



**PDF
WCAG
2.0**

**PDF Accessibility Standard
for HRSDC**

PARTICIPANT MANUAL

April 2013

© HRSDC (2013)

TABLE OF CONTENTS

■	ABOUT THIS COURSE	5
	1.1 Background and Intent	5
	1.2 Objectives	5
	1.3 References	6
■	INTRODUCTORY SESSION	9
■	EXERCISES	10

ABOUT THIS COURSE

1.1 Background and Intent

The Government of Canada is committed to ensuring that a high level of Web accessibility is applied uniformly across its Web sites. This course outlines how WCAG (Web Content Accessibility Guidelines) 2.0 applies to PDF, and how to meet those guidelines.

This course has been primarily designed for IT specialists who may use PDFs as part of their web sites. However, it may also be of interest to communications specialists or graphic designers involved with technical aspects of Web design and development.

1.2 Objectives

Participants will be able to assess and fix PDF documents for accessibility issues, given familiarity with the WCAG 2.0 guidelines, according to the new standards on web accessibility for the government of Canada.

Treasury Board's Commitment to Accessibility

Canadians have the right to obtain information and services from Government of Canada Web sites regardless of the technologies they use. The key to effective implementation of universal accessibility lies in designing sites to serve the widest possible audience and the broadest possible range of hardware and software platforms, from adaptive technologies to emerging technologies.

For many Canadians, accessing Web content is more complicated than clicking a mouse and typing on a keyboard. Some Canadians rely on adaptive technologies such as text readers, audio players and voice-activated devices to overcome the barriers presented by standard Internet technologies. Others may be limited by their own technology.

The World Wide Web Consortium's Web Accessibility Initiative (WAI) has developed universal accessibility guidelines. Along with these WAI guidelines and the Government of Canada's standards on Accessibility, Usability, and Interoperability are directed toward ensuring equitable access to all content on Government of Canada Web sites.

1.3 References

Government of Canada web resources:

Standard on Web Accessibility:

<http://www.tbs-sct.gc.ca/pol/doc-eng.aspx?section=text&id=23601>

Standard on Web Usability

<http://www.tbs-sct.gc.ca/pol/doc-eng.aspx?section=text&id=24227>

Standard on Web Interoperability

<http://www.tbs-sct.gc.ca/pol/doc-eng.aspx?id=25875>

Standard on Mobile Applications

http://www.gcpeia.gc.ca/wiki/Standard_on_Mobile_Applications

World Wide Web Consortium web resources:

W3C PDF techniques for WCAG 2.0 (Working Draft):

<http://www.w3.org/TR/WCAG20-TECHS/pdf.html>

W3C Web Accessibility Initiative (WAI):

<http://www.w3.org/WAI/>

W3C Web Content Accessibility Guideline (WCAG) 2.0):

<http://www.w3.org/TR/WCAG20/>

W3C WCAG 2.0 Quick Reference document:

<http://www.w3.org/WAI/WCAG20/quickref/>

W3C WCAG 2.0 Technique Collections:

<http://www.w3.org/TR/2010/NOTE-WCAG20-TECHS-20101014/intro.html>

Other web based accessibility resources:

Web Accessibility In Mind (WEBAIM) PDF Accessibility:

<http://webaim.org/techniques/acrobat/>

Accessibility related tools:

Color contrast analyzer:

<http://www.paciellogroup.com/resources/contrast-analyser.html>

PDF Accessibility Checker

<http://www.access-for-all.ch/en/pdf-lab/pdf-accessibility-checker-pac.html>

The Donna Jodhan case and relevant Human Right links:

Donna Jodhan case:

http://www.gcpeia.gc.ca/gcwiki/images/1/1d/SDO_Information_Session_June_2011_Presentation_by_DoJ_Legal_-_eng.ppt

Canadian Charter of Rights and Freedoms:

http://laws.justice.gc.ca/eng/charter/page-1.html#_anchorbo-ga:l-l-gb:s_15

Canadian Human Rights Act:

<http://laws-lois.justice.gc.ca/eng/acts/H-6/page-1.html#h-3>

Policy on the Duty to Accommodate:

<http://www.tbs-sct.gc.ca/pol/doc-eng.aspx?id=12541§ion=text#cha7>

EXERCISES

Group Introduction Exercise

Purpose: To provide an opportunity for you to discuss accessibility concerns with your colleagues.

Proposed duration: 10 minutes

Introduce yourselves

- Who you are, where you work and what you do?
- Why are you interested in accessibility?
- Do you create PDFs, and if so, what tools do you use to create PDFs?

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

Purpose: To learn to properly deal with images in a PDF document.

WCAG Checkpoints: 1.1.1 Non-text Content

Proposed duration: 10 minutes

PDF1 ensures all information-conveying images have alternate text.

Steps:

- Open the inaccessiblePDF.pdf file in Adobe Acrobat Pro
- Run the Accessibility Check on the document. – This will find most images, but it's best to double-check manually
- Find all graphics that convey information
- Check alternate text for those images:
 - Hold your mouse cursor over the image and check the pop-up; or
 - Find the image in the tag tree (usually in a <figure> tag), right-click, select "Properties...", and check the Alternate Text textbox in the Tag tab.

PDF2: Creating bookmarks in PDF documents

Purpose: To check bookmarks

WCAG Checkpoints: 2.4.5 - Multiple Ways

Proposed duration: 5 minutes

Bookmarks offer a different way of traversing a PDF document.

Steps:

- Open the inaccessiblePDF.pdf file in Adobe Acrobat Pro
- Open the bookmarks panel (View -> Show/Hide -> Navigation Panes -> Bookmarks)
- Check to see if there are any bookmarks listed, and if there are, click on them to see if they go to the correct place

A document doesn't need bookmarks if it's short, but should have them if it's long.

**PDF3:
Ensuring correct tab and reading order in PDF documents**

Part A: Reading order

Purpose: to ensure reading order matches visual or logical order.

WCAG Checkpoints: 1.3.2 - Meaningful Sequence

Proposed duration: 20 minutes

Steps:

1. Open the Tag Tree panel (View -> Show/Hide -> Navigation Panes -> Tags)
2. Expand the tag tree
3. Navigate through the tags using the arrow keys watching the highlighted areas in the PDF window to ensure that the tags are in a logical order

Note that the reading order does not have to match the visual order. The reading order has to be logical so as not to be confusing to someone using a screen reader.

Part B: Tab Order

Purpose: to ensure tab order matches visual or logical order.

WCAG Checkpoints: 2.4.3 - Focus Order

Proposed duration: 20 minutes

Steps:

1. Find the first form element in the PDF document and select it.
2. Hit tab and check where the cursor jumps to.
3. Repeat Step 2 until the first element has focus again.

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

Purpose: To ensure decorative images are not in the tag tree, or have no alt text.

WCAG Checkpoints: 1.1.1 Non-text Content

Proposed duration: 5 minutes

Bookmarks offer a different way of traversing a PDF document.

Steps:

- Open the inaccessiblePDF.pdf file in Adobe Acrobat Pro
- Look for decorative image
- Check for alternate text (as per PDF1)

PDF5: Indicating required form controls in PDF forms

Purpose: To ensure that forms will not submit unless all required fields are filled in

WCAG Checkpoints: 3.3.1 - Error Identification, 3.3.2 - Labels or Instructions, 3.3.3 - Error Suggestion

Proposed duration: 5 minutes

Note: PDF5 only applies to PDF documents with forms to be electronically submitted.

Steps:

- Open the inaccessiblePDF.pdf file in Adobe Acrobat Pro
- Find required form field(s)
- On the Forms toolbar, click Edit
- Right-click the required form field and select "Properties..."
- Check the "Required" checkbox in the "General Tab"

PDF6: Using table elements for table markup in PDF Documents

Purpose: To ensure that tables are tagged with the <table> tag.

WCAG Checkpoints: 1.3.1 – Info and Relationships

Proposed duration: 5 minutes

Steps:

- Open the inaccessiblePDF.pdf file in Adobe Acrobat Pro
- Look for tables in the document and the tag tree
- Check that all tables are marked up with <table> elements
- A <table> should hold <tr> elements only
- A <tr> should hold <td> and <th> elements.
- Empty cells should be coded as <td> even when in a <tr> that holds other <th>s. In other words, <th> elements should never be empty.

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

Purpose: To ensure that scanned documents are converted to text.

WCAG Checkpoints: 1.4.5 - Images of Text

Proposed duration: 5 minutes

Steps:

- Open the Brochure-Dubai.pdf file in Adobe Acrobat Pro
- Look for tags.

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

Purpose: To ensure that abbreviations and acronyms have their associated meanings programmatically determinable.

WCAG Checkpoints: 3.1.4 - Abbreviations (AAA)

Proposed duration: 1 minute

3.1.4 - Abbreviations is a AAA WCAG requirement. The government only requires AA. Therefore none of our documents fail this criteria.

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

Purpose: Identify headings in the document and tag tree and ensure they're properly coded

WCAG Checkpoints: 1.3.1 - Info and Relationships, 2.4.1 - Bypass Blocks

Proposed duration: 10 minutes

Steps:

- Open the inaccessiblePDF.pdf file in Adobe Acrobat Pro
- Look through the document looking for what look like headers
- For each one, select with the cursor, and click "Find tag from selection"
- Check to ensure that it's a <h#> and that it's the correct level (1, 2, 3, etc.)
- Check the tag tree for <h#> tags. For each <h#> tag, is its content really a header?

PDF10: Providing labels for interactive form controls in PDF documents

Purpose: To check that form elements have labels (tool-tips) associated with them.

WCAG Checkpoints: 1.3.1 - Info and Relationships, 3.3.2 - Labels or Instructions, 4.1.2 - Name, Role, Value

Proposed duration: 5 minutes

Steps:

- Open the inaccessiblePDF.pdf file in Adobe Acrobat Pro
- Find form elements
- Check (using mouse-over) to ensure there is a label/tool-tip pop-up

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

Purpose: Identify headings in the document and tag tree and ensure they're properly coded

WCAG Checkpoints: 1.3.1 - Info and Relationships, 2.1.1 - Keyboard, 2.1.3 - Keyboard (No Exception) (AAA), 2.4.4 - Link Purpose (In Context), 2.4.9 - Link Purpose (Link Only) (AAA)

Proposed duration: 5 minutes

Steps:

- Open the inaccessiblePDF.pdf file in Adobe Acrobat Pro
- Find links
- Click on links to see if they open to where they're supposed to
- Check the links in the tag tree to see if they're using a <link> tag.

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

Purpose: Identify headings in the document and tag tree and ensure they're properly coded

WCAG Checkpoints: 1.3.1 - Info and Relationships, 4.1.2 - Name, Role, Value

Proposed duration: 5 minutes

Steps:

- Open the inaccessiblePDF.pdf file in Adobe Acrobat Pro
- Find form elements
- Check (using mouse-over) to ensure the label/tool-tip pop-up corresponds with the visual label. I.e: PDF10 checks to see if a label is there; PDF12 checks to see if the label is meaningful and correct.

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

Purpose: To check that links have meaningful text associated with them.

WCAG Checkpoints: 2.4.4 - Link Purpose (In Context)

Proposed duration: 5 minutes

Steps:

- Open the inaccessiblePDF.pdf file in Adobe Acrobat Pro
- Find links in the tag tree
- Check the properties and look for "Alternate Text" in the General tab.

PDF14: Providing running headers and footers in PDF documents

Purpose: To ensure information that's repeated on every page (ex: page number, document name, author, document version, etc.) is in the header/footer.

WCAG Checkpoints: 2.4.8 - Location (AAA), 3.2.3 - Consistent Navigation

Proposed duration: 5 minutes

Steps:

- Open inaccessiblePDF.pdf
- Check for "Add a title..." and <http://10.24.153.103/accessibility/inaccessible.html> [2013-04-17 2:14:16 PM] in the tag tree.
- Should they be in the tag tree? Compare with accessiblePDF.pdf.

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

Purpose: To ensure information that's repeated on every page (ex: page number, document name, author, document version, etc.) is in the header/footer.

WCAG Checkpoints: 3.2.2 - On Input

Proposed duration: 5 minutes

Steps:

- Open inaccessiblePDF.pdf
- Tab to the Submit button
- Hit Enter.
- What happens?

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

Purpose: To ensure document has the correct default language set.

WCAG Checkpoints: 3.1.1 - Language of Page

Proposed duration: 10 minutes

Steps:

- Open inaccessiblePDF.pdf
- Run Accessibility Full Check
- If any "No language" errors are returned, fail this checkpoint
- View Document Properties (ctrl+d)
- Select "Advanced" tab
- Check the last text box for the default language. It's not necessarily a fail if there's no attribute there. If it's not there, continue with the following procedures.
- Click "Okay"
- Open the tag panel on the left
- Select the top tag, right-click, and select "Properties..."
- Select the "Tag" tab
- Check the "Language" dropdown box.
- Check all tags for correct language.

PDF17: Specifying consistent page numbering for PDF documents

Purpose: To ensure page numbering is consistent throughout a document

WCAG Checkpoints: 1.3.1 - Info and Relationships, 2.4.8 - Location, 3.2.3 - Consistent Navigation

Proposed duration: 10 minutes

Steps:

- Open inaccessiblePDF.pdf
- Check the page numbering in the page and in the page number window in Acrobat Pro. Do they match?

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

Purpose: To check for a document title.

WCAG Checkpoints: 2.4.2 - Page Titled

Proposed duration: 5 minutes

Steps:

- Press ctrl+d and select "Description" tab.
- The first text field is the document title.
- Check the title to ensure that it's there, and that it's correct

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

Purpose: learn to check language on any part of a document in a language different than the default language (set in PDF16)

WCAG Checkpoints: 3.1.1 - Language of Page, 3.1.2 - Language of Parts

Proposed duration: 10 minutes

- Open inaccessiblePDF.pdf in Acrobat Pro
- Find all parts of a document that are written in a different language than the default.
- For each of those:
 - find them in the tag tree
 - right-click and select "Properties..."
 - In the Tag tab, check the Language drop-down box.

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

Purpose: To ensure tables are tagged as tables, and are properly tagged.

WCAG Checkpoints: 1.3.1 - Info and Relationships

Proposed duration: 30 minutes

Steps:

- Find each the table in the PDF document
- Check to see what how it's tagged
- Make sure header cells are tagged as <TH> and data cells are tagged as <TD>
- Ensure each row of cells is in a <TR>
- Look through the tag tree for all <Table> tags, checking they are actual tables.

PDF21: Using List tags for lists in PDF documents

Purpose: Ensure that lists, including nested lists, are tagged properly.

WCAG Checkpoints: 1.3.1 - Info and Relationships

Proposed duration: 20 minutes

Lists should be tagged with the <L> tag. All child tags of <L> are tags. tags can contain a wide variety of tags. The heading of a list should come before the <L> tag.

Steps:

- Find all lists in the document.
- Ensure that each list item is within a tag.
- Ensure that each list and nested list is within its own <L> tag.
- Ensure that the heading of the list comes before the <L> tag.

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

Purpose: Ensure that the user is notified when a form is submitted and a field has a required format, that format is followed by the user.

WCAG Checkpoints: 3.3.1 - Error Identification, 3.3.3 - Error Suggestion

Proposed duration: 5 minutes

Note that this checkpoint refers to forms that are to be submitted online. Our department uses HTML for online form submission. This checkpoint is rarely examined in a PDF assessment.

Steps:

- Open accessiblePDF.pdf in Acrobat Pro.
- Find the Phone Number field in the form section.
- Enter gibberish into the field and tab out.

PDF23: Providing interactive form controls in PDF documents

Purpose: to ensure that forms are fillable.

WCAG Checkpoints: 2.1.1 - Keyboard, 2.1.3 - Keyboard (No Exception) (AAA)

Proposed duration: 10 minutes

Note that some fields may not be editable on purpose, for example a signature field. A signature field requires a hand-written signature to be done with a pen, not filled in as part of a form.

Steps:

1. Open inaccessiblePDF.pdf in Acrobat Pro.
2. For each form element, check to see if it's editable.