

# **MATC Web Accessibility**

## ***508 Legal Standards***

### **A Brief Overview of Section 508 Standards**

**February 1, 2006**

# Accessibility Discussion Points — 508 Standards

## ***Web-Based Intranet and Internet Information and Applications (1194.22)***

The following discussion items are taken from The Access Board web site. This is the federal agency committed to accessible design.

The following are requirements that must be followed by Federal agencies when producing web pages. Since we receive federal funds, we must comply also.

The following points are taken from Section 1194.22 (a through p) of the 508 Standards.

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

#### **What is a text equivalent?**

A text equivalent means adding words to represent the purpose of a non-text element.

If you are using a non-text navigational action, the image must be accompanied by actual text that states the purpose of the image.

If the image is used to represent page content, the image must have a text description accompanying it that explains the meaning of the image.

Example: ``

#### **How much information actually needs to be in the text equivalent?**

When it is possible, communicate the same information as its associated element.

Web authors often use transparent images for spacing. For these graphics, use an empty alt attribute

Example: ``

#### **What is meant by the term, non-text element?**

A non-text element is an image, graphic, audio clip, or other feature that conveys meaning through a picture or sound.

#### **How should audio presentations be treated?**

If a presentation is audio only (not part of a multi-media presentation [audio & video]) a text transcript would meet the accessibility requirement.

However, if an audio presentation is part of a multimedia presentation, the audio portion must be captioned. Again, multimedia includes both audio and video.

#### **What are the ways to assign text to elements?**

Both the `<IMG>` (image tag) and `<APPLET>` (applet tag) can accept the “alt” attribute.

Using this attribute will enable web designers to include text that describes the picture directly in the `<IMG>` or `<APPLET>` tag.

Example: `<IMG src="images/logo1.gif" alt="Madison Area Technical College Logo" />`

The “alt” attribute in the <APPLET> tag only works for browsers that provide support for Java. If this is turned off in a browser the support for this will be turned off. A better alternative for the applet tag is to include the alternative text between opening and closing <APPLET> or <OBJECT> tags.

Example: <APPLET CODE=”myapplet.class” width=”200”, height=”200”>  
This applet displays current news for our college.  
</APPLET>

One other way to provide text description is to include it in the page in the surrounding context.

Example: <p>Below is a picture of me during my vacation!</p>  
<IMG src=”pictureofvacation.jpg” alt=”A picture of me on vacation” />

### ***b. Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.***

#### **What are considered equivalent alternatives?**

Captioning for the audio portion and audio description of visual information of multimedia presentations are considered the equivalent alternatives.

Captioning must be synchronized with the audio.

#### **If a web site offers audio files with no video, do they have to be captioned?**

Again, in this case the answer would be no because here it would not be considered multimedia. However, since audio is considered a non-text element, a text equivalent, such as a transcript, must be available.

Similarly, a (silent) web slide show presentation does not need to have an audio description accompanying it, but it does require text alternatives to be associated with the graphics.

### ***c. Web pages shall be designed so that all information conveyed with color is also available without color.***

#### **Why is this provision necessary?**

When colors are used as the sole method for identifying screen elements or controls, persons who are color blind as well as those people who are blind or have low vision may find a web page unusable.

#### **Does this mean that all pages have to be displayed in black in white?**

No, but it does require that some other method of identification, such as text labels, must be combined with the use of color. This also addresses the use of color to indicate an action.

Example: “Press the green button to start.” An additional way should identify the start besides using a color (e.g., labeling the button, adding an alt text to the button, etc.).

### ***d) Documents shall be organized so they are readable without requiring an associated style sheet.***

#### **What are the potential problems posed by style sheets?**

Users with low vision may create their own style sheet so that, regardless of what web page they visit, all text is displayed the way they need it to be displayed.

If designers set up their pages to override user-defined style sheets, people with disabilities may not be able to use those pages. For good access, it is critical that designers ensure that their web pages do not interfere with user-defined style sheets.

In general, the 'safest' and most useful form of style sheet are "external" style sheets, in which style rules are set up in a separate file.

Example: in your source code in the head section the following would be coded.

```
<link rel="stylesheet" type="text/css" href="section508.css">
```

Our style sheet is linked as follows:

```
<link rel="stylesheet" type="text/css" href="/matc/classes/default.cs">
```

### **What if I want to use a style that is not in the MATC default style sheet?**

It would be better if you create an additional style sheet and include it in your header above the current MATC style sheet. Even better would be to suggest the style to Web Strategies and ask them to place it in the main css for MATC. This way all web authors can use the style.

### ***e) Redundant text links shall be provided for each active region of a server-side image map.***

#### **How do "image maps" work?**

An 'image map' is a picture/map on a web page that provides different links to other web pages, depending on where the user clicks on the image.

There are 2 types of maps: client-side and server-side.

With client-side image maps, each "active region" in a picture can be assigned its own link (URL) that specifies what web page to retrieve when that portion of the picture is selected. HTML allows each active region to have its own alternative text, just like a picture.

With server-side image maps, clicking on the map only specifies the coordinates within the image when a mouse is depressed. The ultimate selection of the link or URL must be deciphered by the computer serving the web page.

#### **Why is this provision necessary?**

When a web page uses a server-side image map, browsers cannot indicate to the user the URL that will follow when a region of the map is activated. Therefore, the redundant text link is necessary to provide access to the page for anyone not able to see or accurately click on the map.

### ***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.***

#### **Why do client-side image maps provide better accessibility?**

Client-side image maps allow an author to assign text to each image map "hot spot".

Someone with a screen reader can easily identify and activate regions of the map.

Example:

```

```

```
<map name="Map">
```

```
<area shape="rect" coords="0,2,64,19" href="general.htm" alt="Information about us">
```

```
</map>
```

**g) Row and column headers shall be identified for data tables.**

**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.**

**Why are these two provisions necessary?**

They require that tables be coded according to the rules of the markup language being used for creating tables.

Large tables of data can be difficult to interpret if a person is using non-visual means to accessing the web. They can get easily lost. They need a method to associate the headings with each cell.

The use of the <pre> tag is not allowed. Web authors are also required to use a method to provide an association between a header and its related information. There are several methods.

**How can HTML tables be made readable with assistive technology?**

Using the “scope” attribute is the most effective and simplest way of making tables compliant.

**Using the scope attribute**

The first row of each table should include column headings by using the <TH> tag. These tags should use the following attribute: scope=“col”. By doing this, the text in the cell becomes associated with every cell in that column. There is no need to include special attributes.

By adding the scope=“row” to the next set of <TD> tags, you will associate the text in that cell will all in that row.

Example:

```
<table>
<tr><th>&nbsp;</th><th scope=“col”>Spring</th><th scope=“col”>Summer</th></tr>
<tr><td scope=“row”>Betty</td><td>9-5</td><td>10-6</td></tr>
<tr><td scope=“row”>Wilma</td><td>10-6</td><td>9-5</td></tr>
</table>
```

	<b>Spring</b>	<b>Summer</b>
Betty	9-5	10-6
Wilma	10-6	9-5

**Other ways to make your pages accessible**

Use the “ID” and “Header” attributes in tables. You can also use the summary attribute but it is not sufficiently supported by major assistive technology manufacturers to warrant recommendation.

**i) Frames shall be titled with text that facilitates frame identification and navigation.**

**Why is this provision necessary?**

Frames can present difficulties for users with disabilities when frames are not easily identifiable to assistive technology. Users with disabilities may become lost if the difference between the two frames is not clearly established.

**It is recommended to not use frames at all.**

If you have existing content that is in frames and it is not feasible to make changes at this time, make sure that the frames are clearly identified.

Include text within the body of each frame that clearly identifies the frame. Additionally, using meaningful text in the <frame> tag by using the “title” attribute will allow a description of the frame as a “quote-enclosed string.

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

#### **Why is this provision necessary?**

Users can have a seizure triggered by displays that flicker, flash, or blink, particularly if the flash has a high intensity and is within certain frequency ranges.

#### **How can flashing or flickering elements be identified?**

Flashing or flickering elements are usually added through technologies such as animated gif's, Java applets, or third-party plug-ins or applications.

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

#### **What must a text-only page contain to comply with this provision?**

Text-only pages must contain equivalent information or functionality as the primary pages. Also, the text-only page shall be updated whenever the primary page changes.

***l) When pages utilize scripting languages 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.***

#### **What accessibility problems can scripts cause?**

Web authors are responsible for providing script information in a fashion that can be read by assistive technology. If it is not done properly, a screen reader will often read the content of the script itself in a meaningless jumble of numbers and letters. It will not be interpreted or used.

#### **How can web developers comply with this provision?**

Typically, using Javascript as part of the <a> tag is an easy way to invoke Javascript functions.

Example: <a href="javascript:myFunction();" > Start myFunction</a>. This technique doesn't typically cause issues with assistive technologies but when it is used surrounding an image, requiring a user to click an image to start the function.

Some developers suggest using the "title" attribute in the <a> tag to help with this issue.

Example: <a title="this link starts myFunction" href="javascript:myFunction();" ></a>

This tag is supported by some but not all assistive technologies.

Another issue occurs when web developers rewrites the text in the status line. This is difficult or impossible to detect with a screen reader. Although rewriting the status line does not necessarily always interfere with the accessibility or inaccessibility of the JavaScript URL, web developers should ensure that all important information conveyed in the status line also be provided through the "alt" attribute.

Example: <a href="javascript:myFcn();" onmouseover="status='Nice Choice'; return true;" onmouseout="status=";" ></a>

JavaScript uses so-called "event handlers" as a trigger for certain actions or functions to occur. For instance, a web developer may embed a JavaScript function in a web page that automatically checks the content of a form for completeness or accuracy.

Web developers must exercise caution when deciding which event handlers to use in their web pages, because different screen readers provide different degrees of support for different event handlers.

### **Recommendations for popular event handlers**

onClick – This can work well, but do not use the onClick event handler for form elements that include several different options (e.g., select lists, radio buttons, checkboxes) unless absolutely necessary.

onDbClick – This creates an accessibility problem because it can be confusing to users and should be avoided.

onMouseDown and onMouseUp – This should be used sparingly, if at all, because it can be quite confusing. Developers should opt for onClick instead of the onMouseDown.

onMouseOver and onMouseOut – These events handlers neither can be accessed by the mouse nor interfere with accessibility – a screen reader simply bypasses them entirely. Web developers who use these event handlers should be careful to duplicate the information provided by these event handlers through other means.

onLoad and onUnload – these do not present accessibility problems.

onChange – This is common for JavaScripts based on a selection from within a <select> tag. It presents tremendous accessibility problems for many commonly used screen readers and should be avoided. Web developers should use the onClick event handler to accomplish the same function.

onBlur and onFocus – They are not common. While they don't necessarily cause an issue, their behavior can be confusing enough to a web page visitor that they should be avoided.

***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 the standards.***

#### **Why is this provision necessary?**

This provision requires that web pages provide a link to a plug-in that will meet the software provisions. MATC hosts a web page with all the necessary links for plug-ins. It is recommended that you link to that page for users to download the necessary plug-in.

***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 cues.***

#### **Why do electronic forms present difficulties to screen readers?**

The interaction between form controls and screen readers can be unpredictable. HTML forms pose accessibility problems when web developers separate a form element from its associated label or title.

The use of tables separating the label and field clearly shows the accessibility problem inherent in forms.

When the relationship between the titles "First Name" or "Last Name" and their respective input boxes may be obvious from a visual inspection, the relationship is not obvious to a screen reader. Instead, a screen reader may simply announce "input box" when encountering each input box. The form elements can become separated from their labels when using a table structure for your form.

#### **How can developers provide accessible HTML forms?**

Developers should place labels adjacent to input fields, not in separate cells of a table.

Some developers that continue to use tables, use the <label> tag mark specific elements as "labels" and then associate a form element with that label.

```
<FORM>
<TABLE>
<TR>
<TD><B><LABEL FOR="first">First Name:</LABEL></B></TD>
<TD><INPUT TYPE="TEXT" NAME="FIRSTNAME" ID="first"></TD>
</TR>
</TABLE>
```

**Avoid** the following use of the <LABEL> tag because it is not supported by all screen readers.

```
<LABEL>
<TR>
<TD><B>FIRST NAME:</B></TD>
<TD><INPUT TYPE="TEXT" NAME="FIRSTNAME"></TD>
</TR>
</LABEL>
```

***o) A method shall be provided that permits users to skip repetitive navigation links.***

**Why do navigational links present impediments to screen readers and other types of assistive technologies?**

This provision provides a method to facilitate the easy tracking of page content that provides users of assistive technology the option to skip repetitive navigation links.

MATC has set "skip navigation" within the MATC public web site.

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

**Why do timed responses present problems to web users with disabilities?**

Sometimes web pages expire if a response is not received within a specific amount of time. Someone's disability can have a direct impact on the speed with which he or she can read, move around, or fill in a web form. When a timed response is required, the user shall be alerted via a prompt and given sufficient time to indicate whether additional time is needed.