

Technical Assistance Document for Oklahoma's Web-based Intranet and Internet Information and Applications

Draft in Progress – December 1, 2004

1. Introduction

- Overview
- Reason Purpose

2. Standards

- Definitions
- Sections - A B C D E F G H I J K L M N O P Q R S

3. Compliance Checklists

4. Best Practice

5. Testing and Validity

6. Resources

7. Contact Information

8. Upcoming Sections

1. Introduction

Overview

Oklahoma is one of many states that acknowledges the federal accessibility standards outlined in Section 508 of the Rehabilitation Act, as amended by the Workforce Investment Act of 1998.

In keeping with the spirit of that law, the Oklahoma legislature passed HB 2197 in April 2004, which requires state agencies to make electronic information technology accessible. In coordination of Section 2 of HB 2197, Oklahoma has developed state standards on accessibility for information technology, based on existing Section 508 standards, for the following areas: 1) Software Applications and Operating Systems, 2) Web-based Intranet and Internet Information and Applications, 3) Telecommunications Products, 4) Video and Multimedia Products, and 5) Desktop and Portable Computers.

Oklahoma's Technical Assistance Document (TAD) for Web-based Intranet and Internet Information and Applications offers additional assistance to those who need more detailed directions or advice for making their electronic and information

technology compliant according to Oklahoma's standards. It also offers a compliance checklist and examples and instructions on how to achieve compliance with each standard. Supplemental sections provide resources and testing as well as a listing of organizations and additional information sources.

Oklahoma's standards for Web-based Intranet and Internet Information and Applications closely mirror Section 508 standards. The standards use slightly modified language in six standards (a, c, j, k, l, and m) and add three additional standards (q, r, and s) to the Section 508 standards. These modifications, based largely on World Wide Web Consortium (W3C) accessibility standards, improve the clarity of some standards and add additional important guidelines for improved accessibility.

Reason/Purpose

Oklahoma's Technical Assistance Document is provided as a resource and guide for agency personnel as well as vendors for the development of Web sites and Web applications in compliance with Oklahoma's Web-based Intranet and Internet Information and Applications standards. This document can also serve as supplemental training for state agency, higher education, and career tech personnel. This document can also assist vendors to develop products or services that comply with Oklahoma's standards or determine if existing products or services already comply.

2. Standards

Definitions

The following definitions apply to these standards.

- Flicker. A repeated, rapid, or fluctuating variation of brightness, contrast, or position on a display.
- Key pages. Pages that represent the upper portions of a Web site's hierarchy with respect to navigation including home pages of major subdivisions of content or services.
- Meaningful text equivalent. Text that accurately and thoroughly conveys the content of a non-text element.
- Modification. Alterations or deletions in a web page, document or component, except where the changes are the result of:
 - + Automated retrieval of information from a database;
 - + Content retrieved, framed or otherwise imported from an external site or web-based service;
 - + Replacement of digital publications received from outside sources.

A) A meaningful text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content) except for captioning of audio information which shall comply with (b) of this section.

What: When non-text elements including, but not limited to, images, graphs, video, and animations are created to convey information to the user, alternative methods to deliver the same information shall be provided.

Why: Individuals with vision impairments may be unable to read or perceive information presented in visual elements such as images, graphs, and videos. For images, screen reading software reads the alternate text instead. For more complex non-text elements, such as charts, graphs, and videos, a more thorough description is required to describe the content contained within the elements.

How: All images must have appropriate and meaningful alternate text. As a rule of thumb, consider what you might say if you were reading the web page to someone over the telephone. You do not need to include the words "photo of" or "graphic of."

- * For images that contain words or letters – use alternate text that includes the same words or letters.

- * For image links – use alternate text that identifies the link's destination or function. You do not need to include the words "link to."

- * For graphics and photographs that provide information, provide alternate text that describes the information that the graphic or photo is intended to convey. Do not simply use "graphic" or "photo" as alternate text as they are not meaningful and do not describe the content contained within the graphic or photo.

- * For images that are invisible, purely decorative, or otherwise do not convey meaning – use alt="" (two double quotes with no space between) to indicate that the image can safely be ignored by a screen reader.

- * For graphs, charts, or other images that require a more thorough written description for comprehension, the longdesc (long description) attribute of the element can be used to provide a link to a full description. Also, a dlink (d-link) can be used to link to extended descriptions for images that are complex. Since longdesc is not supported by all current web browsers, it should not be used as the only method of providing a full description.

Ref: WCAG 1.1; 504 a

Difference between this standard and 508: The Oklahoma standard adds "meaningful" in front of "text equivalent" and specifies that audio information (considered a non-text element) is defined later in the document.

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

What: Multimedia generally refers to recorded or live media containing video, audio, or both video and audio tracks. Equivalent alternatives that are synchronized with the multimedia presentation allow a user with disabilities to interpret the information conveyed by the presentation.

Why: Individuals who are deaf or hard of hearing may require captions or an alternate method to access the audio information of multimedia. Blind or low-vision individuals may require audio descriptions to access the visual information in multimedia presentations.

How: Whenever possible, captions should be implemented using Synchronized Multimedia Integration Language (SMIL) to synchronize the display of text from a transcript with the video. A less desirable alternative is to add captions through standard video recording before converting to a web format. Video captions should be synchronized with the action taking place on the screen.

Audio descriptions should be provided when they are necessary to present information displayed in a video format.

An alternative for audio presentations is a text transcription. Text transcriptions should be provided in HTML and linked from near the audio presentation.

Ref: WCAG 1.3, 1.4; 508 b

C) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup. Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen.

What: Color is often used to indicate special functions or status. For example, required form fields are frequently indicated with red labels. Foreground and background colors can be specified by the web page author.

Why: Users who are blind, color-blind, or have limited or vision may miss information presented with color. Users with limited or low vision or color-blindness may have difficulty reading text that is similar in color to its background.

How: Whenever color is used as an indicator, use a non-color-based indicator as well. For example form fields could be identified with asterisks as well as color. For text, use dark colors on light backgrounds, or vice versa. Avoid combinations of red and green as well as busy background images.

Ref: WCAG 2.1, 2.2; 508 c

Difference between this standard and 508: The second sentence has been added to the original Section 508 standard.

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

What: The positioning features of Cascading Style Sheets can be used to position elements visually almost anywhere on a web page.

Why: Screen readers read through the elements on a web page in the order in which they appear in the page code, regardless of how they are positioned using style sheets. It is essential that the reading order match the logical flow of the document so that a screen reader user would hear the document in the same order that a visual reader would read it.

How: Check the reading order by following the order in which elements appear in the page code. Reading order can usually be adjusted by rearranging the order in which elements are defined in the code. Methods to skip repetitive navigation, as defined in section (o) of this document, can also assist the user in reading the content on a page.

Ref: WCAG 6.1; 508 d

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

What: Server-side image maps use coordinates to specify active regions of the image. When a user activates an area of a server-side image map, the request is sent back to the server to be processed. Browsers cannot indicate to the user the URL that will be followed when a region of the server-side map is activated.

Why: Redundant text links are necessary to provide access to the page for any person not able to see or accurately click on the map.

How: Redundant text links should be placed in close proximity to the image utilizing a server-side image map. Whenever possible, use client-side image maps instead of server-side image maps. Server-side image maps should only be used when the map regions cannot be defined as client-side images maps with an available geometric shape.

Ref: WCAG 2.1; 508 e

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.

What: Properly coded client-side image maps are more accessible than server-side image maps because each defined area can be given alternate text.

Why: Client-side image maps can provide the links related to the map in a text format which can be read with the use of assistive technology.

How: When using client-side image maps, provide an alternate text description to each active region of the client-side image map. Use alternate text that identifies the text equivalent of the active region as well as the link's destination or function. The words "link to" do not need to be included in your alternate text.

Ref: WCAG 9.1; 508 f g)

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

What: Headers identify the content of each row and/or column of a data table.

Why: A screen reader can use the table headers to provide row and column information while a user explores the data cells within a table

How: Identify content of data table by using column and row headings.

* Use <th> (column header) elements to identify the content contained in table columns.

* Alternately, the scope="col" (for column headers) can be used with either the <th> or <td> elements to identify column headings.

* Use <td> (table data) elements scope="row" (for row headers) to identify cells that contained within that row.

Ref: WCAG 5.1; 508 g

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.

What: Markup can be used to associate data cells with the appropriate header cells.

Why: Appropriate markup associating data cells and header cells in complex data tables allow users of assistive technology to better comprehend the information contained within the table.

When complex data tables are read by screen readers, users of screen readers may experience problems associating data with the appropriate row and column headers. Screen readers typically read tabular data in the order in which it is coded within the HTML source.

How: When possible, simplify complex tables by rearranging them or dividing them into separate tables. When a complex table is required, use advanced table markup, such as headers, axis, scope, <col> and <colgroup>, to fully indicate relationships between data cells and headers.

See W3C's "Tables in HTML Documents"
(<http://www.w3.org/TR/html401/struct/tables.html#h-11.2.6.1>) for complete details on how to markup complex tables.

Ref: WCAG 5.2; 508 h

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

What: HTML frames are used to divide web pages into separate areas, each displaying a separate web page. Each frame is identified by a name attribute and each page contained within a frame is identified by its <title> element.

Why: To navigate pages with frames, users who are blind must be able to identify the different frames and understand the purpose of each frame. Most screen readers identify frames by speaking the name and/or page title of each frame.

How: Give each frame an understandable name that indicates the frame's function. For example, use name="Navigation" and name="Content" rather than name="nav" and name="right". Set the <title> element of each page contained within a frame to match the name attributes or to identify the current content of that frame.

Note: Traditionally, the "name" attribute is used for programming and should not contain spaces. The title attribute, which can contain spaces, can also be used to set a more descriptive name for each frame. However, this technique is not yet supported by all screen readers.

Ref: WCAG 12.1; 508 i

J) Pages and elements shall be designed so that screen flicker does not occur between frequencies 2 Hz and 55 Hz.

What: Animated graphics, Flash, Java and other techniques are often used to create a variety of animated effects.

Why: Flickering or blinking between 2 Hz and 55 Hz (cycles per second) can trigger epileptic seizures. In addition, elements that move may be more difficult to view by individuals who use screen magnifying techniques. Animation can also be distracting for individuals with cognitive or other visual impairments.

How: Do not cause elements to blink regularly between 2 and 55 Hz. Since there is not an easy method to measure the flicker rate of a flickering element, the best solution is to limit use of animation to only what is necessary in the web page.

Ref: WCAG 7.1, 7.2, 7.3; 508 j

Difference between this standard and 508: This standard is a minor change from Section 508 to improve clarity and does not change the meaning or intent of the original standard.

K) A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the 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. The non-accessible version must be as accessible as possible.

What: Separate accessible or "text-only" versions are often offered instead of providing a single accessible site.

Why: Text-only pages can be used to provide equivalent information if accessibility compliance cannot be met in any other way. The content of the text-only page must be updated when the primary page is updated to keep information equally available to individuals who may not be able to interpret and read the primary page.

How: A text-only presentation of the content and functionality of the original page can be created. A link to the text-only page should be prominently placed to allow users and assistive technology to locate and navigate to the text-only portion.

Note: Manually developing and maintaining a separate "text-only" version of an entire site is tremendously demanding of time and resources. Given advances in accessibility techniques and assistive technologies, "text-only" sites are not necessary in most cases. Follow these standards to develop a single site that is universally accessible and efficient to maintain.

Ref: WCAG 11.4; 508 k

Difference between this standard and 508: This section has been modified from the Section 508 standard to include the last sentence.

L) When pages utilize scripting or other programmatic elements to display content, the information provided by the script shall also be provided in an equivalent text format that can be processed and interpreted by assistive technology. When pages utilize scripting or other programmatic elements to create user interfaces, user interaction shall be input device independent.

What: Scripting or other programmatic elements are often used to dynamically show or hide the content that appears on a web page or to perform important functions, such as checking that entries in form fields are appropriate. Client-side scripts, such as JavaScript, are scripts that are run by the user's web browser. Client-side scripts must be supported by and compatible with the user's browser in order to work.

For user interfaces, scripting or programmatic elements can be used within a web browser to automate certain tasks and enable pages to change and respond to user input. Some events are triggered by either mouse or keyboard actions. For example, an image can change color when the mouse pointer hovers over it (the onmouseover event).

Why: Older assistive technologies and web browsers may not support client-side scripting at all. Even current assistive technologies may interact in unexpected ways with content that is displayed using scripts, such as by skipping text that is

dynamically displayed or reading text that is dynamically hidden. Users need to be able to access the same essential content and functionality whether scripts are fully, partially, or not supported. It is not safe to assume that users with disabilities will have scripting support turned off.

Users with physical impairments may be able to use the keyboard but not the mouse. Individuals who cannot see the mouse pointer on the screen may use the keyboard or voice commands for all interactions. Scripts that can only be triggered by the mouse are not usable by these individuals.

How: Whenever scripts are used, it is the responsibility of the page developer to ensure that all information and functionality is accessible. There are several resources available as software downloads or as online applications (see appendix) to assist developers in testing for accessibility, but no computer software or application can determine full accessibility compliance. If there is any doubt, err on the safe side by ensuring that the essential elements of the page do not rely on scripts.

Note: One approach to ensuring accessibility with scripts is to include a back-up method of providing the same information and functionality that does not require scripts. For example, if a client-side script is used to check an entry in a form field, a server-side script could make the same check. Similarly, if scripts are used for "drop-down" menus, the same menu choices could be provided in an appropriate location elsewhere on the current or subsequent page. Additionally, scripting features that are purely decorative and do not present any significant information or functionality do not need to be made accessible. (However, remember Guideline 8.1 - Avoid flickering, blinking, and unnecessary animation.)

Whenever using a mouse-only event (e.g., onmouseover, onmouseout) to trigger a significant script action, also use the corresponding keyboard event (e.g., onfocus, onblur). Also make sure that keyboard events do not unintentionally trigger script actions. For example, keyboard users should be able to arrow through the choices in a <select> list without triggering each choice (e.g., onchange).

Ref: WCAG 6.3; 508 l

Difference between this standard and 508: This standard has been modified from the original Section 508 standard. The term "programmatic elements" has been added in the first sentence. "User interfaces" has been moved to a second sentence and information dealing with input device independency for user interfaces has been added.

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 Oklahoma Software Applications and Operating Systems standards (a) through (l).

What: "Applets" and "plug-ins" refer to a variety of web technologies, such as Java and Flash, which can be used to create advanced, interactive content on web pages. Both require additional software to be downloaded, installed, and run before the content can be viewed or used. Applets and plug-ins also operate with their own user interfaces, which are separate and different from that of standard web pages.

Why: Because applets and plug-ins have their own interfaces, they must be accessible in and of themselves. If essential content or functionality is presented using an applet or plug-in that is not accessible, it will not be usable by individuals with disabilities. Users may not already have the plug-in or applet required to access the information contained, therefore a link to a location where the appropriate plug-in or applet can be downloaded must be provided.

How: Check with the manufacturer and/or developer of each applet or plug-in to determine if and how the technology is accessible. When an accessible applet or plug-in is available, provide users with a link to any special instructions or software that is necessary.

Wherever a link is provided to an inaccessible applet or object, also provide a link to content in an alternate, accessible format. Make sure that the information and functionality is completely equivalent and up-to-date. Be sure to consider whether the inaccessible version is actually necessary.

In cases where it is impossible to create an equivalent accessible version, such as with some geographical imaging and mapping systems, exceptions may be necessary.

Ref: WCAG 8.1; 508 m

Difference between this standard and 508: This standard has been slightly modified to refer to Oklahoma's software and operating system standards rather than Section 508's standards.

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.

What: HTML forms include input elements such as buttons (`<input type="button">`), text boxes (`<input type="text">`), list boxes (`<select>`), and many others. Each field is typically identified by a text label.

Screen reader and keyboard users move between form fields (and links) using the Tab key. The order in which form fields receive focus is called the tab order. By default, the tab order follows the order in which elements appear in a page's HTML code.

Why: Screen readers cannot always determine which label belongs to which field based on positioning alone. The <label> element makes this association clear.

When screen magnification software enlarges a web page it also reduces the field of view. If form field label is positioned far away from its field, it may be impossible for a screen magnifier-user to view both the field and the label at the same time.

When filling out a form, screen readers typically read only the field's label. Screen magnifiers will focus on the field and its label, and instructions may be out of the field of view.

Depending on the design and layout of a page, the tab order may not match the visual (or logical) order of fields on a form. Reading fields out of their intended order can be disorienting for a screen reader or keyboard user.

How: Use the <label for=""> tag to label every form field.

Note: The value of a label's "for" attribute is the corresponding field's ID, not its name.

Position labels immediately adjacent to fields, preferably in standard locations, such as on the left or above text boxes and list boxes and on the right of checkboxes and radio buttons.

Use the "fieldset" element to group form controls into semantic units and describe the group with the "legend" element. Use "optgroup" to organize long lists of menu options into smaller groups.

Special instructions should be given before the form field and within the field label if possible. If instructions are too long to appropriately fit within the label, they should be given in an instructions section in advance of the form.

Make sure that fields appear in the HTML code in the logical order and/or use tab index to set the appropriate order.

Note: Tab index is not supported by Internet Explorer or Netscape versions 4 and below.

Ref: WCAG 9.4, 10.2; 508 n

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

What: Repetitive navigation lists or menus are commonly repeated on every page of a web site.

Why: By providing a method that permits users to skip repetitive navigation links, users that utilize a screen reader technology, keyboard navigation, or other assistive technology may access the content area of a web page without having to read or tab through the navigation each time they load a page. This significantly increases the

efficiency and usability of web pages for users who depend on these assistive technologies for access.

How: Provide a link at the beginning of navigation lists that links to a target at the beginning of the main content area of the page. This link must be available to screen reader and keyboard users, and is recommended to be visible when the page is viewed normally.

Ref: 508 o

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

What: Some web pages, frequently those that require a user to log in with an ID and password, "reset" themselves after a certain period of inactivity. Typically, any form entries that have been partially completed are erased and the user must start over.

Why: Users with visual, physical, or cognitive disabilities may require more time than average to read and interact with a web page.

How: Provide a clear explanation of any time limits and offer the user a way to extend or remove the limits if necessary. Avoid using time limits unnecessarily.

Ref: WCAG 7.5; 508 p

Q) Use valid, industry recognized web programming standards including a document type definition or the equivalent.

What: The World Wide Web Consortium (W3C) sets and publishes standards for most web programming languages, including HTML 4.01, XHTML 1.0, CSS Level 1 & 2, DOM, and SMIL. Programming code is considered "valid" when it follows all the rules and conventions specified in the published standards.

Why: Screen readers and other assistive technologies can more accurately interpret and interact with web pages that are built using valid, standard code. W3C languages

How: Indicate the programming language you are using by starting your code with a document type declaration such as:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

Use the W3C HTML Validation Service (<http://validator.w3.org>) and W3C CSS Validation Service (<http://jigsaw.w3.org/css-validator>) to check your code. Refer to the World Wide Web Consortium site (<http://www.w3.org>) for full specifications and documentation.

Ref: WCAG 3.2, 11.1

Difference between this standard and 508: This standard has been added to Oklahoma's standards and is not part of Section 508.

R) Identify the primary natural language of the document.

What: HTML uses the lang attribute to specify language in a web page. It can be set for any HTML element.

Why: Some screen readers and Braille devices are able to pronounce words in their appropriate language if it is specified.

How: Use the lang attribute on the <HTML> element to identify the primary language of each document. For example, <HTML lang="en"> identifies that English is the primary natural language for an HTML document.

Ref: WCAG 4.3

Difference between this standard and 508: This standard has been added to Oklahoma's standards and is not part of Section 508.

S) A link to the agency's Web site accessibility policy (if existing) and contact information for compliance issues related to the accessibility of electronic and information technology shall be included on home pages and other key pages.

What: An agency's Web site accessibility policy as well as a contact person for accessibility issues should be identified and presented on the agency's Web site. Contact information should include e-mail, telephone, TTY, and mailing address.

Why: The agency's Web site accessibility policy provides valuable information to users of the Web site. Contact information for accessibility compliance issues allows individuals to report accessibility problems or request information in an alternate accessible format. These resources should be readily available on home pages and other key pages on the agency's Web site to allow users to find them easily.

How: Provide a link to the agency's Web site accessibility policy as well as contact information on home pages and other key pages on the agency's Web site. Key pages include the agency home page, unit index pages, and pages that are common entry pages for users arriving at the site from search engines. Contact information may be listed directly or linked to from the home page or key pages. For best results, provide the policy link and contact information in site templates so that they appear on every page in a consistent location.

Ref: n/a

Difference between this standard and 508: This standard has been added to Oklahoma's standards and is not part of Section 508.

3. Compliance Checklists

In progress, will be attached.

4. Best Practice

Will be added later – this section is for extended standards.

- A. Introduction
- B. Extended Descriptions and Explanation of Best Practices
- C. Examples in Usage/Techniques
- D. Best Practices Checklist

5. Testing and Validity

In progress, will be attached.

- A. Testing (Introduction)
- B. Computer Testing
- C. Human Testing
- D. Software (+ plug-ins)
- E. Templates

6. Resources

More to come – descriptions, reorganization, etc.

* Access Board and Standards

<http://www.access-board.gov/508.htm>

* Access IT technical assistance for education

<http://www.washington.edu/accessit>

* Assistive Technology Program

Oklahoma ABLE Tech (800) 257-1705

<http://okabletech.okstate.edu>

* Center for Applied Special Technology

<http://www.cast.org>

* Disability Law Resource Project Technical support for the Americans with Disabilities Act and Educational Entities

<http://www.dlrp.org>

* Equal Access to Software and Education

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

* Federal Section 508

<http://www.section508.gov>

* National Center for Accessible Media

<http://ncam.wgbh.org>

* Oklahoma Legislative Task Force on Electronic and Information Technology Accessibility

For more information contact Oklahoma ABLE Tech (800) 257-1705

* Trace Center

<http://www.trace.wisc.edu>

* Web Accessibility Initiative-W3C Guidelines for web design

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

* WebAIM

<http://www.webaim.org>

* Oklahoma.gov Online tutorial for accessible web design

<http://www.ok.gov/webaim/>

7. Contact Information

This document was originally authored by members of the Electronic and Information Technology Advisory Council, Subcommittee 2: Web-based Intranet and Internet Information and Applications; Video and Multimedia Products.

Lead:

Chip Diffendaffer, Southwestern Oklahoma State University

Subcommittee Chair:

Kevin Sesock, Assistive Technology Specialist, Student Disability Services

Oklahoma State University,

Phone: 405-744-2024

Coordinating Organization for the EITA Advisory Council:

Oklahoma ABLE Tech, 1514 W. Hall of Fame, Stillwater, OK, Phone: 1-800-257-1705
or 405-744-9748.

<http://okabletech.okstate.edu>

Members:

Julie Brantley Department of Rehabilitation Services

Elaine Boykin, Department of Rehabilitation Services

Robert Breisch, Department of Career Tech

Lisa Counts, OK.gov

Joshua Craig, Web Administrator for the Oklahoma State Regents for Higher
Education

Rod Davidson, Department of Human Services

Brenda Dawes, OK ABLE Tech

Chip Diffendaffer, Southwestern Oklahoma State University

Ed Eckenstein, Water Resources Board

Teri Ferguson, Oklahoma State University - OKC

Phillip Jackson, Oklahoma County MIS

Linda Jaco, OK ABLE Tech

David Jinks, Department of Career Tech

Michael O'Hasson, Department of Libraries

Jason Price, Department of Rehabilitation Services

Lee Roberts, Rose Rock Design

Kevin A. Sesock, Oklahoma State University

William Wright, Oklahoma Health Care Authority

Upcoming sections:

8. Appendix

9. Glossary

10. References and Documents

11. Index