

Web Course Curriculum Applications

High Tech Center Training Unit

Of the California Community Colleges at the
Foothill-De Anza Community College District

21050 McClellan Road
Cupertino, CA 95014
(408) 996-4636

www.htctu.fhda.edu

Table of Contents

ACCESSIBILITY & COURSEWARE MANAGEMENT SYSTEMS	5
ACCESSIBILITY RESOURCES FOR COURSEWARE MANAGEMENT SYSTEMS.....	11
USING THE ALT-TAG	13
CREATING ACCESSIBLE TABLES	19
EVALUATION AND REPAIR TOOLS LIST.....	23
BOBBY EVALUATION TOOL	25
A-PROMPT EVALUATION TOOL.....	27
ADDITIONAL WEB RESOURCES	29

Accessibility & Courseware Management Systems

Introduction

Courseware management systems allow users to create and move educational content quickly and easily to the Internet. These systems help remove barriers to posting academic information to the World Wide Web by providing a mechanism that eliminates the need to learn complex web authoring tools or web-based computer languages necessary to deliver information. Additionally, courseware management systems provide a template that can be used throughout a college and university setting, ensuring that the information delivered to students is similar in design and appearance.

One important aspect of courseware management systems that should be considered by faculty and administrators is the accessibility of such web-based interfaces. Assistive computer technology, most notably screen-reader software, does not always interface with courseware management systems seamlessly. Blind or low-vision students may have difficulty navigating through a courseware management system using the keyboard, while hearing impaired students may not respond to auditory cues triggered by a mouse command.

Additionally, many of the features of these management systems are not under the control of faculty members. Simple changes to graphic images or hyperlink names, for instance, are not always possible unless one has access to the underlying programming code.

There are solutions that faculty can implement using courseware management systems to make web-based course content more accessible to not only students with disabilities, but universally accessible to all students. The following information identifies the accessibility requirements established by the Section 508 Standards and how certain courseware management systems respond to these standards.

Section 508 Standards and Courseware Management Systems

It is important to note the difference between the courseware management system shell interface and the content posted to that interface by curriculum developers. Courseware management systems have a responsibility to insure the shell provides an accessible interface. Instructors and course developers have a responsibility to insure the content and format of the information posted to a courseware management system shell is accessible. Listed below are the Section 508 Standards and the steps courseware management systems have taken to insure an accessible interface.

(A) A text equivalent to every non-text element shall be provided (e.g., via “alt”, “longdesc”, or in element content).

Blackboard: Blackboard allows users to add an “alt-tag” by accessing the Multimedia Options settings and inputting information into the Alternative Text field.

WebCT: WebCT provides users with the ability to include an “alt-tag” to an image that is uploaded into the WebCT shell. Users can enter this “alt-tag” by entering text into the “Title” field. “Alt-tags” should be descriptive and relevant to the image being described.

(B) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.

Blackboard: Blackboard does not contain multimedia presentations.

WebCT: WebCT does not contain multimedia presentations.

(C) Web Pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.

Blackboard: The Blackboard shell implements this function throughout the application.

WebCT: This does not apply to the WebCT shell.

(D) Documents shall be organized so that they are readable without requiring an associated style sheet.

Blackboard: Blackboard is accessible without using requiring style sheets.

WebCT: This change has been made for WebCT version 3.6.

(E) Redundant text links shall be provided for each active region of a server-side image map.

Blackboard: Blackboard does not make server-side image maps.

WebCT: WebCT does not use server-side image maps.

(F) Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.

Blackboard: Blackboard only uses client-side image maps.

WebCT: WebCT does not use server-side image maps.

(G) Row and column headers shall be identified for data tables.

Blackboard: Blackboard has added row and column headers for data tables within the shell interface.

WebCT: This change has been made for WebCT version 3.6.

(H) Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.

Blackboard: Additional code has been added into the Blackboard interface to provide greater accessibility for data tables. Blackboard has chosen to use the WAI guidelines for creating accessible tables to satisfy this function.

WebCT: This change has been made for WebCT version 3.6. WebCT uses the “id” and “header” method to associate data cell content with the row and column headers of a data table.

(I) Frames shall be titled with text that facilitates frame identification and navigation.

Blackboard: Blackboard provides names for the frames within the Blackboard shell. Screen-readers should not have difficulty navigating through the frames in Blackboard, however, the names of the frames are abbreviated which may cause some confusion as to what frame the user is viewing. See the WebCT example below for a possible solution.

WebCT: WebCT does provide more meaningful titles for the frames in the WebCT interface. However, the screen-reader program JAWS does not read the “Title” attribute and reads the “Name” attribute. Thus, a JAWS user may still have difficulty navigating through the different frames. A possible solution is for the JAWS user to rename the frame name in the Dictionary Manager. For example, if the frame is named “main_win”, then use the Dictionary Manager to identify “main_win” as “Main content window” or another appropriate title.

(J) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.

Blackboard: The Blackboard shell does not use blinking or flickering elements within the shell interface.

WebCT: The WebCT shell does not use blinking or flickering items that could cause screen flicker between these frequencies.

(K) A text-only page with equivalent information or functionality, shall be provided to make a website comply with the provisions of these standards, when compliance cannot be accomplished in any other way. The content of a text-only page shall be updated whenever the primary page changes.

Blackboard: A text-only page would not be appropriate for the Blackboard shell due the nature of the technology.

WebCT: A text-only page would not be appropriate for the WebCT shell due the nature of the technology. WebCT continues to make changes to the course shell instead.

(L) When pages use scripting language to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.

Blackboard: Javascript is used in the Blackboard shell and meets this requirement. Elements requiring javascripting developed within the shell should be accessible using assistive technology.

WebCT: Javascript is used in the WebCT shell and is designed to increase usability. There does not seem to be major accessibility issues with the use of javascript in the WebCT shell.

(M) When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with (Section 508) 1194.21 (a) through (l).

Blackboard: Applets such as the Chat module or Virtual Classroom are not functional to screen-reading technology. There is no information as to when these modules will become accessible.

WebCT: The WebCT shell does not require the use of third-party plug-in technology for normal operation. Internal applets, such as the Chat and Whiteboard elements, are not functional using screen-reading technology and remain inaccessible. No information is provided as to when these elements will be functional to screen-reader users.

(N) When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and clues.

Blackboard: Blackboard has taken several steps to develop accessible forms from within the Blackboard shell. Form labels are adjacent to the form input elements they represent. According to Blackboard, text labels will precede text input fields; text labels will follow radio button options; and, are located above text area inputs.

WebCT: WebCT has made corrections to approximately 1200 forms at the Student and Designer level.

(O) A method shall be provided that permits users to skip repetitive navigation links.

Blackboard: Skip navigation links have been implemented into the Blackboard shell software.

WebCT: This feature is now provided in WebCT version 3.6. The link allows the user to bypass the navigation bar of the WebCT shell.

(P) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.

Blackboard: Blackboard allows instructors to set time limits for students taking a quiz in the Blackboard shell. Instructors should be aware if there are any students requesting additional time to complete a quiz. Blackboard will automatically logoff a user if the user does not interact with the system within a three hour time period.

WebCT: WebCT allows instructors to develop quizzes that may have a timed component to the quiz. The instructor has the capacity to alter or remove the timed component to the quiz function in the WebCT shell.

Accessibility Resources for Courseware Management Systems

Blackboard

<http://access.blackboard.com>

<http://www.blackboard.com>

The Blackboard accessibility website contains information pertaining to the accessibility of the Blackboard interface and what steps faculty can take to develop accessible content to post to within the shell. Blackboard has also created a tutorial for screen-reader users in order to assist users navigating the interface as well as tips how to access content within Blackboard.

eCollege

<http://www.ecollege.com>

eCollege provides a different service as courseware management systems are utilized. Rather than just providing a shell interface that instructors are required to develop content within, eCollege will assist the faculty member in developing a web-based curriculum specific to the needs of the instructor. For more information about eCollege, contact Heather Matson at: heatherm@ecollege.com

Prometheus

<http://www.prometheus.com>

Prometheus is a courseware management system developed by the George Washington University. Unlike other courseware management systems, Prometheus provides open access to the source code of the Prometheus platform. This allows IT administrators at the university/college to create new features or customize applications in order to better serve students and faculty. Faculty can evaluate the Prometheus interface by creating a free account at: http://www.prometheus.com/product/free_account.html

WebCT

<http://www.webct.com>

WebCT has several websites that can assist faculty with the development of accessible content for the WebCT interface. The WebCT Discussion Forum also contains information and resources for faculty to share. For more information regarding accessibility within the WebCT shell, enter the keyword “accessibility” into the WebCT Search function.

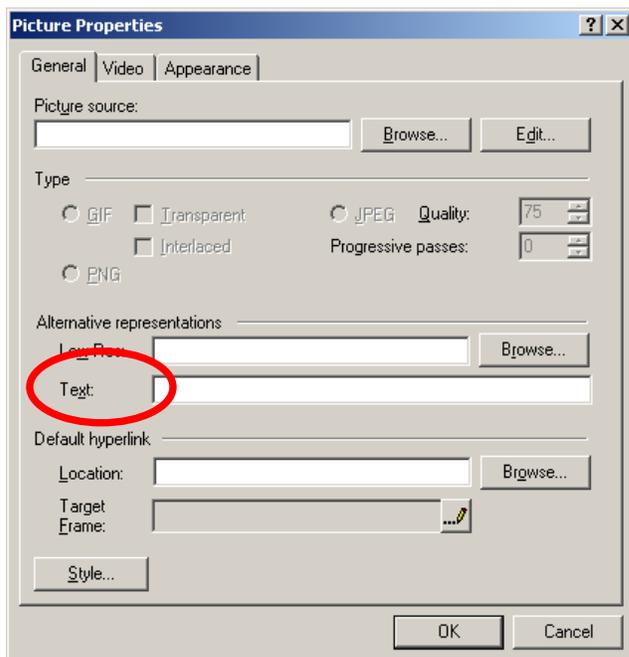
Using the Alt-Tag

Introduction

An important component of developing accessible web pages is to add the alt-attribute to images. This allows visually impaired individuals using screen reader software or a text-only browser to "hear" the description of the image presented on the web page. Adding the alt-attribute is required under the WAI Priority 1 Guidelines and the Section 508 Standards.

Using FrontPage 2000:

1. Insert the image to the web page using the **Insert** command from the menu bar.
2. **Right click** on the image using the mouse, or select the image and use the **Alt+Enter** keystroke, to access the Picture Properties menu.
3. Under the **Alternative Representations** heading, type the alternative text for the image in the **Text:** box.

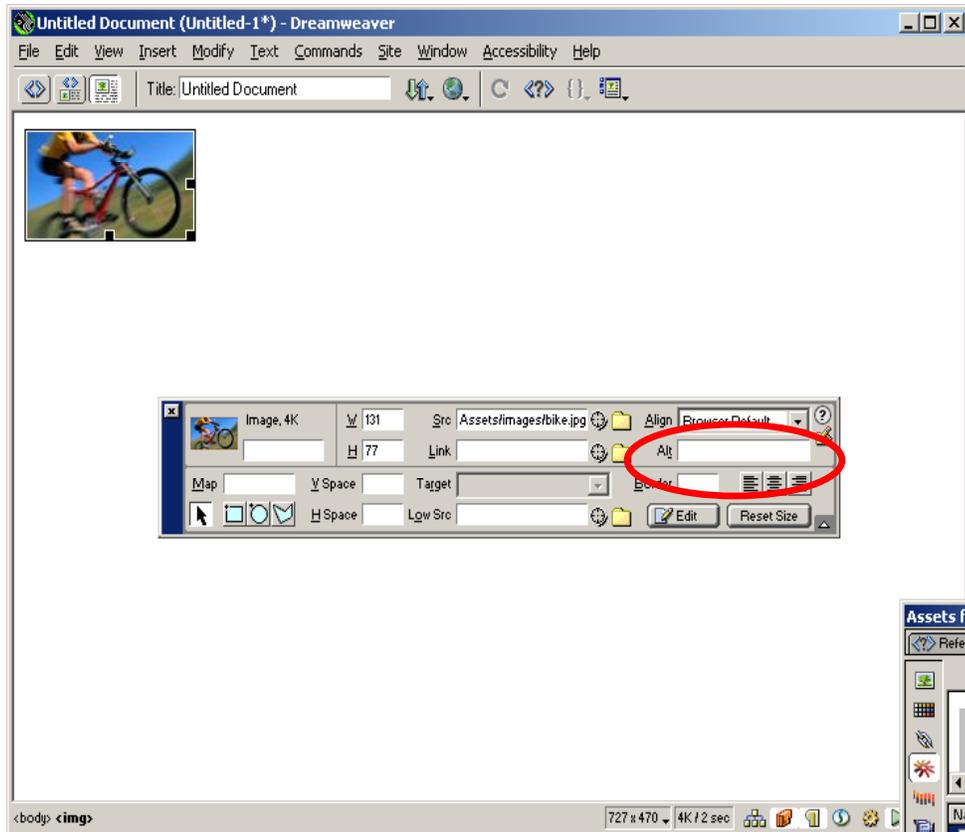


4. Select **OK**.

You have just added an alt-tag to an image.

Using Dreamweaver 4:

1. Insert the image to the web page using the **Insert** command from the menu bar, or the keyboard command **Ctrl+Alt+I**.
2. **Right click** on the image using the mouse, or select the image using the arrow and shift key. The Picture Properties box will then appear (see screen shot).



3. Under the **Alt** heading, type the alternative text for the image in the box.

You have just added an alt-tag to an image.

Editing the HTML Code:

You can edit the HTML code using several different methods. Most web page authoring programs will allow the developer to switch between the WYSIWYG interface and the underlying HTML code interface. You can also edit the HTML code by opening the web page in Notepad, WordPerfect, or Word 2000. The HTML code can then be edited as a regular text document. Here is more information for adding an alt-attribute by editing the HTML code:

1. Identify the image that you wish to add the alt-attribute. An easy method is to look for the words **img src=**. This will be the line in which you place the alt= tag

```
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#FFFFFF" text="#000000">

</body>
</html>
```

2. Insert the HTML code **alt="descriptive text goes here"** before the > sign.
3. This will add the alt-attribute to the desired image.

You have just added an alt-tag to an image by editing the HTML code.

Using the d-Link:

The d-link is an interim solution that web designers are using to communicate the meaning of an image in greater detail. This solution will be replaced by the use of the LONGDESC attribute when browsers are able to recognize HTML 4.0. Until then, it is appropriate to use this solution to provide the user with a greater description of the image presented on the web page.

The convention for the "d-link" option is to place a lower-case "d" near the lower right region of the desired image. The lower-case "d" is then hyperlinked to a text only document that describes the image in more detail. An example is provided below.

1. Insert an image into the web page you are creating.
2. Add a lower-case "d" to the bottom right corner of the image.



3. Create a separate web page that contains a more detailed description of the image you are displaying on your web page.
4. Create a hyperlink from the lower-case "d" to the separate web page that contains the detailed description of the image on your web page.

This will create a "d-link" that will provide more information to a visually impaired or blind individual.

Editing a JAVA Applet:

Java applets are widely used by web designers who desire an interactive or more dynamic presentation for the internet visitor. JAVA applets are essentially small programs that can be launched from a web page and will run on the visitor's computer. The advantage of a JAVA applet is that it will not write information to the user's hard disk and the program can be run on several different operating system platforms.

A major disadvantage of JAVA applets is that these programs are not accessible to screen-readers. For example, a screen-reader program will not read a JAVA applet that displays updated stock values on a web page. Classroom management systems, such as Blackboard or WebCT, contain virtual chat modules that are JAVA applets. Once again, screen-readers will not recognize or communicate to the user information presented in a JAVA applet window.

To communicate to a user that a JAVA applet is on the page, a web author can use the alt-attribute.

Example:

1. Locate the applet code on in the HTML. This will look like <APPLET>.
2. Insert the appropriate description of the JAVA applet using the following syntax:
alt= "descriptive text goes here" .
3. Another method to use is to simply insert text between the <APPLET> and </APPLET> tags. The inserted text could be a brief description of the applet and what is the function of the applet. This is beneficial for users who have applets turned off in their internet browser or their internet browser does not support JAVA applets.

Creating Accessible Tables

The Scope Method

(from Section 508 Standards, www.access-board.gov)

Using the "scope" attribute is one of the most effective ways of making HTML compliant with these requirements. It is also the simplest method to implement. The scope attribute also works with some (but not all) assistive technology in tables that use "colspan" or "rowspan" attributes in table header or data cells.

The first row of each table should include column headings. Typically, these column headings are inserted in <TH> tags, although <TD> tags can also be used. These tags at the top of each column should include the following attribute:

```
scope="col"
```

By doing this simple step, the text in that cell becomes associated with every cell in that column. Unlike using other approaches (notably "id" and "headers") there is no need to include special attributes in each cell of the table. Similarly, the first column of every table should include information identifying information about each row in the table. Each of the cells in that first column are created by either <TH> or <TD> tags. Include the following attribute in these cells:

```
scope="row"
```

By simply adding this attribute, the text in that cell becomes associated with every cell in that row. While this technique dramatically improves the usability of a web page, using the scope attribute does not appear to interfere in any way with browsers that do not support the attribute.

Example:

```
<table>
<tr>
<th>&nbsp;</th>
<th scope="col" >Spring</th> <th scope="col" >Summer</th> <th scope="col"
>Autumn</th> <th scope="col" >Winter</th> </tr>
<tr> <td scope="row" >Betty</td> <td>9-5</td> <td>10-6</td> <td>8-4</td><td>7-
3</td>
</tr>
<tr> <td scope="row" >Wilma</td> <td>10-6</td> <td>10-6</td> <td>9-5</td> <td>9-
5</td>
</tr>
<tr> <td scope="row" >Fred</td> <td>10-6</td> <td>10-6</td> <td>10-6</td> <td>10-
6</td>
</tr>
```

</table>

This table would be displayed as follows:

	Spring	Summer	Autumn	Winter
Betty	9-5	10-6	8-4	7-3
Wilma	10-6	10-6	9-5	9-5
Fred	10-6	10-6	10-6	10-6

The "ID" and "Headers" method

(from Section 508 Standards, www.access-board.gov)

The "id" and "headers" attributes require that every data cell in a table include special attributes for association. Although its usefulness for accessibility may have been diminished as browsers provide support for the "scope" attribute, the "id" and "headers" attributes are still very useful and provide a practical means of providing access in smaller tables.

The following table is much more complicated than the previous example and demonstrates the use of the "id" and "headers" attributes. Both methods provide a means of complying with the requirements for data tables in web pages. The table in this example includes the work schedules for two employees. Each employee has a morning and afternoon work schedule that varies depending on whether the employee is working in the winter or summer months. The "summer" and "winter" columns each span two columns labeled "morning" and "afternoon." Therefore, in each cell identifying the work schedule, the user needs to be told the employee's name (Fred or Wilma), the season (Summer or Winter), and the shift (morning or afternoon).

Example:

```
<table>
<tr>
<th>&nbsp;</th>
<th colspan="2" id="winter" >Winter</th>
<th colspan="2" id="summer" >Summer</th>
</tr>
<tr>
<th>&nbsp;</th>
<th id="am1" >Morning</th>
<th id="pm1" >Afternoon</th>
<th id="am2" >Morning</th>
<th id="pm2" >Afternoon</th>
</tr>
<tr>
<td id="wilma" >Wilma</td>
```

```

<td headers="wilma am1 winter" >9-11</td>
<td headers="wilma pm1 winter" >12-6</td>
<td headers="wilma am2 summer" >7-11</td>
<td headers="wilma pm2 summer" >12-3</td>
</tr>
<tr>
<td id="fred" >Fred</td>
<td headers="fred am1 winter" >10-11</td>
<td headers="fred pm1 winter" >12-6</td>
<td headers="fred am2 summer" >9-11</td>
<td headers="fred pm2 summer" >12-5</td>
</tr>
</table>

```

This table would be displayed as follows:

	Winter		Summer	
	Morning	Afternoon	Morning	Afternoon
Wilma	9-11	12-6	7-11	12-3
Fred	10-11	12-6	9-11	12-5

Summary Attribute:

Not all assistive technology products will interpret data tables using these options in the same manner, but data tables encoded in an accessible format will provide a greater opportunity for those needing assistive technology to gain access to on-line information. Using the **summary** attribute can also assist screen-reader users in understanding the layout and content presented in a table. It is important to note that not all assistive technology developers support this attribute at the current time.

Example:

1. Design the table using either the **scope** or **id** and **header** attributes.
2. After the <table-tag, and before the > sign, insert the attribute **summary="descriptive table summary goes here"**.
3. A good description of the table will include a brief identification of the content displayed in the row, the column, and the cell.
4. If possible, check the table and summary using screen-reader technology.

Evaluation and Repair Tools List

The following tools evaluate a page or a website for potential accessibility concerns. Standards for “accessible” range from the Web Accessibility Initiative (WAI) Web Content Accessibility Guidelines (v.1.0), and the Section 508 Standards of the Rehabilitation Act (effective June 21, 2001). These tools provide information covering how to fix flagged errors as well as provide methods to evaluate and "repair" a website. Bobby and A-Prompt are discussed in the next section in detail.

A-Prompt

<http://aprompt.snow.utoronto.ca/>

A-Prompt is a free program developed between the Adaptive Technology Resource Centre (ATRC) at the University of Toronto and the TRACE Center at the University of Wisconsin. A-Prompt uses the WAI Guidelines and Section 508 Standards to evaluate HTML code, suggests compliant HTML code syntax, and allows the developer to edit and replace code.

Bobby

<http://www.cast.org/bobby/>

Bobby is a free service to help Web page authors identify and repair significant barriers to access by individuals with disabilities. Enter the URL of the page that Bobby is to examine and click Submit. This dialog will only test one page at a time. You may download a version that will test an entire site.

InSight/InFocus

<http://www.ssbtechnologies.com/>

InSight performs automated diagnostics to determine web site accessibility. It tests for violations listed in the Web Content Accessibility Guidelines established by the World Wide Web Consortium. InFocus is a software tool that retrofits web sites, automatically making them accessible. These programs are designed for persons familiar with authoring web pages and provide tools to retrofit large websites.

VisCheck

<http://www.vischeck.com>

VisCheck is a free, on-line program that will evaluate your web page for color contrast and simulate what your site will look like to an individual with colorblindness. The download version requires the Photoshop software.

Bobby Evaluation Tool

Introduction

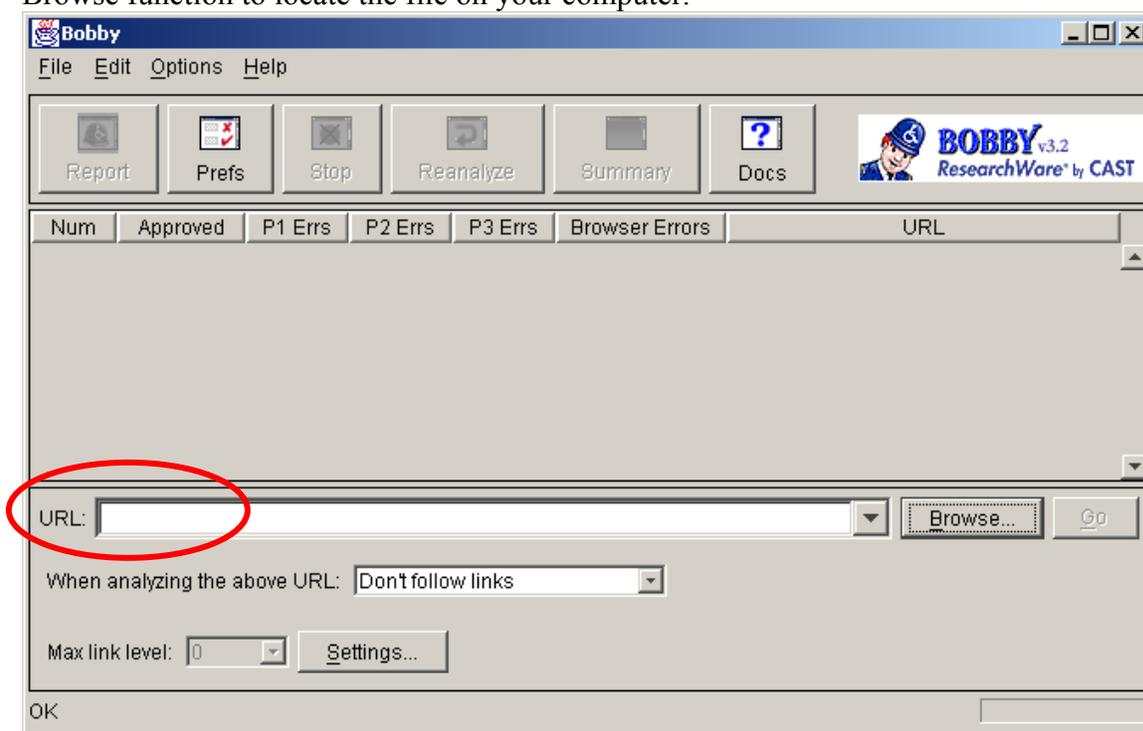
Bobby is a free product developed at the Center for Applied Special Technology for web authors. Bobby is a tool that will evaluate a web page or website and identify potential accessibility concerns that may impair the ability of an individual to access the website. There are two version of Bobby; a user may download the Bobby tool or access the Bobby website to perform an accessibility review. For extensive reviews or to evaluate pages from behind a firewall, it is recommended to use the download version of Bobby.

Bobby performs an evaluation of a site based on the Web Content Accessibility Guidelines 1.0 (WCAG) developed by the Web Accessibility Initiative (WAI). The advantage of Bobby is the library of information provided through the Bobby interface and, via hyperlinks, through the WAI website.

Evaluating a Site

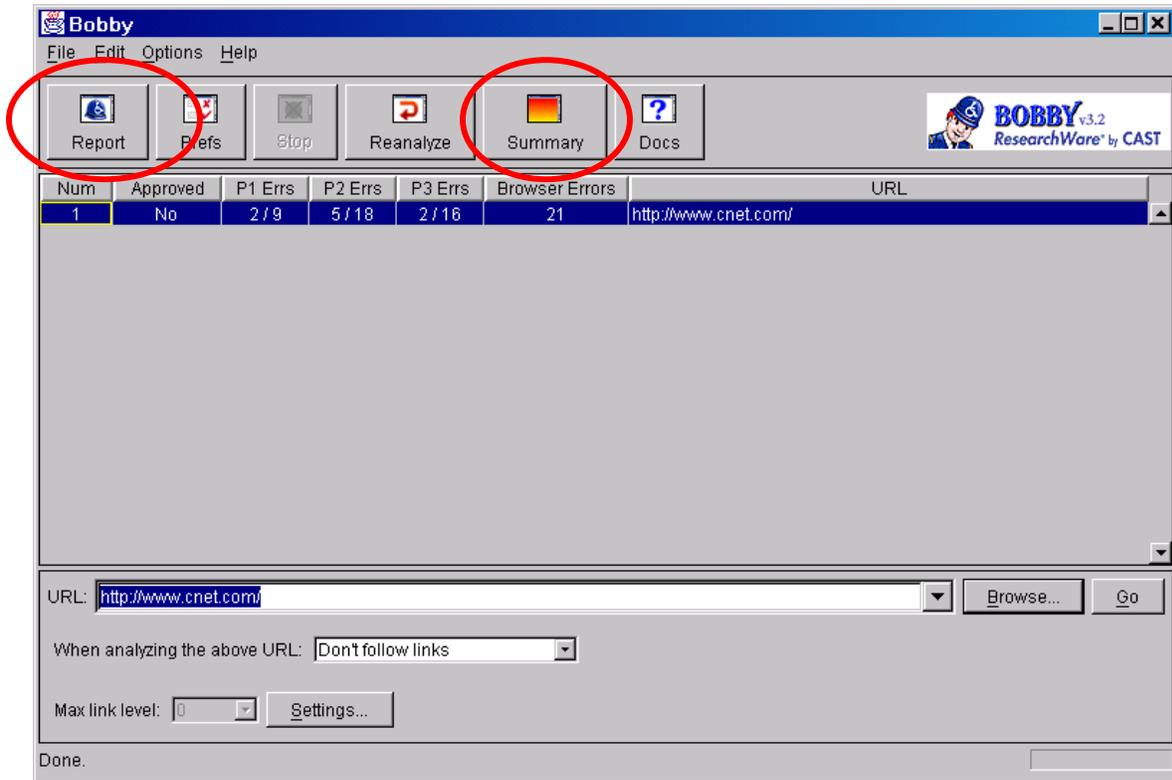
Download Bobby from the CAST website and install the program onto the computer. You can start Bobby through the Start menu and going to the Cast folder.

In the URL field, enter the address of the website that you wish to evaluate, or use the Browse function to locate the file on your computer.



Once you have entered in the necessary information, press the GO button.

The Report button will provide you with an accessibility report of the evaluated page.
The Summary button will provide you with an overview of the accessibility issues of the entire web site.



You can customize how "deep" into a website Bobby will analyze by adjusting the **When analyzing the above URL:** drop down menu. Additionally, you can designate the Priority level that Bobby uses to evaluate the site by selecting the **Prefs** button and choosing the various Priority levels.

A-Prompt Evaluation Tool

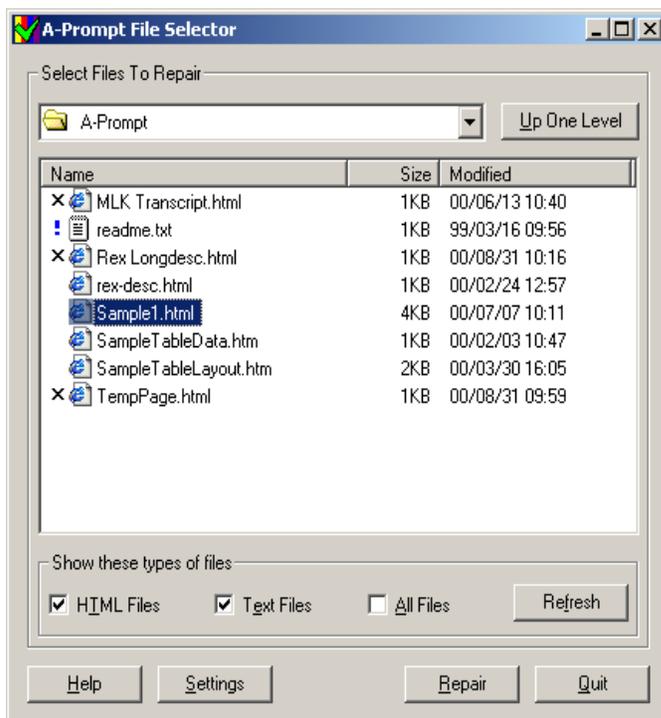
Introduction

A-Prompt (Accessibility-Prompt) is free evaluation tool developed by the Adaptive Technology Resource Centre at the University of Toronto and the TRACE Center at the University of Wisconsin. A-Prompt is designed to assist web authors in designing and developing accessible web page content by evaluating the HTML code against a list of potential accessibility issues. The most recent version of A-Prompt (ver. 1.0.5) contains an evaluation checklist that will compare a website's HTML code structure against the Section 508 Standards of the Rehabilitation Act and the Web Content Accessibility Guidelines developed by the Web Accessibility Initiative.

The A-Prompt website (<http://aprompt.snow.utoronto.ca/index.html>) contains further information regarding the development of A-Prompt as well as a brief overview of its functions. **You will need WinZip or similar program in order to install A-Prompt.**

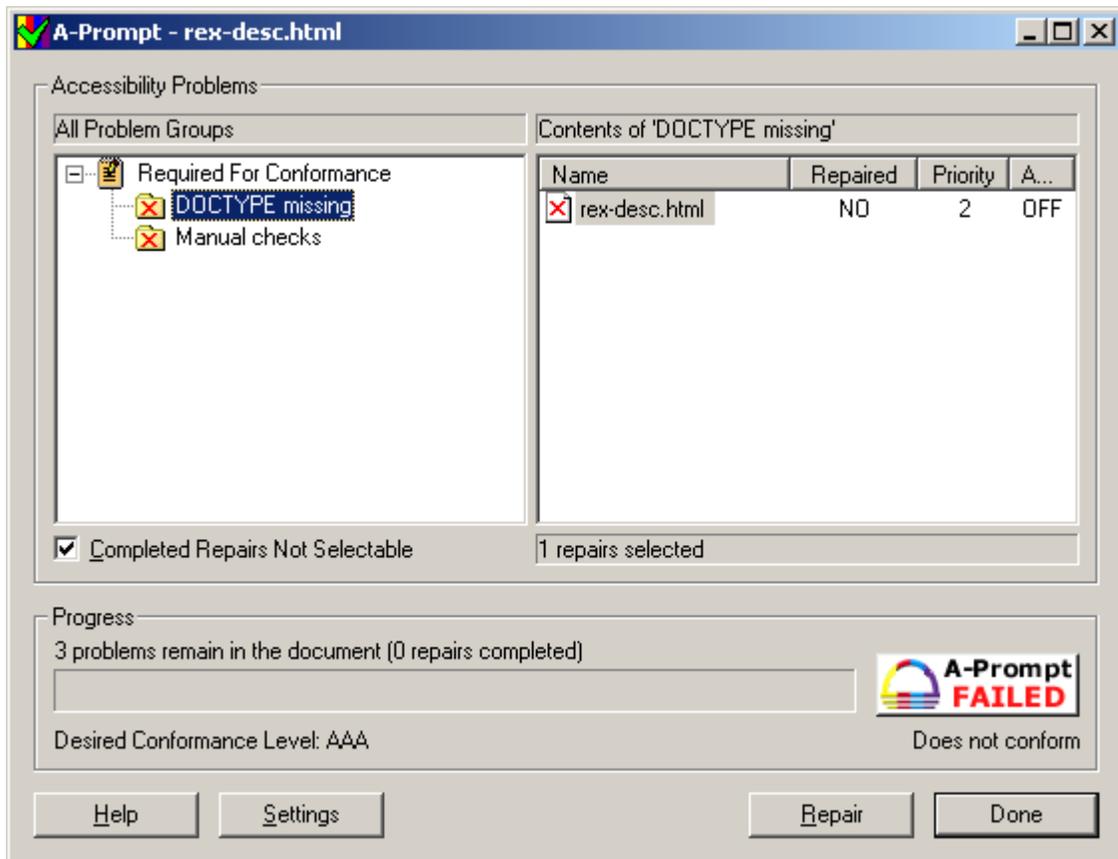
Evaluating a Site

A-Prompt begins with the **File Selector** interface. At this level, you can choose the .html files that you want to evaluate or browse to the file folder location your website is saved. To evaluate a file, select the appropriate file and click the Repair button.



At this level, you can also choose the conformance level at which the web pages will be evaluated. Click on the Settings button, and you can select either the WAI Priority Guidelines or the Section 508 Standards.

After selecting the file you wish to evaluate, select the accessibility problem in the left frame.



In the right frame, select the accessibility problem and click on the Repair button. This action will alert you to the changes necessary to make the web page accessible and prompt you for the necessary alterations. Once you have completed the accessibility improvements to the files, select the **Done** button. **Remember to save your corrected files after selecting the Done button.**

The Help button can provide more information as to specific questions or guidance in using the A-Prompt program.

Additional Web Resources

Accessible Web Design Guidelines

Microsoft: <http://www.microsoft.com/enable/dev/web/default.htm>

WebAIM: <http://www.webaim.org/howto/frontpage.php>

EASI: <http://www.isc.rit.edu/~easi/>

A-Prompt

<http://aprompt.snow.utoronto.ca/>

Bobby

<http://www.cast.org/bobby/>

InSight/InFocus

<http://www.ssbtechnologies.com/>

Section 508 Standards

<http://www.section508.org>

<http://www.access-board.gov>

Usability.gov

<http://www.usability.gov>

VisCheck

<http://www.vischeck.com>

Web Accessibility Initiative (WAI) Guidelines

<http://www.w3c.org/WAI/>

WebAIM

<http://www.webaim.org>

WebABLE

<http://www.webable.com/>

Macromedia

<http://www.macromedia.com/macromedia/accessibility/tools/contents.html>

<http://www.macromedia.com/macromedia/accessibility/>