

---

# Making the Web OPAC More Accessible

**Karen A. Coombs, SUNY Cortland**

View this presentation at:

<http://www.librarywebchic.net/presentations/naaug2004>

---

## An Introduction to Web Accessibility

- Accessibility means designing so that more people can use your web site in more situations.
  - Accessibility means making web sites that are perceivable, operable, and understandable.
  - Web Accessibility and Web Usability go hand in hand
  - Web Accessibility and Web Standards go hand in hand
  - Usable Accessibility
    - more than using an accessibility checker
    - interact with the web pages in various configurations
  - Two important themes in web accessibility: ensuring graceful transformation, and making content understandable and navigable
  - Reasons to provide accessible websites
    - compliance with regulatory and legal requirements
    - web site available to more people (better exposure)
    - web site available in more situations (more places, devices)
    - better design and implementation
    - cost savings
-

## **Web Accessibility takes into consideration any and all of the following:**

- Disabled Users
    - Visual
      - blindness, low vision, color-blindness
    - Hearing
      - deafness, various degrees of hearing loss
    - Motor
      - inability to use a mouse, slow response time, limited fine motor control
    - Cognitive
      - learning disabilities, distractibility, inability to remember or focus on large amounts of information
  - New and Old Technology
    - PDAs
    - Cell phones
    - WebTV
    - Slow connections
    - Different platforms, different browsers
  - And more
-

## W3C's Web Content Accessibility Guidelines

- The World Wide Web Consortium has created a list of Web Content Accessibility Guidelines
    - [version 1 finalized](#) [[checklist](#)]
    - [version 2 in draft stages](#)
  - Version 1 contains a list of guidelines that are divided up into priorities
    - **Priority 1**

A Web content developer must satisfy this checkpoint. Otherwise, one or more groups will find it impossible to access information in the document. Satisfying this checkpoint is a basic requirement for some groups to be able to use Web documents.
    - **Priority 2**

A Web content developer should satisfy this checkpoint. Otherwise, one or more groups will find it difficult to access information in the document. Satisfying this checkpoint will remove significant barriers to accessing Web documents.
    - **Priority 3**

A Web content developer may address this checkpoint. Otherwise, one or more groups will find it somewhat difficult to access information in the document. Satisfying this checkpoint will improve access to Web documents.
  - The W3C's Web Content Accessibility Guidelines are not the only accessibility "rules".
    - Section 508 [[checklist](#)]
      - based on Priority Level 1 of W3C's WCAG 1.0
      - additional rules added or reworded
    - State Accessibility Regulations ([New York State 99-3](#))
-

## Methods for Assessing Web Accessibility

- Validate your markup (HTML, XHTML, CSS)
    - <http://validator.w3.org/>
  - Manually evaluate your web pages against the Web Content Accessibility Guidelines
  - Use a Tool for Evaluating Web Accessibility
  - Test web page using different web browsers, platforms and assistive technologies
    - Opera, Firefox, Lynx
    - JAWS, Windows Eyes
  - Test web page with stylesheets and client-side scripting (Javascript) turned off
  - Conduct user testing
    - good practice for both *accessibility* and *usability*
-

## Tools for Evaluating Web Accessibility

- There is no one single tool that will completely evaluate your pages for web accessibility.
- Many of the checkpoints need to be evaluated by a human.
- You need to understand the standards so that you can make judgements regarding whether or not a page is accessible.

### Some Accessibility Tools and Their Uses

- Bobby Online Service (Watchfire) <http://bobby.watchfire.com/bobby/html/en/index.jsp>
    - Evaluates Web Pages Against WCAG 1.0 Priority 1-3 and Displays Errors: the tool automatically checks the Web site for WCAG 1.0 Priority 1-3 for automated errors and lists these errors in the report.
    - Evaluates Web Pages Against Section 508 and Displays Errors: the tool checks the accessibility of the page using Section 508 standards.
    - Displays instances of types of errors on page
  - Opera
    - See what the web page looks like:
      - without images
      - without JavaScript
      - without a stylesheet
      - without tables
      - when you enlarge the text
  - Lynx
    - This text-only browser will give you the send of what it is like to navigate in a text-only environment using the keyboard. Very similar to testing with Opera but *forces* keyboard-based navigation.
  - JAWS
    - This screen reading software will give you a very good idea of what the page sounds like to a blind user. Turn your monitor off to get the full effect
  - [Color Picker and Tester](#)
    - See what certain color combinations look like to those with various types of color-blindness
-

## **Accessibility and the WebOPAC**

- Evaluating Accessibility Compliance
  - Some WebOPACs in Opera
  - Accessibility Basics
  - Tables for Layout
  - Accessibility and Forms
  - Other Ways to Enhance the Web Accessibility of the WebOPAC
-

## Evaluating Accessibility Compliance

- Issues with assessing compliance
    - WebOPAC is dynamic in nature
    - Impractical to check individual pages using a tool such as Bobby
      - Save a given page as HTML and check
      - Some pages can be checked with direct linking
    - Really need to test WebOPAC as a whole
  - Best Bets for Evaluating Compliance
    - Manually evaluate the WebOPAC against the WCAG
      - This will reveal many issues
    - Test the WebOPAC as if you are using it
      - Opera
      - Lynx
  - How does the WebOPAC rate?
    - passes all of the WCAG Priority 1 checkpoints
    - passes some of the WCAG Priority 2 checkpoints
    - passes few of the WCAG Priority 3 checkpoints
-



## Some Example WebOPACs in Opera

- [Harvard](#)
  - [SUNY Binghamton](#)
  - [Texas Tech](#)
  - [University of Iowa](#)
-

## Accessibility Basics

- Use text for text, not images
  - Use relative font sizes
  - Alt tag images
  - Text equivalent for any audio
  - Avoid using color alone to convey information
  - Table Tips
    - Do not use tables for layout unless the content will make sense when the tables are linearized
    - If a table is use for layout, do not use any structural markup for the purpose of visual formatting only
    - Use table headings and/or scope attribute to designate table header information
-

## Tables for Layout

- ALEPH uses many tables for layout on almost any page
    - header layout is tables
  - Tables can create problems for screen readers
  - Sometimes tables are necessary
    - Search results screen
  - Many of the layout things done with tables can be done using CSS
    - head-1
    - find-include-buttons
    - short-include-buttons
    - short-include-2
-

## Example: Basic Search Screen

The screenshot shows the top navigation bar of the Cortland Out of Box ALEPH system. It includes a logo on the left and a series of menu items: Sign-in, End Session, User, Preferences, Databases, and Feedback. Below these are secondary items: Browse, Search, Results List, Previous Searches, and Basket. A second row of navigation options includes Basic Search, Multi-field, Multi-base, Advanced Search, and CCL. The main content area is titled "Basic Search" and contains a search form with the following elements:

- Type word or phrase**: A text input field.
- Field to search**: A dropdown menu currently set to "All Fields".
- Words adjacent?**: Radio buttons for "No" (selected) and "Yes".
- Buttons**: "Go" and "Clear".

- [Page redone using CSS](#)
    - Wireframe redo [pulled HTML ALEPH generated and rewrote]
      - can be used to make the modifications to the actual ALEPH files
    - Not exact match but very close
    - Left comments which show which file contains different pieces of the page
    - Style modification are included in the <style> element
  - What was done
    - removed all tables for layout
    - added label elements and associated each label with input elements using for and id attributes
    - set a tab order for form elements
-

## Web-based Forms and Accessibility

- Label element
  - associating a label with an input element
  - id attribute
  - for attribute
- fieldset element
  - allows you to group together a set of form-related elements
  - legend element allows you label the group of elements
- Setting a tab order for a page
  - all form-related elements can have a tab attribute

### Example Code:

```
<form method=get
name=form1
action="http://oswlib.library.oswego.edu:4390/F/88SSH8CSV41NN7LAAGY2JTCLM4ADX6Q7AVLS26JD88C2NVINDU-00083">
<input type="hidden" name="func" value="find-b">
<label for="search_phrase">Type word or phrase
<input size=40 name="request" value="" id="search_phrase" tabindex="1">
</label>
<label for="index">
Field to search
<!-- filename: find-code-include -->
<select name=find_code id="index" tabindex="2">
<option value=WRD >All Fields</option>
<option value=WTI >Title</option>
<option value=WAU >Author</option>
<option value=WSU >Subject</option>
<option value=ISSN >ISSN</option>
<option value=ISBN >ISBN</option>
<option value=SYS >System number</option>
<option value=BAR >Barcode</option>
</select>
</label>
<div id="adjacent">
Words adjacent?
<label for="adjacent_no">
<input Type="radio" name="adjacent" value="N" checked id="adjacent_no" tabindex="3"> No
</label>
<label for="adjacent_yes">
<input type="radio" name="adjacent" value="Y" id="adjacent_yes" tabindex="4"> Yes
</label>
</div>
<p>
<input type="image" alt=" Go " src="http://oswlib.library.oswego.edu:4390/www_eng/icon/f-go.gif"
border="0" tabindex="5">
<a href="javascript:document.form1.reset();" tabindex="6">
</a>
</p>
</form>
```

---

## Other Ways to Enhance the Web Accessibility of the WebOPAC

- Use Media Types
    - aural
    - braille
    - handheld
    - print
  - Change images that contain text to actual text; where possible
  - Provide keyboard shortcuts to important links
-

## Analyzing Cost versus Benefit of Accessibility Modification

1. Does this modification need to be made to meet government accessibility regulations?
  2. How significantly does this alter the page from the original version?
    - Upgrades
    - Support
  3. How do these modifications make the WebOPAC more usable for all users?
  4. What staff is necessary to make and support these modifications?
-

## Further Information

Adams, C. (2004). "Accessible Stylish Form Layout",

<http://www.themaninblue.com/writing/perspective/2004/03/24/> Accessed: 6/11/2004

Bohman, P. (2003). "Using Opera to Check for Accessibility",

<http://www.webaim.org/techniques/articles/opera>, Accessed: 6/11/2004

Dive into Accessibility - <http://www.diveintoaccessibility.org/>

Moss, T. (2003). "What is Web Accessibility", <http://www.alistapart.com/articles/wiwa/> Accessed: 6/10/2004

Section 508: The Road to Accessibility - <http://www.section508.gov/>

Thatcher, J., Waddell, C., Henry, S., Swierenga, S., Urban, M., Burks, M., Regan, B., & Bohman, P. (2002). *Constructing Accessible Web Sites*, Birmingham, U.K. : Glasshaus.

Web Accessibility in Mind - <http://www.webaim.org/>

Web Accessibility Initiative (WAI) Home Page - <http://www.w3.org/WAI/>

W3C WCAG Checklist - <http://www.w3.org/TR/WCAG10/full-checklist.html>

---



[HOME](#)

[Print a copy of this presentation](#)