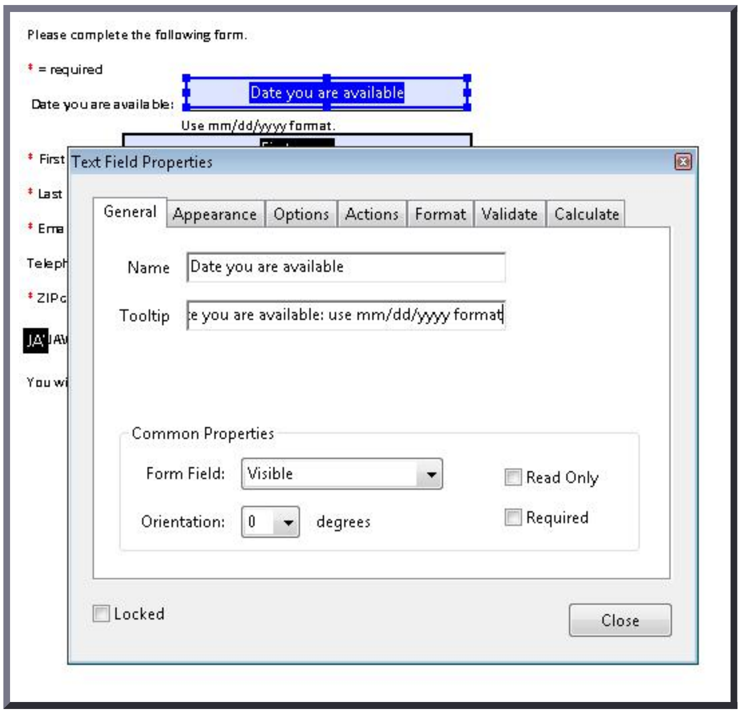
You will be contacted within a week.

This example is shown in operation in the working example of Interactive Controls in LiveCycle Designer.

Example 3: Editing form controls in PDF documents using Adobe Acrobat 9 Pro

This example is shown with Adobe Acrobat Pro. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

To edit fields, select the context menu for the field and select Properties... The properties menu for that form field lets you modify it, as shown in the following image.


Please complete the following form.

* = required

Date you are available: Date you are available

Use mm/dd/yyyy format.

Text Field Properties

General | Appearance | Options | Actions | Format | Validate | Calculate

Name: Date you are available

Tooltip: e you are available: use mm/dd/yyyy format

Common Properties

Form Field: Visible

Orientation: 0 degrees

Read Only

Required

Locked

Close

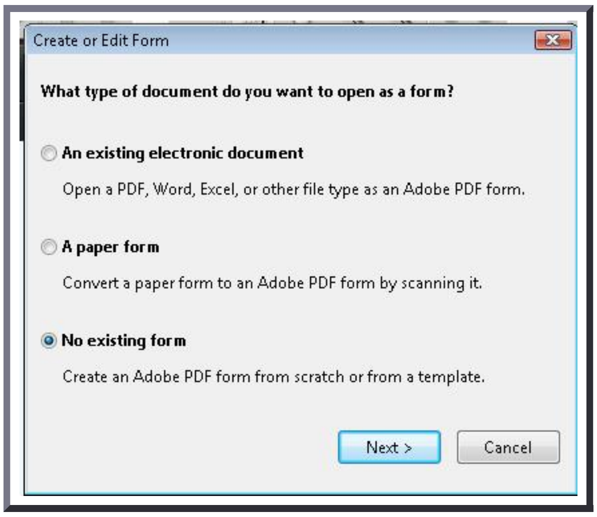
Note: The tooltip is not keyboard accessible but will be screen-reader accessible: see PDF12: Providing name, role, value information for form fields in PDF documents.

Example 4: Creating new interactive forms with Adobe LiveCycle Designer ES 8.2.1

This example is shown with Adobe LiveCycle Designer. There are other software tools that perform similar functions. See the list of other software tools in PDF Authoring Tools that Provide Accessibility Support.

You can use Adobe LiveCycle Designer to create new forms. In addition to invoking this standalone tool from the Windows Start menu, you can invoke it in Adobe Acrobat Pro:

1. Forms > Start Form Wizard...
2. Select the No Existing Form radio button, as shown in the following image.


Create or Edit Form

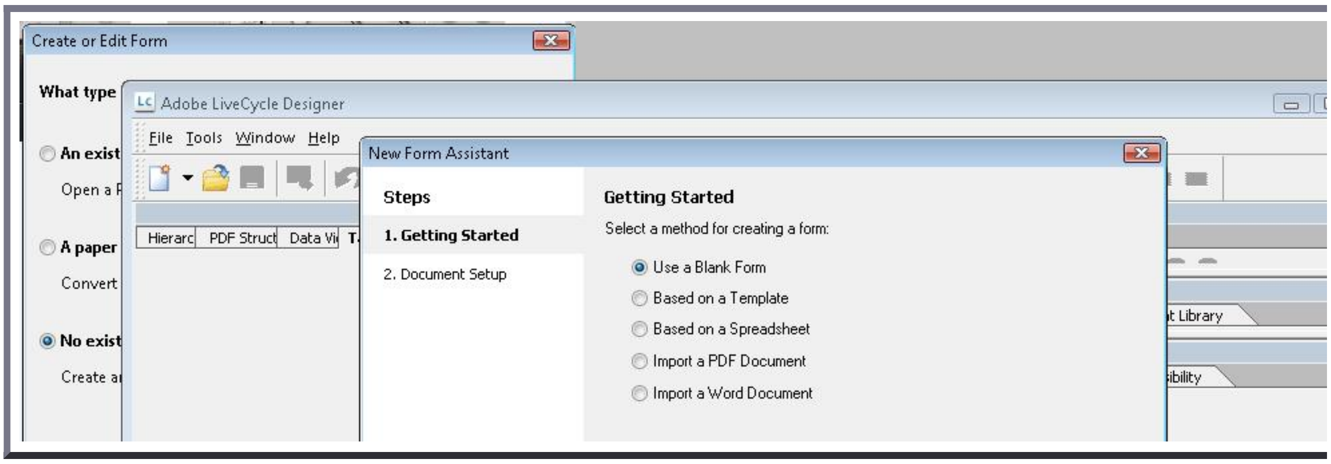
What type of document do you want to open as a form?

An existing electronic document
Open a PDF, Word, Excel, or other file type as an Adobe PDF form.

A paper form
Convert a paper form to an Adobe PDF form by scanning it.

[✓ No existing form]
Create an Adobe PDF form from scratch or from a template.

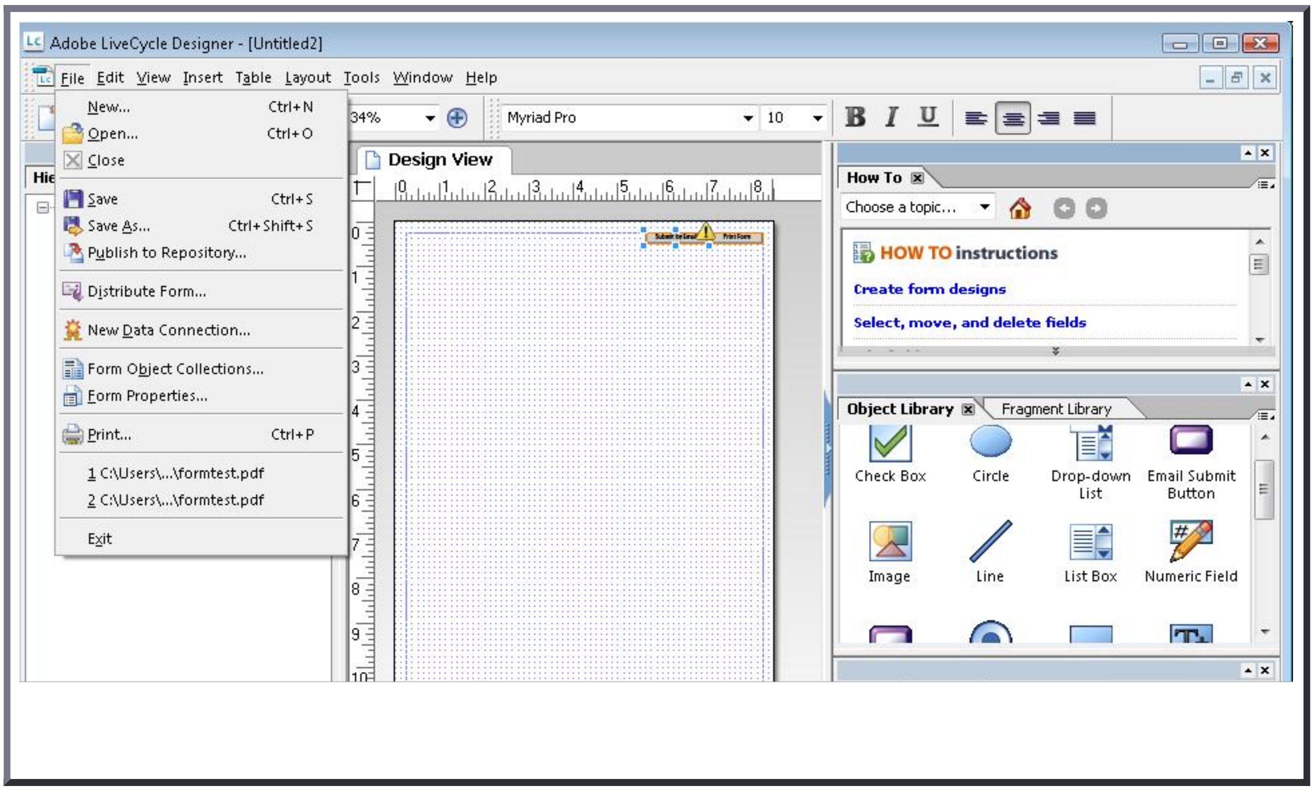
Clicking Next invokes LiveCycle Designer and the first page of the New Form Assistant. as shown in the following image.



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 Word Document

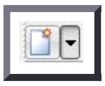
When you invoke LiveCycle Designer from the Windows Start menu, the Form Wizard is available from File > New...

The New Form Assistant creates a blank form. Use the Object Library in the right pane to select form controls.



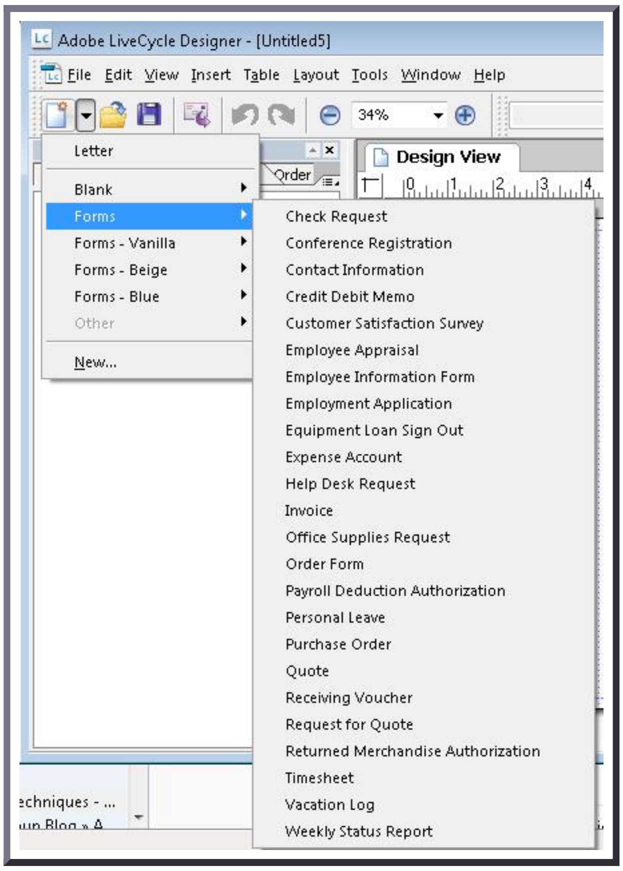
You can also use LiveCycle Designer to create forms based on commonly used forms templates.

1. Invoke the Template Assistant wizard from the New pulldown:



.

2. Select Forms and then select an appropriate type of form. Then, you can personalize the form by swapping out placeholder text, graphics, form fields, and properties with custom objects that you provide or define.



Example 5: Adding a text field in a PDF document using the /Tx field type

The following code fragment illustrates code that is typical for a simple text field such as shown in Examples 1 and 2. This is typically accomplished by an authoring tool.

```
<< /AP -dict-
   /DA /Helv  0 Tf 0 g
   /DR -dict-
   /F 0x4
   /FT Tx              % FT key set to Tx for Text Field
   /P -dict-
   /Rect -array-
   /StructParent 0x1
   /Subtype Widget
   /T Date you are available   % Partial field name Date
   /TU Date you are available: use mm/dd/yyyy format % TU tool tip value serves as short description
   /Type Annot
   /V Pat Jones
>>
...
<Start Stream>
 BT
  /P <</MCID 0 >>BDC
  /CS0 cs 0  scn
  /TT0 1 Tf
    -0.001 Tc 0.003 Tw 11.04 0 0 11.04 72 709.56 Tm
    [(P)-6(le)-3(as)10(e)-3( )11(P)-6(rin)2(t)-3( Y)8(o)-7(u)2(r N)4(a)11(m)-6(e)]TJ
  0 Tc 0 Tw 9.533 0 Td
  ( )Tj
  -0.004 Tc 0.004 Tw 0.217 0 Td
  [(\()-5(R)-4(e)5(q)-1(u)-1(i)-3(r)-3(e)-6(d)-1(\))]TJ
 EMC
  /P <</MCID 1 >>BDC
  0 Tc 0 Tw 4.283 0 Td
  [( )-2( )]TJ
   EMC
   /ArtifactSpan <</MCID 2 >>BDC
   0.002 Tc -0.002 Tw 0.456 0 Td
  [(__)11(___)11(___)11(___)11(___)11(_)11(____)11(___)11(___)11(__)]TJ
  0 Tc 0 Tw 13.391 0 Td
  ( )Tj
  EMC
 ET
<End Stream>
```

Resources

Resources are for information purposes only, no endorsement implied.

- Section 12.7 (Interactive Forms) in PDF 1.7 (ISO 32000-1)
- Adobe XML Forms Architecture (XFA)

Related Techniques

- G202: Ensuring keyboard control for all functionality
- PDF3: Ensuring correct tab and reading order in PDF documents
- PDF12: Providing name, role, value information for form fields in PDF documents
- PDF15: Providing submit buttons with the submit-form action in PDF forms

Tests

Procedure

1. For each form control, verify that it is properly implemented by tabbing to each form control and checking that it can be activated or that its value can be changed from the keyboard.

Expected Results

- #1 is true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

CSS1
"Cascading Style Sheets, level 1," B. Bos, H. Wium Lie, eds., W3C Recommendation 17 Dec 1996, revised 11 Jan 1999. Available at http://www.w3.org/TR/REC-CSS1/.

CSS2
"Cascading Style Sheets, level 2," B. Bos, H. Wium Lie, C. Lilley, and I. Jacobs, eds., W3C Recommendation 12 May 1998. Available at http://www.w3.org/TR/CSS2/.

CSS21
"Cascading Style Sheets, level 2 revision 1," B. Bos, T. Çelik, I. Hickson, H. Wium Lie, eds., W3C Candidate Recommendation 25 February 2004. Available at: http://www.w3.org/TR/CSS21/

CSS3
[CSS 2.1 and CSS 3] Roadmap, CSS WG's table of modules and publication dates.

PDF
"PDF", Adobe Systems. Available at http://www.adobe.com/devnet/pdf.html.

HTML4
"HTML 4.01 Specification," D. Raggett, A. Le Hors, I. Jacobs, eds., W3C Recommendation 24 December 1999. Available at http://www.w3.org/TR/html401/

WCAG20
"Web Content Accessibility Guidelines 2.0," B. Caldwell, M. Cooper, L. Guarino Reid, and G. Vanderheiden, eds., W3C Working Draft 11 December 2007. This W3C Working Draft is available at http://www.w3.org/TR/2007/WD-WCAG20-20071211/. The latest version of WCAG 2.0 is available at http://www.w3.org/TR/WCAG20/

XHTML1
"XHTML 1.0 The Extensible HyperText Markup Language (Second Edition)," S. Pemberton, et al., W3C Recommendation 26 January 2000, revised 1 August 2002. Available at: http://www.w3.org/TR/xhtml1/.

This Web page is part of Techniques for WCAG 2.0. The entire document is also available as a single HTML file. See the The WCAG 2.0 Documents for an explanation of how this document fits in with other Web Content Accessibility Guidelines (WCAG) 2.0 documents.