STATE of ARIZONA

Government **I**nformation **T**echnology **A**gency	**Statewide** **POLICY** **P130 Rev 1.0**	**TITLE: Web Site Accessibility** **Effective Date: September 15, 2006**

1. AUTHORITY

The Government Information Technology Agency (GITA) shall develop, implement and maintain a coordinated statewide plan for information technology (IT) (A.R.S. § 41-3504(A (1))) including the adoption of statewide technical, coordination, and IT Policy and Standards (A.R.S. § 41-3504(A (1(a)))).

2. PURPOSE

The purpose is to establish a statewide policy that provides an accessibility model for the development and implementation of Arizona State government web sites that minimizes technical barriers to accessibility for individuals with disabilities.

3. SCOPE

This applies to all budget units. Budget unit is defined as a department, commission, board, institution or other agency of the state organization receiving, expending or disbursing state funds or incurring obligations of the state including the Arizona Board of Regents but excluding the universities under the jurisdiction of the Arizona Board of Regents and the community college districts and the legislative or judicial branches. A.R.S. § 41-3501(2).

The Budget Unit Chief Executive Officer (CEO), working in conjunction with the Budget Unit Chief Information Officer (CIO), shall be responsible for ensuring the effective implementation of Statewide Information Technology Policies, Standards, and Procedures (PSPs) within each budget unit.

4. POLICY

To provide an accessibility model in which web content authors, format designers, and software developers within budget units understand their roles in providing persons with disabilities, access to existing and developing State web sites. This model is designed for the general public with disabilities who are able to handle general-purpose web content under ordinary operating conditions, see ***Attachment A, Web Site Accessibility Technical Suggestions and Examples***, for further specifics.

The information presented in Attachment A is designed to assist and aid budget unit webmasters in developing accessible web pages. Because of vast changes and improvements in information technologies, budget unit webmasters are encouraged to use other available technical methods that will promote equivalent or greater access to and use of websites in being compliance with Section 508 (see page 31). This accessibility model is not intended to cover other accessibility issues related to Section 508.

The accessibility model consists of the following web page features that are to be addressed for all State web sites:

4.1 **GRAPHICS**
Simple images; linked images; content images; graphical text; ASCII Art; list bullets; spacer images; animated graphics–text equivalent; animated graphics-frame rate.

4.2 **INFORMATION IN COLOR**
Information in color; color contrast.

4.3 **MOVING CONTENT**
Static background color; moving text.

4.4 **DOWNLOADABLE FILES**
Graphics in downloadable files; PDF files.

4.5 **REPETITIVE CONTENT**
Skipping navigation links.

4.6 **DATA TABLES**
Identifying row and column headers; using SCOPE to group table cells.

4.7 **MULTIMEDIA**
Sounds 10; speech-short clips (up to 60 seconds); speech-long clips; video-short clips (up to 10 seconds); video-long clips.

4.8 **IMAGE MAPS**
Image map graphics-text equivalent; client-side image map regions; server-side image map regions.

4.9 **STYLE SHEETS**
Style sheets.

4.10 **FORMS**
Forms-label placement; forms-associating labels and controls; forms-time responses.

4.11 **SCRIPTS**
Scripts-text equivalents; scripts-keyboard accessibility.

4.12 **APPLETS and PLUG-INS**
Applets and plug-ins-links; applets and plug-ins-text equivalent information or functionality.

4.13 **FRAMES**
Frames-labeling; frames-NOFRAMES elements.

4.14 **KEYBOARD CONTROL**
Keyboard control.

4.15 **TEXT-ONLY PAGES**
Text-only versions.

4.16 **BIOMETRIC IDENTIFICATION**
Alternative identification.

4.17 **EQUIVALENT FACILITATION**
Equivalent access.

5. RESPONSIBILITY

It is the responsibility of each budget unit to remain current with the development of accessible information technology through their ADA/508 Coordinators.

6. KNOWN LIMITATIONS OF THIS POLICY

Individuals with (or without) disabilities access the Web with widely varying sets of capabilities, software, and hardware. While this policy does provide the ability for individuals with disabilities to access Web based information, this policy does not include requirements for the production of every known accessibility need, due to known limitations on existing technology. The limitations of this policy are as follows:

6.1 Braille – this policy does not address Braille production.
6.2 Synthesized Speech production – not covered by this policy.
6.3 Input Modalities – no voice input, only keyboard and pointing devices.

7. REFERENCES

7.1. A. R. S. § 1-621 et seq., “Purchase of Insurance; coverage; limitations, exclusions; definitions.”
7.2. A. R. S. § 41-1335 (A (6 & 7)),“State Agency Information.”
7.3. A. R. S. § 41-1339 (A),“Depository of State Archives.”
7.4. A. R. S. § 41-1461, “Definitions.”
7.5. A. R. S. § 41-1463, “Discrimination; unlawful practices; definition.”
7.6. A. R. S. § 41-1492 et seq., “Prohibition of Discrimination by Public Entities.”
7.7. A. R. S. § 41-2501 et seq., “Arizona Procurement Codes, Applicability.”
7.8. A. R. S. § 41-3501, “Definitions.”
7.9. A. R. S. § 41-3504, “Powers and Duties of the Agency.”
7.10. A. R. S. § 41-3521, “Information Technology Authorization Committee; members; terms; duties; compensation; definition.”
7.11. A. R. S. § 44-7041, “Governmental Electronic Records.”
7.12. Arizona Administrative Code, Title 2, Chapter 7, “Department of Administration Finance Division, Purchasing Office.”
7.13. Arizona Administrative Code, Title 2, Chapter 10, “Department of Administration Risk Management Section.”
7.14. Arizona Administrative Code, Title 2, Chapter 18, “Government Information Technology Agency.”
7.15. Statewide Information Technology Policy P100.
7.16. Digital Millennium Copyright Act (1998).
7.17. Title 17 United States Code.
7.18. Title 18 United States Code § 2319 & 2320.
7.19. Federal Rehabilitation Act, Section 508.
7.20. Telecommunications Act of 1996, Section 255
7.21. State of Arizona Technology Related Assistance for Individuals with Disabilities Act, Section 103.

8. ATTACHMENTS

Attachment A. Web Site Accessibility Technical Suggestions and Examples

(Attachment A, provides technical suggestions that are cited as possible good software designs and practices and does not profess to be a complete and final document. This is due to a variety of other technical methods and solutions that webmasters may want to pursue to implement the accessibility model).

Attachment A

Web Site Accessibility Technical Suggestions and Examples

State agencies may use a variety of technical methods to be in compliance with Section 508, (see page 31). The information presented in this Appendix is designed to assist the webmaster in writing accessible web pages.

For additonal information please see:

Section 508/Federal IT Accessibility Initiative
http://www.section508.gov/

World Wide Web Consortium Guidance
http://www.w3.org

Tools & Resources
http://www.section508.gov/index.cfm

Department of Justice, Section 508 Self Evaluation
http://www.usdoj.gov/crt/508/web2.htm

Graphics

Section 508 Standards related to suggestions in this section

> **Section 508, 1194.22(a)**
> *“A text equivalent for every non-text element shall be provided (e.g., via “alt,” “longdesc,” or in element content).”*
>
> **Section 508, 1194.22(j)**
> *“Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.”*

Related Definitions

Non-text element: Any element on a Web page that is graphical in nature and/or cannot be accessed using a text-only browser or screen reader (e.g., images, sounds, JavaScript).

Text equivalent: Meaningful text substitute for a non-text item; the length and content of the text equivalent depends on the type of non-text item being represented.

Brief description (use `alt` attribute): A description of up to twelve words.

Detailed description (use `longdesc` attribute): A description longer than twelve words.

ASCII Art: Pictures created by piecing together the characters commonly found on a computer keyboard.

Suggestion 1: Simple Images

Related Section 508 Suggestions: 1194.22(a)

For each image that conveys information regarding page structure, (e.g. colored lines, page breaks, and borders), page content, or site navigation, a brief description should be provided using the `alt` attribute of the `<img>` element.

Example

```
<img src=”governor.jpg” alt=”Photograph of Arizona’s Governor”>
```

Suggestion 2: Linked Images

Related Section 508 Standards: 1194.22(a)

For linked images, a brief description of the destination or purpose of the link should be provided using the `alt` attribute of the `<img>` element.

Related Section 508 Standards: 1194.22(a)

Example

```
<a href=”/”>
<img src=”homebutton.gif” alt=”Return to home page”>
</a>
```

Suggestion 3: Content Images

Related Section 508 Standards: 1194.22(a)

For images such as graphs and charts that convey information, equivalent alternative text should be provided:

1. Use the `alt` attribute of the `<img>` element and give a brief summary of the content.

2. A detailed description of the image content should also be provided to allow non-sighted users to access the information available to sighted users. Two possible methods for providing the detailed information on the image are:

    - In the text next to (or surrounding) the image, or

    - On a separate page accessible via a link to a detailed description. The link should be placed adjacent to the images.

    - For multiple descriptions use an indexed separate page with appropriate detailed descriptions.

Technical Explanation

The example below demonstrates the use of the longdesc attribute and the adjacent link. Both techniques perform similar functions, but the longdesc attribute is not yet widely supported by the installed base of Web browsers. Historically, the adjacent link to a detailed description has been indicated with the text “D” (thus the term “D-link”); however, text such as “detailed description” may be more readily understood. It may be appropriate to use the image itself as the link to its detailed description.

In the source document, economics.html:

```
<img src=”figure1.gif” alt=”Illustration of the relationship
   between supply and demand” longdesc=”chartdesc.html#figure1”>
<a href=”chartdesc.html#figure1”>(detailed description)</a>
<a name=”figure1return”> </a>
```

In chartdesc.html, which happens to be a single document containing the descriptions of all charts and figures in economics.html:

```
<a name=”figure1”> </a>
Figure 1 demonstrates the relationship between supply and demand.
   Figure 1 is a chart showing a diagonal line beginning in the
   upper left corner and ending in the lower right corner. This
   indicates an inverse relationship. As supply increases, demand
   decreases. As supply decreases, demand increases.
<a href=”economics.html#figure1return”>Return to the paper</a>
```

Suggestion 4: Graphical Text

Related Section 508 Standards: 1194.22(a)

For a graphic that is just a bitmap of text, provide the exact text equivalent for the text using the `alt` attribute of the `<img>` element. Where a special markup language for symbols exists—such as MathML—use that instead of a simple text equivalent.

Example

```
<img src=”button_visitors.gif” alt=”Visitor Information Center”>
```

Suggestion 5: ASCII Art

Related Section 508 Standards: 1194.22(a)

ASCII art cannot be read by screen readers and is not to be used except as part of a graphic. A description of the graphic and any text content should be provided using the `alt` attribute of the `<img>` element. The ASCII art symbols (e.g., “XXXX”) should not be used for `alt` text.

Not recommended

```
I like my job. :-)
```

Not recommended

```
I like my job. <img src=”smiley.gif” alt=”:-)”>
```

Recommended

```
I like my job. <img src=”smiley.gif” alt=”(smile)”>
```

Suggestion 6: List Bullets

Related Section 508 Standards: 1194.22(a)

When images are used as structural elements in the place of list bullets, an asterisk, “*”, should be used as the text equivalent in the `alt` attribute of the `<img>` element.

Example

```
Remember that the experience of browsing the Web with a screen
   reader is: <br>
<img src=”bullet.gif” alt=”*”>linear<br>
<img src=”bullet.gif” alt=”*”>time-consuming<br>
<img src=”bullet.gif” alt=”*”>relatively context-free<br>
```

Suggestion 7: Spacer Images

Related Section 508 Standards: 1194.22(a)

For transparent images used to control spacing in page layouts, “ “ (a blank space) should be given as the text equivalent in the `alt` attribute of the `<img>` element.

Example

```
<img src=”spacer.gif” alt=” ”>
```

Suggestion 8: Animated Graphics—Text Equivalent

Related Section 508 Standards: 1194.22(a)

For animated images (GIFs), a text equivalent can be provided using the `alt` attribute of the `<img>` element. The text equivalent should include the following information:

1. A description of the graphic including color (e.g., “Calico cat on a green background”).

2. A description of the movements/actions (e.g., “brushing her face with her paw”).

3. Any text or information content provided by the graphic.

Example

```
<img src=”kitty.gif” alt=”Keeping it clean. Image shows a calico
   cat on a green background, brushing her face with her paw.”>
```

Suggestion 9: Animated Graphics—Frame Rate

Related Section 508 Standards: 1194.22(j)

Ensure that the frame rate of animated GIFs is lower than 2 fps or higher than 55 fps. Do not “stack” frames within those GIFs in such a way that the image will change at a rate between 2 and 55 fps.

Information in Color

Section 508 Standards related to standards in this section

Section 508, 1194.22(c)
“Web pages shall be designed so that all information conveyed with color is also available without color, for example, from context or markup.”

Suggestion 10: Information in Color

Color alone should not be used to convey information or require action.

Related Section 508 Standards: 1194.22(c)

Not recommended

```
Required fields are indicated in red:<br>
<font color=”#ff0000”>First name:</font>
```

Recommended

```
Required fields are indicated in red and marked with a *:<br>
<font color=”#ff0000”>First name (*):</font>
```

Suggestion 11: Color Contrast

Related Section 508 Standards: 1194.22(c)

Contrast between foreground and background colors should be sufficient to allow the page to be read when printed on a black-and-white printer.

Technical Explanation

For further information regarding colorblindness and selecting color palettes that will accommodate individuals with color deficiencies, including tools for testing your site’s color contrast, see http://www.tessPub.com/colours.html or http://www.labs.bt.com/people/rigdence/colours/ or http://www.lighthouse.org/color_contrast.htm

Moving Content

Section 508 Standards related to standards in this section

Section 508, 1194.22(a)
“A text equivalent for every non-text element shall be provided (e.g., via “alt,” “longdesc,” or in element content).”

Section 508, 1194.22(j)
“Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.”

Also See Standard 9: Animated Graphics—Frame Rate

Suggestion 12: Static Background Color

Related Section 508 Standards: 1194.22(j)

Avoid “stacking” body tags to change a document’s background color.

Technical Explanation

The “stacked body tag” technique (demonstrated below) causes the screen to flicker at an unsafe rate.

Example (Not Recommended)

```
<body bgcolor=”#ff0000”>
<body bgcolor=”#cc0000”>
<body bgcolor=”#990000”>
<body bgcolor=”#660000”>
<body bgcolor=”#330000”>
<body bgcolor=”#000000”>
<!-- This technique, used to change background colors on a page,
   causes the screen to flicker at an unsafe rate. -->
```

Suggestion 13: Moving Text

Related Section 508 Standards: 1194.22(a)

A non-moving text equivalent should be provided for all moving text.

Technical Explanation

Users of assistive technology cannot access moving text. See the Scripts and Applets and Plug-in sections for information on providing text equivalents for moving text generated using these methods.

Downloadable Files

Section 508 Standards related to standards in this section

Section 508, 1194.22(a)
"A text equivalent for every non-text element shall be provided (e.g., via "alt," "longdesc," or in element content)."

Suggestion 14: Graphics in Downloadable Files

Related Section 508 Standards: 1194.22(a)

For downloadable files containing tables, charts and other information conveyed using graphics, in-text descriptions of graphics should be provided (see Standard 3, Content Images).

Suggestion 15: PDF Files

Related Section 508 Standards: 1194.22(a)

PDF files should be translatable into HTML using the Adobe Access plug-in available from the Adobe Website. PDF files may not be usable with assistive technologies even if translatable; therefore, provide an additional copy of the file in an alternate format that is usable by assistive technologies.

Definition

Alternate formats that are usable by assistive technologies may include, but are not limited to, Braille and Nemeth Code Braille for math and symbolic information, ASCII text, large print, recorded audio, and/or electronic formats that comply with Section 508, 1194.21 (a) through (l).

Repetitive Content

Section 508 Standards related to standards in this section

Section 508, 1194.22(o)
“A method shall be provided that permits users to skip repetitive navigation links.”

Suggestion 16: Skipping Navigation Links

Related Section 508 Standards: 1194.22(o)

A method (such as the use of transparent GIFs linked to HTML bookmarks) should be provided to allow users with assistive technology to bypass repetitive navigation links.

Example

```
<body>
<a href=”#content”>
<img src=”spacer.gif” height=”0” width=”0” alt=”Skip to the
   content of this page”>
</a>
<!-- Navigation links and other repetitive content go here. -->
<a name=”content”> </a>
<!-- Page content starts here. -->
```

Data Tables

Section 508 Standards related to standards in this section

Section 508, 1194.22(g)
“Row and column headers shall be identified for data tables.” [Note: this does not apply to layout tables.]

Section 508, 1194.22(h)
“Mark-up 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.”

Suggestion 17: Identifying Row and Column Headers

Related Section 508 Standards: 1194.22(g)

Data table column and row headings should be identified using `<th>` elements. Note: `<th>` elements should not be used in tables that are used strictly for visual layout.

Example: See Standard 18 example.

Suggestion 18: Using the scope Attribute to Group Table Cells

Related Section 508 Standards: 1194.22(h)

When there are two or more logical levels of row or column headers, HTML mark-up should be used to associate data and header cells.

Example for a Simple Table

```
<table>
<tr>
<th scope=”col”>Campus</th>
<th scope=”col”>Year</th>
<th scope=”col”>Enrollment</th>
<th scope=”col”>Freshman enrollment</th>
</tr>
<tr>
<th scope=”row”>ASU East</th>
<td>1999</td>
<td>1500</td>
<td>600</td>
</tr>
<th scope=”row”>ASU East</th>
<td>2000</td>
<td>1900</td>
<td>750</td>
</tr>
<!-- ... -->
</table>
```

Note: Complex tables may require additional research.

Multimedia

Section 508 Standards related to standards in this section

Section 508, 1194.22(b)
"Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation."

Section 508, 1194.22(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 §1194.21(a) through (l)."

Suggestion 19: Sounds

Related Section 508 Standards: 1194.22(b), 1194.22(m)

For sounds (e.g., wav files), the text equivalent of the sound should be provided using the `alt` attribute of the `<a href>` element.

If a sound conveys a message (such as "uh oh" when a user tries to submit a form), also send an equivalent text message to the screen.

For linked sounds (e.g., wav files), the text equivalent of the sound will be provided using the `alt` attribute of the `<a href>` element. For embedded sounds, the text equivalent of the sound will be provided within the `<object>` element (i.e., between the opening and closing `<object>` tags). If the clip requires a plug-in, the text equivalent should include a link to obtain the plug-in.

Examples

For linked sounds:

```
<a href=”gotmail.wav” alt=”You’ve got mail!”>Mail reminder</a>
```

For embedded sounds:

```
<object data=”gotmail.wav” codetype=”audio/basic”>
You’ve got mail!
</object>
```

Suggestion 20: Speech—Short Clips (up to 60 seconds)

Related Section 508 Standards: 1194.22(b), 1194.22(m)

For short message clips, text equivalents should be placed on the page next to the sound file hyperlink.

For linked short message clips, text equivalents should be placed on the page next to the sound file hyperlink. For embedded short message clips, text equivalents will be placed within the `<object>` element (i.e., between the opening and closing `<object>` tags). If the clip requires a plug-in, the text equivalent should include a link to obtain the plug-in.

Examples

For linked sounds:

```
<a href=”govmessage.wav”>A message from the Governor</a>: Welcome
   to the State of Arizona Website! We are continually working to
   improve this site and make the information easily accessible.
   You can access all aspects of State Government through these
   pages—including the Governor’s Office.
```

For embedded sounds:

```
<object data=”govmessage.wav” codetype=”audio/basic”>
A message from the Governor</a>: Welcome to the State of Arizona
   Website! We are continually working to improve this site and
   make the information easily accessible. You can access all
   aspects of State Government through these pages—including the
   Governor’s Office.
</object>
```

Suggestion 21: Speech—Long Clips

Related Section 508 Standards: 1194.22(b), 1194.22(m)

For longer clips, a synchronized text equivalent should be provided.

The text equivalent should contain: 1) a word-for-word transcript and 2) descriptions of any sounds that will help users understand content.

If a plug-in is required to access the clip and/or the synchronized text equivalent, provide a link to the plug-in within the `<object>` element used to embed the clip—or, if the clip is linked rather than embedded, near the clip (not within the text equivalent).

Technical Explanation

Use SMIL, Quicktime, or SAMI to synchronize the text equivalent/transcript with the sound. RealNetworks’ RealProducer (http://www.realnetworks.com/products/producerplus) and WGBH’s MAGpie (http://ncam.wgbh.org/Webaccess/magpie) are two of many tools that will do this.

Suggestion 22: Video—Short Clips (up to 10 seconds)

Related Section 508 Standards: 1194.22(b), 1194.22(m)

For short video clips, text equivalents will be placed on the page next to the video file hyperlink.

For linked short video clips, text equivalents will be placed on the page next to the video file hyperlink. For embedded short video clips, text equivalents will be placed within the `<object>` element (i.e., between the opening and closing `<object>` tags). If the clip requires a plug-in, the text equivalent should include a link to obtain the plug-in.

Example

For linked video:

```
<a href=”watchyourcar.mov”>Watch Your Car</a>: The video clip
   shows a car on the freeway, late at night. A siren sounds, the
   car screeches to a halt on the side of the road, and a police
   officer walks up to the car requesting driver ID and
   registration information. The video cuts to a man standing
   beside the car with his back to us and hands cuffed. The
   camera zooms in on a “Watch Your Car” decal on the window next
   to him.
<br>
<a href=”http://www.apple.com/quicktime/”>(QuickTime plug-in
   required.)</a>
```

For embedded video:

```
<object data=”watchyourcar.mov” codetype=”video/quicktime”>
Watch Your Car: The video clip shows a car on the freeway, late
   at night. A siren sounds, the car screeches to a halt on the
   side of the road, and a police officer walks up to the car
   requesting driver ID and registration information. The video
   cuts to a man standing beside the car with his back to us and
   hands cuffed. The camera zooms in on a “Watch Your Car” decal
   on the window next to him.
<br>
<a href=”http://www.apple.com/quicktime/”>This video clip
   requires the QuickTime viewer or plug-in.</a>
</object>
```

Suggestion 23: Video—Long Clips

Related Section 508 Standards: 1194.22(b), 1194.22(m)

For longer tracks, a synchronized text equivalent should be provided. In addition, the text equivalent should also be provided as a separate document for users who cannot access media files.

If a plug-in is required to access the clip and/or the synchronized text equivalent, provide a link to the plug-in within the `<object>` element used to embed the clip—or, if the clip is linked rather than embedded, near the clip (not within the text equivalent).

1. The text equivalent should contain:

    - a word-for-word transcript,

    - descriptions of any sounds required to understand the content,

    - descriptions of other information including actions, settings, and characters necessary to understand the video.

2. An audio description (descriptive video) of visual content should be provided.

Technical Explanation

Use SMIL, Quicktime, or SAMI to synchronize the text equivalent/transcript with the video. RealNetworks' RealProducer (http://www.realnetworks.com/products/producerplus) and WGBH's MAGpie (http://ncam.wgbh.org/Webaccess/magpie) are two of many tools that will do this.

Image Maps

Section 508 Standards related to standards in this section

Section 508, 1194.22(e)
“Redundant text links shall be provided for each active region of a server-side image map.”

Section 508, 1194.22(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.”

Suggestion 24: Image Map Graphics—Text Equivalent

Related Section 508 Standards: 1194.22(a)

A brief description should be provided for all image-map images using the `alt` attribute of the `<img>` element (for client-side image maps) or the `<a href>` element (for server-side image maps).

Example: See Standard 25 and Standard 26 examples.

Suggestion 25: Client-Side Image Map Regions

Related Section 508 Standards: 1194.22(e), 1194.22(f)

A brief description should be provided for each hotspot region on a client-side image-map graphic using the `alt` attribute of the `<a href>` element.

Example

```
<img src=”/images/map.gif” usemap=”#campusmap” alt=”Select a
   building from this campus map for more information.”>
<map name=”rightmap”>
<area shape=”rect” alt=”Academic Center Building”
   coords=”14,215,40,251” href=”cntr.html”>
<area shape=”rect” alt=”Flight Simulator Building”
   coords=”259,68,307,99” href=”sim.html”>
<area shape=”rect” alt=”Swimming Pool/Recreation”
   coords=”187,260,222,290” href=”pool.html”>
<!-- ... -->
</map>
```

Suggestion 26: Server-Side Image Map Regions

A redundant listing of all hotspot links within a server-side image map should be placed next to the map.

Related Section 508 Standards: 1194.22(e), 1194.22(f)

Technical Explanation

In the example which follows, the server-side application gismap.cgi accepts x and y coordinates (from a mouse click on a US map) as parameters, determines the state from those coordinates, and processes output accordingly. The example assumes that gismap.cgi will also accept input in the form of a state code. If the server-side application did not accept alternate input as in this example, an alternative would be to supply x and y coordinates that fall within the state (or county, or whatever geographic regions are being presented).

```
<a href=”gismap.cgi”><img src=”gismap.gif” ismap alt=”GIS map,
   select a state from the links which follow.”></a><br>
<a href=”gismap.cgi?state=AK”>Alaska</a><br>
<a href=”gismap.cgi?state=AL”>Alabama</a><br>
<a href=”gismap.cgi?state=AR”>Arkansas</a><br>
<!-- ... -->
```

Style Sheets

Section 508 Standards related to standards in this section

Section 508, 1194.22(d)
"Documents shall be organized so they are readable without requiring an associated style sheet."

Suggestion 27: Style Sheets

Related Section 508 Standards: 1194.22(d)

Documents using cascading and/or Javascript-enabled style sheets should be readable when the style sheet support is not available from the user's browser.

Technical Explanation

Style sheets are used to format Web content by defining the appearance of different elements, whether those elements are standard (e.g., `<h1>`) or developer-defined (e.g., `<p class=”boldness”>`). When styles are used to enhance the appearance of a well-structured document, the document should still be readable without style sheet support. When styles are used to convey information in a strictly visual fashion, such as grouping information by font or color, the document is not readable without style sheet support.

Javascript-enabled style sheets are sometimes used to hide and display content based on certain programmatic events. When style sheets of this nature are used, care should be given to ensure that the sections of the document are in a logical order when read from start to finish and, more generally, that the document is readable without style sheet support.

Forms

Section 508 Standards related to standards in this section

Section 508, 1194.22(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."

Section 508, 1194.22(p)

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

Suggestion 28: Forms—Label Placement

Related Section 508 Standards: 1194.22(n)

Form control labels should be placed immediately above or to the left of the associated control.

Example

```
<!-- immediately above -->
First name
<br>
<input type=”text” name=”name”>
<!-- immediately to the left -->
First name: <input type=”text” name=”name”>
<!-- or in the same table row -->
<tr>
<td> First name:</td>
<td><input type=”text” name=”name”></td>
</tr>
```

Suggestion 29: Forms—Associating Labels and Controls

Related Section 508 Standards: 1194.22(n)

Form control labels should be explicitly associated with form controls by placing HTML code for the controls inside the appropriate `<label>` elements.

Example

```
<label for=”firstname”>First name: </label>
<input type=”text” id=”firstname” tabindex=”1”>
```

Suggestion 30: Forms—Timed Responses

Related Section 508 Standards: 1194.22(p)

In the program code associated with an electronic form, users should be provided the opportunity to continue working with the form after any system or programmed “time-out” occurs.

Technical Explanation

The user of a screen reader, Braille display, or other alternative technology may require more time to read the form than the time allotted for a response, especially if the form is complex or contains large sections of text. Similarly, text input on a Web form using voice input, sip-and-puff devices, or other alternative input technology may take considerably more time than that allotted for a response.

Scripts

Section 508 Standards related to standards in this section

Section 508, 1194.22(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."

Section 508, 1194.22(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."

Section 508, 1194.21(a)

When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.

Suggestion 31: Scripts—Text Equivalents

Related Section 508 Standards: 1194.22(l)

Each script should be accompanied by a `<noscript>` element that contains a text equivalent for the script or a link to a page containing the text equivalent.

Technical discussion

If a script changes the content of a page without somehow indicating (through text readable by a screen reader) that it has changed the content of the page, the script is not accessible. Such scripts are inaccessible because the script is used to change an element that does not have functional text associated with it. A common example is the rollover technique, sometimes used cosmetically (e.g., to highlight menu options) but sometimes used functionally (e.g., to give more information via the status bar, a changing form field, or a graphic changed elsewhere on the page). If the script provides meaningful content or interface elements, those elements should be made available via functional text equivalents.

Note that the `<noscript>` element is not a complete solution, as many browsers used with screen readers are Javascript-enabled. In these cases, the browser will not deliver the `<noscript>` content, and the screen reader may not handle the scripted content appropriately. Providing a `<noscript>` element is only part of the solution where scripted content is concerned.

Example (for a script that provides information)

```
<!-- for a ticker-tape script, just supply the text -->
<script language=”Javascript”>
<!--
// here’s the script...
-->
</script>
<noscript>
Breaking news... Thermostats in state buildings to be set two
   degrees higher this summer... Dress code to be casual for all
   state agencies.
</noscript>
```

Example (for a script that provides functionality)

```
<!-- for a script that provides functionality, write a server-
   side alternative (e.g., CGI, ASP) and point the user to it -->
<script language=”Javascript”>
<!--
// here’s the script...
-->
</script>
<noscript>
<a href=”jsapp-equiv.cgi”>Equivalent version without
   Javascript</a>
</noscript>
```

Suggestion 32: Scripts—Keyboard Accessibility

Related Section 508 Standards: 1194.22(n), 1194.21(a)

User-activated scripts should be usable and navigable without a mouse.

Technical Explanation

Any navigation, interaction, and data entry enabled on a Web site must be available using only the keyboard. This includes, but is not limited to, scripted forms and scripted events. As regards scripts and other objects that are not directly keyboard-accessible, the Access Board notes in the preamble to Section 508 that "... some other method of working with the Web page must be provided." Scripts can almost always be made keyboard-accessible.

A script may create a barrier to accessibility if the event handler triggering the script cannot be activated through the keyboard. For example, rollover navigation can often only be activated through mouse movement. People with disabilities who cannot use a computer mouse—such as those who are blind or those who are quadriplegic—cannot trigger the rollover event. Another example is the use of an onChange event to allow one-click selections from a drop-down list (a `<select>` form field). Using only the keyboard, a user can reach this form field, but can only access the second item in the list. As soon as the user presses the down arrow to review the choices, the field is changed, triggering the onChange event.

When scripting events, avoid relying solely on mouse-driven event handlers. For event handler details, refer to the following table:

Instead of...	*Use...*
OnMouseOver	onFocus
OnMouseOut	onBlur
onMouseClick	onSelect
onMouseDown	onMouseDown *and* onKeyDown (redundantly)
onMouseUp	onMouseUp *and* onKeyUp (redundantly)
onClick	onKeyPress
onDblClick	(no keyboard input equivalent; avoid using this event)
onChange	onBlur (or simply avoid using the event)

Applets and Plug-ins

Section 508 Standards related to standards in this section

Section 508, 1194.22(a)

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

Section 508, 1194.22(m)

"When a Web page requires that an applet, plug-in or other application must 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 1194.21, paragraphs (a) through (l)."

Section 508, 1194.21(a) through (l) Software applications and operating systems

(a) When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.
(b) Applications shall not disrupt or disable activated features of other products that are identified as accessibility features, where those features are developed and documented according to industry standards. Applications also shall not disrupt or disable activated features of any operating system that are identified as accessibility features where the application programming interface for those accessibility features has been documented by the manufacturer of the operating system and is available to the product developer.
(c) A well-defined on-screen indication of the current focus shall be provided that moves among interactive interface elements as the input focus changes. The focus shall be programmatically exposed so that assistive technology can track focus and focus changes.
(d) Sufficient information about a user interface element including the identity, operation and state of the element shall be available to assistive technology. When an image represents a program element, the information conveyed by the image must also be available in text.
(e) When bitmap images are used to identify controls, status indicators, or other programmatic elements, the meaning assigned to those images shall be consistent throughout an application's performance.
(f) Textual information shall be provided through operating system functions for displaying text. The minimum information that shall be made available is text content, text input caret location, and text attributes.
(g) Applications shall not override user selected contrast and color selections and other individual display attributes.
(h) When animation is displayed, the information shall be displayable in at least one non-animated presentation mode at the option of the user.
(i) Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.
(j) When a product permits a user to adjust color and contrast settings, a variety of color selections capable of producing a range of contrast levels shall be provided.
(k) Software shall not use flashing or blinking text, objects, or other elements having a flash or blink frequency greater than 2 Hz and lower than 55 Hz.
(l) When electronic forms are used, 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.

Suggestion 33: Applets and Plug-Ins—Links

Related Section 508 Standards: 1194.22(m), 1194.21(a) through (l)

When a web page requires an applet, ActiveX object, or plug-in to interpret or generate page content, a link to the software should be placed on the page. If a separate applet, ActiveX object, or plug-in permits access to the page via assistive technology, a link to that software should also appear on the page.

Suggestion 34: Applets and Plug-Ins—Text Equivalent Information or Functionality

When an applet, ActiveX object, or plug-in is required to access page content and/or functionality, equivalent information and/or functionality should be provided in a keyboard-navigable text format, included within the `<object>` element.

If an applet is embedded using the `<applet>` element, a description of the applet's purpose should be included in the `alt` attribute of the `<applet>` element. Information and/or functionality equivalent to that provided by the applet should also be provided in a keyboard-navigable text format, included within the `<applet>` element.

Related Section 508 Standards: 1194.22(a), 1194.22(m)

Technical Explanation

Note that the `<applet>` element is deprecated and should not be used. Use the `<object>` element to embed applets. Alternative content is embedded between the `<object>` and `</object>` tags, in the same way that alternative content is embedded between the `<applet>` and `</applet>` tags.

The means of providing "equivalent information and/or functionality" will depend on the function of the applet. In cases where the applet provides information, simply duplicating that information will typically suffice. In the case of more complex applets, such as calculators, data access clients, and other interactive tools, typically it will be necessary to duplicate the applet's functionality on the Web server (e.g., using CGI) and provide an HTML form for the user to run the server-side application. The following three examples describe hypothetical but typical uses of applets, reasonable alternatives to those applets, and HTML techniques to supply those alternatives.

Example 1

The applet is a fancy way of providing information; (e.g., the agency logo and mission statement build in a timed sequence or fly around the screen or something). A reasonable alternative would provide the same information.

```
<object codetype=”application/java” classid=”kewlintro.class”>
<!-- Alternative begins here -->
<h1>The Really Kewl Agency</h1>
The mission of the RKA is to do really kewl things, like this
   tasteless but nifty little Java applet that flies our logo
   around the screen. We bet you have some questions now, like:
<ul>
<li>Aren’t we tragically hip?</li>
<li>Aren’t you jealous?</li>
<li>Don’t you wish you worked here?</li>
<li>Is this how you want to see your tax dollars at work?</li>
</ul>
<!-- Alternative ends here -->
</object>
```

Example 2

The applet is a fancy way of providing navigation; e.g., mouseovers on the highest-level topics drop down a list of next-level topics to click on. A reasonable alternative would provide the same navigation in a structured HTML format.

```
<object codetype=”application/java” classid=”apPopupMenu.class”>
<param name=”menuItems” value=”
 {_,http://www.apycom.com/,_self,pix/home.gif}
 {|Home,http://www.apycom.com/,_self}
 {|-}
 {|Download,http://www.apycom.com/download.html,_self}
 {|Contact,http://www.apycom.com/contact.html,_self}
 {-}
 {|Features & Setup,apmenu/index.html,_self}
 {|Parameters,apmenu/params.html,_self}
 {|Examples}
 {||Example #1,apmenu/ex1.html,_self}
 {||Example #2,apmenu/ex2.html,_self}
   ”>
<!-- Alternative begins here -->
<p>
<a href=”http://www.apycom.com/”>Home</a><br>
<a href=”http://www.apycom.com/download.html”>Download</a><br>
<a href=”http://www.apycom.com/contact.html”>Contact</a>
</p>
<p>
<a href=”apmenu/index.html”>Features and Setup</a><br>
<a href=”apmenu/params.html”>Parameters</a><br>
<ol>
Examples
<li><a href=”apmenu/ex1.html”>Example 1</a></li>
<li><a href=”apmenu/ex2.html”>Example 2</a></li>
</ol>
</p>
<!-- Alternative ends here -->
</object>
```

Example 3

The applet provides a calculator for tuition and estimated total cost of attendance given certain user input. A reasonable alternative would provide the same application on the server using CGI, ASP, or whatever tools the developer chooses, and give the user either a link to the server-side application or the initial form of the application. If a server-side application is impossible, offer information on how the user can obtain similar functionality.

```
<!-- Best technique -->
<object codetype=”application/java” classid=”totalcost.class”>
<!-- Alternative begins here -->
<form action=”totalcost.cgi” method=”post”>
<!-- form inputs, etc. -->
</form>
<!-- Alternative ends here -->
</object>
<!-- Acceptable technique -->
<object codetype=”application/java” classid=”totalcost.class”>
<!-- Alternative begins here -->
This application uses Java, which is unsupported by your browser.
   You may use this <a href=”totalcost.html”>equivalent form to
   estimate your total cost of attendance</a> instead.
<!-- Alternative ends here -->
</object>
```

Frames

Section 508 Standards related to standards in this section

Section 508, 1194.22(a)
"A text equivalent for every non-text element shall be provided (e.g., via "alt," "longdesc," or in element content)."

Section 508, 1194.22(i)
"Frames shall be titled with text that facilitates frame identification and navigation."

Suggestion 35: Frames—Labeling

Related Section 508 Standards: 1194.22(i)

For each `<frame>` element on a frameset page, the `name` and `title` attributes should be used to describe the frame.

Example: See Standard 36 examples.

Suggestion 36: Frames—<noframes> Elements

Related Section 508 Standards: 1194.22(a)

For each frameset page, a `<noframes>` element should be provided that contains a description of each frame page and redundant links to those pages.

Examples

Technique using server-side includes:

```
<frameset cols=”100,*”>
<frame name=”menu” src=”menu.html” alt=”Navigation Menu”>
<frame name=”content” src=”content.html” alt=”Page Content”>
<noframes>
<body>
<h1>Navigation Menu:</h1>
<!-- #include virtual=”menu.html”-->
<h1>Page Content:</h1>
<!-- #include virtual=”content.html”-->
</body>
</noframes>
</frameset>
```

Technique using redundant links:

```
<frameset cols=”100,*”>
<frame name=”menu” src=”menu.html” alt=”Navigation Menu”>
<frame name=”content” src=”content.html” alt=”Page Content”>
<noframes>
<body>
<a href=”menu.html”>Navigation Menu</a>: Provides links to other
   pages on this Web site.
<a href=”content.html”>Page Content</a>: Provides the information
   on this Web page.
</body>
</noframes>
</frameset>
```

In both cases, both menu.html and content.html have `<title>` elements:

```
<!-- in menu.html -->
<head>
<title>Navigation Menu</title>
</head>
<!-- in content.html -->
<head>
<title>Page Content</title>
</head>
```

Not Recommended

Many WYSIWYG design tools automatically supply a `<noframes>` element with content similar to that in the example below. This does not offer equivalent access to the user who cannot access frames. One of the techniques demonstrated above should be used instead.

```
<frameset cols=”100,*”>
<frame name=”menu” src=”menu.html” alt=”Navigation Menu”>
<frame name=”content” src=”content.html” alt=”Page Content”>
<noframes>
This page uses frames, but your browser doesn’t support them.
</noframes>
</frameset>
```

Keyboard Control

Section 508 Standards related to standards in this section

Section 508, 1194.22(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 §1194.21(a) through (l)."

Section 508, 1194.22(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."

Section 508, 1194.21(a)

When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.

Section 508, 1194.21(c)

A well-defined on-screen indication of the current focus shall be provided that moves among interactive interface elements as the input focus changes. The focus shall be programmatically exposed so that assistive technology can track focus and focus changes.

Suggestion 37: Keyboard Control

Related Section 508 Standards: 1194.22(m), 1194.22(n), 1194.21(a), 1194.21(c)

All functions that are possible with a mouse on a Web page must also be accessible to a user through the use of a keyboard, i.e., any Web page navigation, for instance mouseovers, must also be navigable from the keyboard, (usually through the "tab" key).

Technical Explanation

Any navigation, interaction, and data entry enabled on a Web site should be available using only the keyboard. As regards scripts, applets, ActiveX objects, plug-ins, and other objects that are not directly keyboard-accessible, the Access Board notes in the preamble to Section 508 that "... some other method of working with the Web page should be provided." This standard does not apply to networked versions of desktop applications; e.g., PageMaker.

Standard 37 is essential for Web accessibility because blind individuals should navigate the screen using the keyboard and many adaptive devices for a wide range of disabilities emulate the keyboard to provide access to the computer.

Text-Only Pages

Section 508 Standards related to standards in this section

Section 508, 1194.22(k)
"A text-only page, with equivalent information or functionality, shall be provided to make a Web site comply with the provisions of this part, when compliance cannot be accomplished any other way. The content of the text-only page shall be updated whenever the page changes."

Suggestion 38: Text-Only Versions

Related Section 508 Standards: 1194.22(k)

A text-only version of a site may be used only when there is no other way to present text-equivalent information in an accessible format and should be updated whenever the original Web site is changed.

Technical Explanation

Text-only versions of Web sites tend to fall out of sync with the "real" version over time, especially when maintained manually. If a text-only version is absolutely necessary, then it is recommended to automate the text-only presentation so that content for the text-only site is pulled from the same source as that for the "real" site. Such solutions can range from content management systems based on XML or databases to careful use of server-side includes.

However, even when a text-only version is in perfect sync with its non-accessible counterpart, the existence of a "separate but equal" site has a negative connotation for many users, who will always wonder what they're missing on the "real" site. Care must be taken to provide an equally rich experience in the text-only version, and the "text-only version" approach should be viewed as an absolute last resort.

Biometric Identification

Section 508 Standards related to standards in this section

Section 508, 1194.21(e)
"When biometric forms of user identification are used, an alternative form of identification, which does not require the user to posses particular biological characteristics, shall be provided."

Suggestion 39: Alternative Identification

Related Section 508 Standards: 1194.22(m), 1194.21(e)

A non-biometric alternative for user identification should be provided when a Web service or function relies on biometric identification.

Related Section 508 Standards: 1194.22(m), 1194.21(e)

Technical Explanation

This standard does not require a specific alternative. That selection is left up to the agency, which may choose a less expensive form of identification.

Equivalent Facilitation

Section 508 Standards related to suggestions in this section

Section 508, 1194.5
"Nothing in this part is intended to prevent the use of designs or technologies as alternatives to those prescribed in this part provided they result in substantially equivalent or greater access to and use of a product for people with disabilities."

Suggestion 40: Equivalent Access

Related Section 508 Standards: 1194.5

Nothing in this part is intended to prevent the use of designs or technologies as alternatives to those prescribed in this part provided they result in equivalent or greater access to and use of information and services when accessed by people with disabilities.

Technical Explanation

This document is not intended to resolve all issues of website accessibility. Because of the rapid changes in information technology, many of the standards may not promote accessibility in the future. Agencies are encouraged to deviate from these suggestions whenever the result will promote equivalent or greater access to and use of web sites by persons with disabilities.