

# Understanding Accessibility

*A Guide to Achieving Compliance  
on Web Sites and Intranets*

Robert B. Yonaitis



HiSoftware Publishing  
Concord, New Hampshire USA

HiSoftware Publishing,  
Concord, New Hampshire 03301 USA

© 2002 - 2005 by HiSoftware, Inc.  
6 Chenell Drive, Suite 280  
Concord, NH 03301 USA

Voice: (603) 229-3055  
Fax: (603) 223-9741

Web: <http://www.hisoftware.com>  
Email: [info@hisoftware.com](mailto:info@hisoftware.com)

All rights reserved. Published 2002  
Printed in the United States of America

Substantial discounts on bulk quantities of HiSoftware Publishing books are available to corporations, professional associations, educational institutions, and other organizations. For details and discount information, contact HiSoftware sales at +1(603) 229-3055 or send e-mail to [sales@hisoftware.com](mailto:sales@hisoftware.com).

**ISBN**  
1-930616-03-1

Copyright 2002 by Hiawatha Island Software Company, Inc ("HiSoftware"). All rights reserved. TRADEMARKS: AccMonitor™, AccRepair™, AccVerify™, AccessibilityWATCH™, HiSoftware®, are trademarks of Hiawatha Island Software Company, Inc. Microsoft®, Microsoft® FrontPage®, Microsoft® Office, Windows®, and Windows® NT® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Any and all other product and company names mentioned herein are the trademarks or service marks of their respective owners.

## About the Author

Robert B. Yonaitis is the founder and Chief Executive Officer (CEO) of HiSoftware in Concord, New Hampshire. HiSoftware is a leading global supplier of accessibility management and testing solutions, Knowledge Management software and Web site management solutions for government, corporate, educational, and consumer markets. Robert has more than 14 years of experience in software engineering. Prior to founding HiSoftware, he led engineering teams at many Fortune 500 companies and State and Local Government agencies. Robert has specialized in re-engineering teams to facilitate better performance, decreased time to market and increased quality.



## Contacting the Author

Robert B. Yonaitis  
HiSoftware, Inc.  
6 Chenell Drive, Suite 280  
Concord, NH 03301 USA  
[ryonaitis@aol.com](mailto:ryonaitis@aol.com)

## Previous titles from Robert B. Yonaitis

*WYSIWYG*  
*The Elements of <WebSite> Promotion*  
Understanding Internet Traffic

## **ACKNOWLEDGMENTS**

When HiSoftware Publishing releases a book it is because there is a tremendous need to clarify something related to a technology and the Internet. In writing these books, we use input we have received from countless customers as well as employees of HiSoftware. In this book, we have attempted to identify a strategy that will assist readers in dealing with emerging accessibility standards through the use of evolving technologies. In doing so, we have also tried to create a blue print for “best practices” in accessible Web site design. We would like to acknowledge and thank everyone who seeks to clarify, simplify and bring significant solutions to the accessibility problem.

### **Corporate Contributors**

Amy M. Overka - Content Editor  
Dana L. Simberkoff - Content Editor  
Thomas J. Sweet - Content Reviewer

### **Cover design**

The cover art and design layout are an original work of:

Tandem Designs  
<http://www.tandemdesigns.com/>  
Kenneth Piccini & Jeannie Piccini

# Table of Contents

## 1. GETTING ACQUAINTED WITH THE ACCESSIBILITY PROBLEM AND ITS CHALLENGES ..... 1

PEOPLE WITH DISABILITIES AND ACCESS TO WEB-BASED INFORMATION .....	2
DISABILITIES ADDRESSED BY THE ACCESSIBILITY STANDARDS	4
IMPROVING THE WEB FOR EVERYONE .....	6
INDUSTRY AND THE FUTURE OF ACCESSIBLE TECHNOLOGY ....	6
FEDERAL REGULATIONS FOR ACCESSIBLE WEB CONTENT .....	7
<i>Overview of Subpart B</i> .....	9
<i>Requesting a copy of Section 508</i> .....	9
<i>W3C guidelines for accessible Web content</i> .....	10
<i>When accessibility laws take effect</i> .....	10
WHERE TO START .....	11
<i>Know the accessibility laws.</i> .....	11
<i>Find trusted resources.</i> .....	11
<i>Stay informed.</i> .....	12
<i>Consider automated software.</i> .....	12
<i>Who can help</i> .....	13
CREATING ACCESSIBLE WEB CONTENT .....	14
<i>Differences between Section 508 and W3C</i> .....	14
<i>Why you should build accessible.</i> .....	14
<i>Be an advocate</i> .....	15
ONE SET OF CONTENT .....	17
<i>Why full text is not manageable.</i> .....	18

## 2. THE CHECKPOINTS AND THEIR MEANING ..... 21

UNDERSTANDING PARAGRAPH (A) / W3C P1 (1.1) .....	22
UNDERSTANDING PARAGRAPH (B) / W3C P1 (1.4) .....	24
UNDERSTANDING PARAGRAPH (C) / W3C P1 (2.1) .....	26
UNDERSTANDING PARAGRAPH (D) / W3C P1 (6.1) .....	28

UNDERSTANDING PARAGRAPH (E) & (F) / W3C P1 (1.2) & (9.1) .....	29
UNDERSTANDING PARAGRAPH (G) & (H) / W3C P1 (5.1) & (5.2) .....	31
UNDERSTANDING PARAGRAPH (I) / W3C P1 (12.1) .....	34
UNDERSTANDING PARAGRAPH (J) / W3C P1 (7.1).....	35
UNDERSTANDING PARAGRAPH (K) / W3C P1 (11.4) .....	36
UNDERSTANDING PARAGRAPH (L) / W3C P1 ( <i>PARTIAL</i> 6.3) .....	38
UNDERSTANDING PARAGRAPH (M) .....	40
UNDERSTANDING PARAGRAPH (N) .....	42
UNDERSTANDING PARAGRAPH (O) .....	43
UNDERSTANDING PARAGRAPH (P).....	44
<b>3. INTRODUCTION TO ACCESSIBILITY TESTING.....</b>	<b>45</b>
COMMON ACCESSIBILITY ERRORS .....	45
AN INTRODUCTION TO QUALITY ASSURANCE.....	47
<i>Beta Testing</i> .....	50
INTRODUCTION TO ACCESSIBILITY TESTING .....	51
<i>Page versus Site</i> .....	51
<i>Advanced Content Delivery</i> .....	52
USING TOOLS.....	53
<i>Verification tools</i> .....	53
<i>Remediation (Repair) tools</i> .....	56
<i>Monitoring tools</i> .....	57
<i>What automated tools cannot do</i> .....	57
<i>Evaluating Accessibility Testing and Repair Tools</i> ....	57
EXISTING CONTENT MANAGEMENT SYSTEMS.....	59
EXISTING TESTING ARCHITECTURE .....	59
<i>What is an API?</i> .....	60
<b>4. DEVELOPING A STRATEGY FOR ACCESSIBILITY COMPLIANCE .....</b>	<b>61</b>
LOOKING FORWARD.....	62
THE RIGHT LEADERSHIP.....	64
A PHASED APPROACH .....	65

<i>The Site Assessment</i> .....	66
<i>Setting Goals</i> .....	72
<i>Setting Design Guidelines</i> .....	72
IMPLEMENTING ORGANIZATIONAL GUIDELINES.....	74
<i>Training Developers</i> .....	74
<i>Retrofit current content and develop all new content according to guidelines</i> .....	75
ACCESSIBILITY TESTING.....	76
<i>Unit Testing</i> .....	77
<i>Conformance Testing</i> .....	77
<i>Compatibility Testing</i> .....	77
<i>Assistive Technology Testing</i> .....	78
ACCESSIBILITY MAINTENANCE.....	78
INTRODUCTION TO INTRANETS.....	79
<i>Internet Sites vs. Intranet Sites</i> .....	79
<i>Creating an accessible Intranet</i> .....	80
<i>Intranet Priorities</i> .....	80
TAKE SMALL STEPS FOR LARGE GAINS IN ACCESSIBILITY ....	81
<b>5. ACCESSIBILITY &amp; DYNAMIC CONTENT .....</b>	<b>83</b>
INTRODUCING DYNAMIC CONTENT .....	83
COMMON FORMS OF DYNAMIC CONTENT .....	84
<i>HTML Templates</i> .....	84
<i>Server Side Includes</i> .....	85
<i>Web pages generated from a database</i> .....	86
WEB SITE STRUCTURE .....	87
<i>Add Structure to your Web site</i> .....	88
WHO SHOULD BE TESTING DYNAMIC CONTENT .....	88
WEB BASED APPLICATIONS .....	89
BROWSER EMULATION.....	90
CASE STUDY I - COMMON ISSUES WITH DYNAMIC CONTENT AS SEEN IN A MAJOR E-COMMERCE SITE .....	91
<i>Entering Product Information</i> .....	92
<i>Selling the Product</i> .....	93

<i>Let's fix the problem</i> .....	95
<i>When should the e-commerce company start?</i> .....	96
CASE STUDY II - SUBMITTING YOUR WEB SITE TO A POPULAR SEARCH ENGINE .....	97
<b>6. THE UNSEEN MEANING</b> .....	<b>101</b>
METADATA FOR ACCESSIBILITY .....	101
<i>An introduction to metadata</i> .....	101
<i>Metadata for Accessibility</i> .....	102
<i>The structure of the Meta tag</i> .....	103
<i>Choosing Meta tags</i> .....	103
<i>Adding Meta tags to documents</i> .....	104
WRITING TEXT EQUIVALENTS .....	105
<i>Tips for writing text equivalents</i> .....	105
<i>Branding and Accessibility Guidelines</i> .....	106
<i>Prevent Meaningless Link Text</i> .....	106
ACCESSIBILITY AND IMAGES .....	108
<i>Common but FLAWED practices</i> .....	109
<i>Common Error to Avoid - Text as Image</i> .....	111
<i>Good Practices</i> .....	112
<b>7. TUTORIAL: CORRECTING ACCESSIBILITY ERRORS</b> .....	<b>115</b>
USING THE TUTORIAL .....	116
SECTION 1 - SECTION 508 (A) W3C GUIDELINES 1.1 .....	117
1. <i>Simple Image Requiring Alternative Text</i> .....	119
2. <i>Same image: also serves as a navigation link</i> ....	120
3. <i>An image requiring a long description</i> .....	121
4. <i>Object MS Office Spreadsheet</i> .....	122
5. <i>Applet</i> .....	123
6. <i>Input Elements (also 508 (n))</i> .....	124
7. <i>Inline Frame (IFRAME)</i> .....	125
9. <i>Images As Bullets Example</i> .....	127
SECTION 2 - SECTION 508 (B) W3C GUIDELINES 1.1 .....	129
<i>Audio Files</i> .....	129



SECTION 3 - SECTION 508 (C) W3C GUIDELINES 2.1 .....	131
1. <i>Colored Text as the only way to determine a required step</i> .....	132
2. <i>Required form fields</i> .....	134
SECTION 4 - SECTION 508 (E) W3C GUIDELINES 1.2 .....	136
1. <i>Provide Redundant Text Links for server Side Image Maps</i> .....	137
SECTION 5 - SECTION 508 (G&H) W3C GUIDELINES 5.1 &5.2 .....	138
1. <i>Row and Column Headers shall be identified for Data Tables</i> .....	140
2. <i>All data tables should have a caption</i> .....	142
3. <i>All column &amp; row header cells are required to contain the scope attribute</i> .....	144
4. <i>All DATA cells are required to contain the header attribute if they are HEADER cells</i> .....	146
5. <i>All DATA cells are required to contain the AXIS attribute</i> .....	148
6. <i>When row grouping elements are used, all rows are required to be grouped</i> .....	150
SECTION 6 - SECTION 508 (I) W3C GUIDELINES 1.1 &12.1 .....	152
1. <i>All Frame Elements are required to contain the title attribute</i> .....	154
2. <i>All Frameset Elements are required to contain the noframes element</i> .....	155
SECTION 7 - SECTION 508 (J) W3C GUIDELINES 7.1 .....	157
1. <i>Do not use the BLINK Element</i> .....	158
2. <i>Do not use the MARQUEE element</i> .....	159
SECTION 8 - SECTION 508 (L)&(M) W3C GUIDELINES 6.3	160
1. <i>If you use the SCRIPT element use the NOSCRIPT element</i> .....	162
2. <i>Do not use script in ANCHOR elements</i> .....	163
3. <i>Do not use script in AREA elements</i> .....	164

SECTION 9 - SECTION 508 (0) .....	165
1. <i>Skip Navigation Links</i> .....	166
<b>8. USE ACCESSIBLE SOLUTIONS .....</b>	<b>169</b>
SOFTWARE STANDARDS.....	169
SECTION 508 .....	171
<i>The Voluntary Product Accessibility Template</i> .....	171
<i>Software Output</i> .....	173
<i>Supporting Features</i> .....	174
<i>Remarks and Explanations</i> .....	174
INTRODUCTION TO EITACC DESKTOP SOFTWARE	
STANDARDS .....	175
INTRODUCTION TO MICROSOFT ACCESSIBILITY GUIDELINES	
.....	176
<b>9. ACCESSIBILITY RESOURCES FOR FURTHER</b>	
<b>READING.....</b>	<b>177</b>
SECTION 508 .....	177
RELATED WEB RESOURCES .....	178
W3C / WAI .....	179
SOFTWARE SOLUTIONS.....	179
ACCESSIBILITY SERVICES.....	180
<b>APPENDIX A.....</b>	<b>181</b>
SECTION 508 STANDARDS AND W3C® PRIORITY 1	
CHECKPOINTS .....	181
<b>APPENDIX B.....</b>	<b>187</b>
SUMMARY OF SECTION 508 STANDARDS FOR SOFTWARE	
APPLICATIONS AND OPERATING SYSTEMS.....	187
<b>GLOSSARY .....</b>	<b>189</b>
<b>INDEX.....</b>	<b>193</b>
<b>REGISTER YOUR BOOK.....</b>	<b>195</b>

## INTRODUCTION

*Understanding Accessibility: A Guide to Achieving Compliance on Web Sites and Intranets* is designed to be a complete guide to creating Web sites that achieve compliance with U.S. federal standards for accessible Web content, outlined in the Rehabilitation Act Amendments of 1998, Section 508, Subpart B, §1194.22. Additionally, this guide can assist you in ensuring that your Web-based documents comply with World Wide Web, or W3C®, accessibility standards. In this book you will find:

- Comprehensive information on the accessibility guidelines
- Advice and guidance on how to achieve compliance
- Easy-to-follow instructions that will help you build accessible Web content
- A comprehensive reference on HTML coding for creating accessible sites
- Case Studies for Dynamics and Static Web sites
- Comparisons between W3C and US Federal Standards
- Useful information to help you select accessibility audit software.
- Information on managing accessibility testing
- Guidance in acquiring software based on software accessibility requirements.

This book is for Web Authors, Project Managers and basically any individual or team that is faced with the challenge of creating an accessible Web site or retrofitting a Web site to make it accessible. If you are new to accessibility we suggest you start at the beginning and read the entire book. If you are tasked with repairing sites that

are not accessible, this book will serve as a reference and guide. If you review sites for accessibility this book will help you to select tools and understand accessibility requirements.

## **WHAT'S IN THIS BOOK?**

This book introduces the accessibility initiative and challenges. It provides a complete reference on accessibility remediation. Detail on each section follows:

### ***Chapter One: Getting Acquainted with the Accessibility problem and its challenges***

In this section you will learn about the principal organizations that develop accessibility standards and the governing US laws, and you will also learn the about the types of disabilities addressed by the requirements. Chapter one also directs you to different resources, which can provide help to your organization and provides you with the justification on why you should build your Web site in an accessible format.

### ***Chapter Two: The checkpoints and their meaning***

In this section you will learn what each accessibility checkpoint is, what the goal of the checkpoint is and information about how to successfully comply with the checkpoint.

### ***Chapter Three: Introduction to accessibility testing***

In this section you will be introduced to the task of accessibility testing of Web pages and entire Web sites. In addition you will be introduced to testing, repair, and monitoring tools. We will briefly discuss content management systems and existing testing architectures.

### ***Chapter Four: Developing a strategy for Accessibility compliance***

In this section you will learn how to define training requirements as part of an initial site accessibility assessment.

### ***Chapter Five: Accessibility & Dynamic Content***

In this section you will be introduced to the relationship between dynamic content and the task of making it accessible. This section will include two case studies to demonstrate accessibility issues with dynamic content.

### ***Chapter Six: The unseen meaning***

This chapter covers three distinct methods to deliver a more usable site for those who have limited technology or a disability that prevents them from viewing the visual component of your web site.

#### ***Metadata for accessibility***

In this section you will learn how to use metadata as a means to increase your Web site accessibility.

#### ***Writing text equivalents***

In this section you will learn the best and worst practices for creating alternative text for non-text objects on your Web page.

#### ***Accessibility and images***

In this section you will learn how to properly assign alternative text to images and why. Additionally, we will point out some common but flawed practices for assigning alternative text to images.

### ***Chapter Seven: Tutorial: Correcting Accessibility Errors***

This section is a comprehensive guide and tutorial. This chapter's comprehensive examples and complete set of demonstration files will be invaluable in helping you understand accessibility markup.

### ***Chapter Eight: Use accessible solutions***

In this section you will learn about the accessibility requirements for software under the US Accessibility rules. You will also be introduced to the Voluntary Product Accessibility Template, ("VPAT").

### ***Chapter Nine: Accessibility resources and further reading***

This section provides you with additional information and resources.

## **MORE ABOUT THIS BOOK AND ACCESSIBILITY**

This book was originally published by HiSoftware to complement a suite of accessibility software solutions including AccMonitor, AccRepair, and AccVerify. These HiSoftware solutions feature verification and repair technology that helps users identify and correct Web elements for compliance with accessibility standards. Whether your organization is required to comply with the federal standards for accessibility and technology or not, achieving accessibility benefits everyone.

As you embark on a mission to achieve accessibility, be sure to reference the legal document in which the 508 Standards have been published, *Electronic Information Technology Accessibility Standards; Final Rule*. This document should be read and interpreted—for it is the basis for ongoing review of accessibility standards. Also, consider implementing the recommendations of the W3C®. Although

the law may not mandate compliance with these recommendations, implementation of these standards will increase the global reach of your Web content as well as increase the usability of that content for everyone.





# 1. Getting acquainted with the Accessibility problem and its challenges

---

## *In this chapter*

- Introduction to what is meant by accessibility and access to information on the Internet for people with disabilities
  - US Federal Regulations for accessibility
  - How to begin the task of creating accessible Internet based content
  - Being an advocate for accessible content
- 

Since the Internet was founded, it has become an easy way to publish and locate information. According to the American Heritage Dictionary, something is accessible when it can be “easily obtained.”<sup>1</sup> When information on the Web is accessible, everyone can find and use it.

Accessibility today has two meanings for Web content:

---

<sup>1</sup> Pickett, Joseph P., et al. (Ed.). (2000). The American Heritage Dictionary of the English Language. (4th ed.). Boston: Houghton Mifflin Company. [Internet]. Bartleby. Available: <http://www.bartleby.com/61/>. (Accessed 2001, May 16).

Information is accessible when it meets U.S. federal regulations for Web content. As of June 25, 2001, federally maintained Web sites and networks are required to comply with these accessibility standards.

Information is also accessible when it achieves the highest level of usability. In addition to federal regulations, there are many more suggestions that help everyone to find and use Web-based information. For additional information, contact the W3C® or visit <http://www.w3.org/WAI>.

In this book, the accessibility problem is introduced and comprehensive information is provided to assist your organization in producing and maintaining accessible content.

## **People with disabilities and access to Web-based information**

According to the US Census Bureau, December 1997 U.S. Census brief, one in five Americans have some kind of legal disability. (Source: December 1997 US Census Brief, "Disabilities Affect One-Fifth of all Americans," available at <http://www.census.gov/prod/3/97/pubs/cenbr975.pdf>.) The people impacted by these disabilities may require assistive technologies to access Web-based information. Most people use a browser like Microsoft® Internet Explorer™ or Netscape® Navigator™ to view Web-based information. But since popular browsers do not accommodate the needs of all people, some people use *assistive* technologies in conjunction with their Web browser. Assistive technologies, working in conjunction with the browsers, allow disabled users to access the Web-based information.

For example, if you are blind, you might use an automated screen reader. The screen reader, a software program, not only reads aloud the body text on a page, but also describes Web elements such as images. However, in order for a screen reader to describe images and other Web elements to the user, the HTML or other markup language used to code the page must make this information available to the screen reader.

When a screen reader encounters an image in a Web document it cannot describe the image to the user, unless the author of the Web document has provided alternative text. The alternative text, sometimes created in HTML by using the “alt attribute”, provides a description of the image to the screen reader. If an image in a Web document is a red and white circle and the alternative text description is “bull’s eye”, the screen reader can tell the user that there is a bull’s eye in the Web document. When no alternative text is present for non-text information, the screen reader cannot provide helpful information to the user.

Some Web authors use animations and sounds to convey information. These Web elements and techniques pose similar access problems for disabled users. Someone with a visual impairment or hearing difficulty cannot get helpful information from an audio-based presentation without a text description. A text description for an animation can help a user with a visual impairment, in the same way that a text description for sound can help a user with hearing difficulty. Text descriptions for sound are similar to captioning for television.

In addition to accessibility standards for images and multimedia elements, there are standards that allow users

to view and easily use pages even when their browsers and assistive technologies do not support display features.

Prior to accessibility laws, many Web sites were not compatible with assistive technologies. This shortfall left a large population of users unable to access information on the Web. Accessibility laws ensure that everyone can find and use the same information.

## **Disabilities addressed by the Accessibility Standards**

If you were to ask someone in your organization what an accessible Web site is they would probably reply that an accessible Web site is one that is usable by a blind person. In fact, an accessible Web site does aid in making the information on the site accessible to a blind person while also enabling people with a variety of other disabilities to do so.

### Some common disabilities:

- Blind Users
- Color Blind Users
- Users with weak vision that cannot read small text
- Deaf users
- Hard of hearing users
- Users that cannot use a mouse
- Users with disabilities like arthritis or other motor-control issues
- Photosensitive epilepsy - Not as common but it is addressed by the standard.

If you look at the list of possible disabilities addressed by the accessibility standards you can gain a greater awareness as to the scope of the standards and their importance. The following is a list of disabilities, the common issues they present and/or solutions that can address access issues:

#### **BLIND**

Software that reads content out loud or solutions that translate the content to brail

#### **COLOR BLIND**

Site must be developed to not rely on color

#### **LOW VISION**

Users with low or weak vision will use browser software or magnifying software to adjust for disability

#### **DEAF OR HARD OF HEARING**

Rely on captioning or text transcripts of audio content

#### **MOTOR-CHALLENGED, UNABLE TO USE KEYBOARD AND/OR MOUSE**

People who do not use a mouse rely on keyboard shortcuts or pointing devices held in the mouth  
Additionally speech interfaces provide a potential solution

#### **PHOTOSENSITIVE EPILEPSY**

People with this disability could have a seizure triggered if the computer screen has movement,

flickering or animation, which is not in compliance with the rule.

## **Improving the Web for everyone**

Since the beginning of the “Information Age”, technology has been developing rapidly. Television shows from decades past, like *Star Trek*®, portrayed people walking around with hand-held communicators, and speaking to their computers to use them. Today this is a reality - cellular phones and beepers allow us to remain in constant communication, and handheld computing is commonplace. People are using small devices to download e-mail, get stock quotes, find driving directions, and communicate with friends and relatives. Scenarios like this are becoming more common, and technology providers no longer presume that most people use a desktop computer with a keyboard and a pointing device to surf the Web. Accessibility laws are making Web-based information user-friendly for all users.

Compliance with these standards, not only assists users of assistive technologies, but also can improve access to the Web for these hand-held, wireless devices. These laws are based on best practices for Web authoring and information technology. While many of the laws directly benefit users with disabilities, who might rely on assistive technologies to view information, the laws also benefit everyone else.

## **Industry and the future of accessible technology**

The future of Web accessibility is optimistic. Since federal accessibility laws have been adopted, technology

corporations have publicly declared their commitment to these standards, even though they are not required to do so by law. Moreover, technology executives are collaborating on their visionary ideas for accessibility that extend far beyond legal requirements. Industry leaders have realized that the true power of the Internet lies in its ability to deliver information to all people.

No one knows for certain how accessibility will be made possible in the future. Perhaps it will take the form of a dynamic browser, so that users will no longer need assistive technologies. Or maybe Web authoring software of the future will compensate for the current shortfalls of Web-based information. But until mainstream technology allows all users to access the same information, Web authors must make provisions in the source code of their Web documents so that Web information can be processed by assistive technologies. This book can help you do this.

## **Federal regulations for accessible Web content**

Federal regulations governing accessible Web content resulted from *The Rehabilitation Act Amendments of 1998*, which included *The Workforce Investment Act of 1998* and stemmed from *The Rehabilitation Act of 1973*.

Section 508 contains regulations for information technology. These regulations have been issued in publication S-40, *Electronic Information Technology Accessibility Standards; Final Rule*. This document states that “when Federal agencies develop, procure, maintain, or use electronic and information technology, they shall ensure that the electronic and information technology allows Federal employees with disabilities to have access

to, and use of information and data, by Federal employees who are not individuals with disabilities, unless an undue burden<sup>2</sup> would be imposed on the agency.”<sup>3</sup>

Section 508 addresses accessibility for electronic and information technology. Under Section 508, there are several subparts:

- Subpart A contains general information about the purpose, application, and implementation of 508 standards.
- Subpart B contains technical standards, including Web-based intranet and Internet information and applications. Other components of Subpart B include standards for software products, operating systems, telecommunications products, video and multimedia products, self-contained products, and desktop and portable computers.
- Subpart C contains functional performance criteria.

---

<sup>2</sup> According to the Architectural and Transportation Barriers Compliance Board, “This is consistent with language used in the Americans with Disabilities Act (ADA) and other civil rights legislation, where the term ‘undue burden’ has been defined as ‘significant difficulty or expense.’ However, the agency must explain why meeting the standards would pose an undue burden for a given procurement action, and must still provide people with disabilities access to the information or data that is affected.”

<sup>3</sup> Architectural and Transportation Barriers Compliance Board. (2000). *Electronic Information Technology Accessibility Standards; Final Rule*. Federal Register.



- Subpart D contains standards for information, documentation, and support.

### ***Overview of Subpart B***

Subpart B contains §1194.22, the section that addresses standards for Web-based intranet and Internet information and applications.

*Subpart B - Technical Standards* in its entirety contains the following subsections:

§1194.21 Software Applications and Operating Systems<sup>4</sup>

§1194.22 Web-based Intranet and Internet Information and Applications

§1194.23 Telecommunications Products

§1194.24 Video and Multimedia Products

§1194.25 Self Contained, Closed Products

§1194.26 Desktop and Portable Computers

### ***Requesting a copy of Section 508***

To get a copy of publication S-40 *Electronic Information Technology Accessibility Standards; Final Rule*, contact the Access Board's automated publications order line. Dial (202) 272-5435, press 2 on the telephone keypad, and then press 1. Request publication S-40, *Electronic Information Technology Accessibility Standards; Final Rule*.

---

<sup>4</sup> See Appendix B for a summary of the 508 standards pertaining to software applications and operating systems.

If you use a TTY, call (202) 272-5449. Record the name, address, and telephone number of the person requesting publication S-40, *Electronic Information Technology Accessibility Standards; Final Rule*.

Copies are available in alternative formats including cassette tape, Braille, large print, or computer disk.

The publication is also available on the Internet at:  
<http://www.access-board.gov/sec508/508standards.htm>

### ***W3C guidelines for accessible Web content***

The W3C®'s Web accessibility initiative group (WAI) pursues accessibility of the Web through five primary areas of work:

- Technology
- Guidelines
- Tools
- Education and outreach
- Research and development.

For more information on the W3C visit their Web site at:  
<http://www.w3.org/WAI/>

From here you can find accessibility news and all associated Web accessibility information, standards, and efforts available from the W3C's Accessibility group.

### ***When accessibility laws take effect***

Section 508 requires that federally maintained Web sites and intranets be in compliance with the standards after June 25, 2001. If you are uncertain as to whether or not

your organization is required by law to comply with accessibility standards, contact the Access Board or visit <http://www.access-board.gov>.

## Where to start

### ***Know the accessibility laws.***

Obtain a copy of publication S-40, *Electronic Information Technology Accessibility Standards; Final Rule*, and become familiar with it. This guide also helps to explain laws pertaining to Web content on the Internet and intranets.

Another document from the Access Board provides specific techniques on how you can comply with accessibility standards. Visit <http://www.access-board.gov/sec508/guide/1194.22.htm> for more information.

### ***Find trusted resources.***

There are many resources available to help you develop accessible Web content. The Access Board, the organization administering federal accessibility laws, offers technical support and additional information on accessibility.

Also consider W3C®, Trace Research & Development Center, National Center for Accessible Media, and National Arts and Disability Center. Contact these agencies directly, or visit their Web sites for more information.

### ***Stay informed.***

It is important to stay informed for several reasons:

- As technology develops, accessibility standards are certain to change.
- New techniques for creating accessible Web content are being published frequently.
- As developers enhance the capabilities of browsers and markup language for Web documents, achieving accessibility will become easier.

### ***Consider automated software.***

There are many software solutions on the market to help you verify and achieve compliance with accessibility laws. HiSoftware has a range of solutions to suit your specific needs. HiSoftware solutions comply with accessibility laws for software applications<sup>5</sup>. Visit <http://www.hisoftware.com>, or call +1-(603) 229-3055 for more information.

---

<sup>5</sup> See Appendix B for a summary of the 508 standards pertaining to software applications and operating systems.

### *Who can help*

The Access Board provides technical assistance with the implementation of Section 508 standards.

The Access Board provides online documentation to help you get started. Visit <http://www.access-board.gov/sec508/guide/1194.22.htm> for more information. Or contact the Access Board for support.

The Access Board  
Office of Technical and Information Services  
1331 F Street, NW, Suite 1000  
Washington, DC 20004-1111  
Voice: (800) 872-2253 or (800) 993-2822 (TTY),  
weekdays 10-5:30 EST (Wed. 10AM-2PM)  
Fax: (202) 272-5447  
Web: <http://www.access-board.gov>  
E-mail: [508@access-board.gov](mailto:508@access-board.gov)

## **Creating Accessible Web Content**

Accessibility laws have been issued in publication S-40, *Electronic Information Technology Accessibility Standards; Final Rule*.

The standards that appear in this book have been interpreted from their original form. To read the standards in original form, request a copy of publication S-40, *Electronic Information Technology Accessibility Standards; Final Rule*. Section 508, Subpart B, §1194.22 explains the accessibility standards for Web-based intranet and Internet information and applications.

### ***Differences between Section 508 and W3C***

The Section 508 Standards, which were introduced as a rule for federal Web site development, are an extension and/or modification of the W3C standards. In the appendix of this book there is a mapping of W3C standards to 508 Standards. The overall differences between these two standards are minimal.

In the near term, it is safe to say that if your intentions are to develop a site with the goal of it's being accessible then you can follow W3C Priority One standards or Section 508 Standards. Supporting both standards is also simple once you have made the decision to create an accessible Web site.

### ***Why you should build accessible***

Developers often ask, "Why should our site be accessible?" Aside from any legal or statutory

requirements, the answer to this question is simple - offering equal access to disabled users is compelling, and for many reasons:

- Build your Web site audience.
- Take a position of leadership in your particular business sector.
- Grow a loyal customer base.
- If you are doing business with the government, it is the law.
- It is apparent for non-government Web sites that it will become the law.
- The cost of retrofitting is much higher than building an accessible site from scratch.
- It simply makes good business sense to increase the number of your potential customers.
- It is the right thing to do at a minimal cost.

It is hard to imagine why one would not want to build an accessible Web site. The information and ideas expressed in this book will illustrate that it is not difficult to achieve and maintain an accessible Web site. The remainder of this chapter deals with how to make your content accessible. Full examples are provided as part of the descriptions.

### ***Be an advocate***

There are many good reasons for supporting the creation of accessible Web content. It is important to recognize that these changes will not happen over night. Simply put, an advocate is someone that pleads on another's behalf. The most successful advocates in history have been those who have sought to bring people to their position while

constructively demonstrating the deficits of current circumstances.

Advocates within organizations should start out by trying to educate as many people as possible about the accessibility issue. We have discussed many compelling reasons why an organization should develop Web sites to be accessible. It should be very easy to communicate them to people in your organization. Once it is clear that “developing accessible” is achievable, affordable, and can provide a competitive edge, it will become common rather than the exception.

The following types of activities are recommended for accessibility advocates:

- Educate your team members
- Educate individuals outside of your team
- Whenever you create documentation or any type of information that will be viewed by others, make it accessible, even if it is not a requirement or policy
- Participate in local or national organizations related to the accessibility issue

The following types of activities should be AVOIDED:

- Do not tell people that they will lose site visitors if their site is not accessible, instead explain about the increased user base they would have if their site was accessible.
- Do not attack a manager’s character if they turn you down for resources to perform an accessibility project, instead educate them by justifying the



project in a way that disproves their reasoning for initially rejecting your project.

Remember, if people fully understand the accessibility issue and the positive impact they can make on the issue by building accessible, then they are likely to do so.

## **One set of Content**

It is very important to educate your Web team on Web content accessibility. One frequent “misconception” about accessibility compliance, relates to the use of a second set of content that is developed as a text-only version of a site. Some developers have been told that a second set of content, provided as a text-only site, will be an acceptable replacement to making a site accessible.

This simply may not be true and defeats the spirit and the years of work and research behind making electronic information accessible to all in equal form and content. The full text page cannot portray the true meaning of content unless the content is, by default, full text for the following reasons:

- If an image had no meaning it would not have been used in the first place.
- If a navigation element had no meaning it would not have been used in the first place.

Ask yourself a simple question:

If I create a full text page that actually contains the same content as the main page, does it satisfy accessibility

requirements for all of the people with potential disabilities that may be viewing the site?

- Blind
- Color Blind
- Weak vision
- Deaf or hard of hearing
- Inability to use mouse
- Motor-Control Challenges

**Answer:** Even if you were able to address all of these issues in your text-only version of the Web site, from a management perspective, you will now be faced with synchronizing, updating, and maintaining two versions of the same site. When dealing with the accessibility problem it is more cost effective to make one set of content that is accessible by all!

### ***Why full text is not manageable***

From another perspective full text can be a management nightmare. The goal is to deliver content with exact meaning to all users all of the time. Full text pages are created in several ways:

#### **AUTOMATED**

This is probably the best method however there is a high probability that some data will not translate which means that the content will not be the same. The automated translation of a Web site into a text-only Web site, will meet the same limitations that assistive technologies face - it can only work with information that already has the text equivalents behind non-text elements.

## **MANUAL**

This method can be time consuming not only during creation, but also from an upkeep perspective. In order to make a simple change in a page means you must now remember to update the text page. There is a high probability that the pages will be out of sync after several updates.

## **CONTENT MANAGEMENT SYSTEMS**

Content Management systems will encounter the same challenges described with the automated translation example above. Because a simple translation is not likely, the best way to handle this is to create accessible templates and code from the beginning versus attempting to come up with full text alternative content with equivalent meaning.

Now if full text is the only way you can go then you are permitted to do this - just remember the challenges:

- High cost to implement effective system.
- Content Synchronization.
- Difficult to translate meaning of graphics via automation.



## 2. The Checkpoints and their meaning

---

### *In this chapter*

- Introduction to Section 508 and W3C guidelines
  - Detail on disabilities addressed by each checkpoint
  - Summary of how to validate each checkpoint
- 

The Section 508 and W3C Priority One standards are very similar to each other with a few subtle differences. For this chapter we will cover section 508 checkpoints and their meanings.

*See Appendix A for a complete summary of Section 1194.22 vs. W3C Priority One. This map will allow developers to use the following information regardless of the standards with which they are developing their site to comply.*

This chapter provides a conceptual overview of accessibility requirements, detailing both why they are important and providing a general overview of how to correct accessibility issues. For a complete reference on HTML Markup for Accessible design please see Chapter 7 for the Tutorial and HTML Examples.

## Understanding Paragraph (a) / W3C P1 (1.1)

*(a) A text equivalent for every non-text element shall be provided*

A text equivalent conveys the same information to the user as the original non-text element. The section 508 guideline points out the following items as examples of elements that require text equivalents.

- Buttons
- Check Boxes
- Pictures/Images
- Embedded or Streaming audio or video.

### Special W3C vs. 508

The mapped W3C guideline, 1.1, includes other items, not specified in Section 508 (a), however included under the federal guidelines.

- Input Elements
- Object Elements
- Applet Elements
- Frameset Elements
- IFrames
- Anchors
- Area Elements

These are covered later in 508 in other sections like (n) as well as being required but not implicitly mentioned in paragraph (a).

### **DISABILITIES ADDRESSED BY THIS PARAGRAPH**

The following disabilities are addressed by this paragraph:

- Blind
- Weak vision
- Deaf or hard of hearing

## **DEALING WITH NON-TEXT ELEMENTS**

Chapter 6 of this book goes into detail on how to write text equivalents. The most important points to remember are:

- The Alternative text should fully represent the image meaning and its purpose to the person viewing the web page.
- If a long description is required for the image the longdesc attribute can be utilized to post another URL, which will contain additional information.

## **TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (A)**

It is easy to test your page for compliance to paragraph (a). Automated testing tool such as HiSoftware's AccVerify can quickly test a page or an entire Web site.

## **Understanding Paragraph (b) / W3C P1 (1.4)**

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

### **DISABILITIES ADDRESSED BY THIS PARAGRAPH**

The following disabilities are addressed by this paragraph:

- Blind
- Weak vision
- Deaf or hard of hearing

### **DEALING WITH ALTERNATIVES FOR MULTIMEDIA CONTENT**

A video presentation with an audio component requires captioning. The captioning must be completely synchronized with the audio presentation to allow for the viewer to follow the meaning of the content.

If there is only an audio component there is no need for captioning, however, a text equivalent (Full Transcript) “must” be made available.

If there is a slide show like a PowerPoint slide show available but it is “visual only”, the graphics need to have alternative text representations.

### **EXAMPLE AND TIP FOR POWERPOINT PRESENTATIONS**

For those providing silent Microsoft PowerPoint presentations via the Internet remember to provide alternative text for your images. In PowerPoint 2000 and above you can do this by following the steps below:



1. Select on the Image that that needs alternative text
2. Select Format | Format Picture from the Main Menu
3. Select the “Web” Tab
4. Type in the Alternative Text
5. Click on OK

## **TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (B)**

This is a visual checkpoint. When content developers are delivering multimedia content they must make sure it is in compliance with paragraph (b)

## Understanding Paragraph (c) / W3C P1 (2.1)

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

### DISABILITIES ADDRESSED BY THIS PARAGRAPH

The following disabilities are addressed by this paragraph:

- Blind
- Weak vision
- Color Blind

### DEALING WITH COLOR

The use of color does not prevent a page from meeting the accessibility requirements. A junior designer may believe this is the case; it is important that designers understand that they can use color and be in compliance with accessibility standards. This paragraph (checkpoint) deals with the use of color alone in indicating action or meaning.

#### Notes on color

Do not display emphasis of a word by color alone.

For example do not create a form that requires users to fill out all form field values that are “marked in red.” Use HTML Markup like <Strong> or <EM>. Do not use color alone in an image to indicate an action.

Example: Do not say, “Press the red button to cancel your order or the green button to complete your order”. Color can definitely be used to augment the design of form but use alternative text and /or text on the image to signify cancel and finish.

## TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (C)

It is easy to test your page for compliance to paragraph (c). This check is clearly a visual check and can be completed by testing the output devices that provide a black and white output.

### Printer

Print your Web page on a black and white printer. View the output to assure that all text and images are still visible, and that the emphasis required is still visible.

### Screen

View your page on a black and white monitor or set your monitor to the aforementioned view and view the screen to assure that color was not the only way that information has been presented.

## **Understanding Paragraph (d) / W3C P1 (6.1)**

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

### **DISABILITIES ADDRESSED BY THIS PARAGRAPH**

The following disabilities are addressed by this paragraph:

- Blind
- Color Blind
- Weak vision

### **DEALING WITH STYLE SHEETS**

Style Sheets are intended to separate presentation from content. They provide an easy and efficient way to format HTML documents. Style Sheets enhance the effectiveness of HTML as well as simplify the process of maintaining a Web site. Additionally, the use of Style Sheets enables users to set viewing preferences to accommodate their disability.

With this in mind it is important that a Web site developer does not design a page that will interfere with user defined style sheets.

### **TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (D)**

It is easy to test your page for compliance to paragraph (d). Simply turn off support for style sheets, all information and its meaning should be present without style sheets. Additionally, you can test with a user defined style sheet to make sure that it does not interfere with the ability to understand and navigate the site.

## Understanding Paragraph (e) & (f) / W3C P1 (1.2) & (9.1)

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

### DISABILITIES ADDRESSED BY THIS PARAGRAPH

The following disabilities are addressed by this paragraph:

- Blind
- Color Blind
- Weak vision
- Inability to use mouse
- Motor-Control Challenges

### DEALING WITH IMAGE MAPS

An image map is a single image that is used to offer multiple links to content in a Web site. Image maps are rarely actual maps! When a user selects a link on an image map it takes them to the link connected with the referenced part of the image.

There are two types of Image Maps, client and server side. The main difference between the two is where the actual processing of a visitors action taken on the map is translated.

The rule for paragraph (e) is necessary because the actual link is processed on the server so it is not available to users of assistive technology because of their inability to see the

map. By presenting the user with a list of links, the map is made accessible.

The rule for paragraph (f) is necessary because it allows the user of assistive technology to easily activate regions of the image map. Related Rule: Paragraph (a)

## **TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (E) & (F)**

It is easy to test your page for compliance to paragraphs (e) & (f). AccVerify can quickly test an entire site or page.

## Understanding Paragraph (g) & (h) / W3C P1 (5.1) & (5.2)

*(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.*

### DISABILITIES ADDRESSED BY THIS PARAGRAPH

The following disabilities are addressed by this paragraph:

- Blind
- Weak vision

### DEALING WITH TABLES

Tables can be used for design/Layout or as a means to present data. Tables that are used to present data should be structured so that those users who cannot view the data with its visual reference points can easily navigate them. The structure of these tables implies meaning to the information within them, based on the organization of the table. Therefore, a screen reader should be able to utilize this table structure.

### MAKING A TABLE ACCESSIBLE

The following steps are designed to enable users of assistive technology to understand and navigate tables successfully.

1. Determine the Table Type (If Layout versus data table, than no further action is required. \*Note the

developer should make sure that the table used for layout makes sense when linearized.)

2. For Data Tables first enter a table summary:

The summary attribute allows the developer to provide a summary of the tables information.

Screen readers can take advantage of this **non-visual** (does not appear on the page when viewed in a browser) attribute.

The summary should provide the page viewer with:

- Overview of the table
- Overview of table contents
- Overview of the Structure
- Overview of the Tables purpose

*Optionally the developer can use the Caption tag.*

*The caption appears as part of the document text on a page with a table. The Caption tag is visible on the Web Page.*

3. There are cells in the tables that serve as headers, or labels for the information in the respective row or column. Headers must be identified as header cells. In this step of making the table accessible, you need to locate header cells in the table, and then change the TD tag to a TH tag.

4. Identify Row and Column Headers

**Simple tables** - consider identifying row and column headers with the scope attribute.



**Complex tables** - consider identifying row and column headers with the headers attribute. If the table has more than one row or more than one column of header cells, consider using the axis attribute to group them together.

5. Group multiple header rows or header columns.  
*When header cells are clearly identified the significance of the data cells is easily identified.*

## TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (G) & (H)

It is easy to test your page for compliance to paragraphs (g) & (h). AccVerify can quickly test your Web page or your entire Web site. Remember a tool, which can filter out layout tables, will offer a streamlined approach to testing tables for accessibility.

## **Understanding Paragraph (i) / W3C P1 (12.1)**

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

### **DISABILITIES ADDRESSED BY THIS PARAGRAPH**

The following disabilities are addressed by this paragraph:

- Blind
- Weak vision

### **DEALING WITH FRAMES**

A Framed Site uses combinations of HTML documents, displayed together in the browser to visually divide the computer screen into separate and distinct areas that can be separately redrawn depending on user action. Frames present difficulties for users with disabilities when those frames are not identifiable to assistive technology. Users of assistive technology will find it difficult to navigate frames if the differences between the frames in the frameset are not clearly defined.

To facilitate easy navigation with frames use the “title” attribute available to the frame element.

### **TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (I)**

It is easy to test your page for compliance to paragraph (I). AccVerify can quickly test an entire site or page.

## **Understanding Paragraph (j) / W3C P1 (7.1)**

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

### **DISABILITIES ADDRESSED BY THIS PARAGRAPH**

The following disabilities are addressed by this paragraph:

- Photosensitive epilepsy

### **DEALING WITH PARAGRAPH (J)**

The developer should prevent the screen from:

- Flashing
- Flickering
- Blinking
- Or Having Staggered Movement from a control like the marquee or a News Bot

If the screen behaves in one of the manners listed above, it could trigger a seizure in an individual with the aforementioned disability.

### **TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (J)**

Flashing or flickering elements should be avoided and should be identified by your testing tool. This is a visual as well as an automated test.

## **Understanding Paragraph (k) / W3C P1 (11.4)**

(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.

### **DISABILITIES ADDRESSED BY THIS PARAGRAPH**

The following disabilities are addressed by this paragraph:

- Potentially all disabilities referred to in the standards

### **DEALING WITH TEXT-ONLY PAGES**

Paragraph (k) provides a failsafe measure for pages or content that cannot be made accessible. It requires that there be an accessible alternative. This is commonly referred to as a “Full Text” version of a Web site or page. This alternative should truly be avoided. The cost of maintaining this solution and the likelihood of incomplete content is too high. Additionally the task to make a Web site or Web site content accessible is generally not out of the reach of any organization.

*See Chapter one for more information on full text versions of content.*

### **TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (K)**

This practice is costly to maintain and to test. The requirement is clear that the content on a non-full text page be identical to a full text page. Every page of content must be reviewed in order to assure that it is in compliance.

Because of the detailed testing this method requires, it is strongly recommended that this alternative be used only when there is absolutely no way to make the page accessible.

## Understanding Paragraph (I) / W3C P1 (*partial* 6.3)

(I) 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.

### DISABILITIES ADDRESSED BY THIS PARAGRAPH

The following disabilities are addressed by this paragraph:

- Potentially all disabilities referred to in the standards

### DEALING WITH SCRIPTS

All JavaScript URLs should have meaningful text so as to be useful for people with disabilities to follow. Developers should avoid event handlers as the only method for navigating or completing a page.

Problematic Handlers:

- OnClick
- OnDbClick
- OnMouseDown
- OnMouseUp
- OnMouseOver
- OnMouseOut
- onChange

Event Handlers that require no user interaction and are device independent are not problematic:

- OnLoad
- OnUnload
- OnBlur

- OnFocus

Remember that all script function should include a NOSCRIPT tag for those browsers or assistive technologies that do not have script support.

## **TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (L)**

It is easy to test your page for compliance to paragraph (l). AccVerify can quickly test an entire site or page for compliance. The identification of every JavaScript event handler is a good place to start on a comprehensive site to begin the assessment process.

## Understanding Paragraph (m)

(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).

### DISABILITIES ADDRESSED BY THIS PARAGRAPH

The following disabilities are addressed by this paragraph:

- Potentially all disabilities referred to in the standards

### DEALING WITH APPLET OR PLUG-INS

This is one of the more complex checkpoints or paragraphs with which a Web content developer or site manager needs to deal. The reason it is so complex is because it becomes both a Web content and potentially software accessibility verification issue.

The simple part of this paragraph is if a plug-in, object, applet or viewer is required then the developer must provide a link on the page so the user can get easy access to the required component.

*NOTE: If you have a link to 30 pdfs on your page there is no need to put 30 links to the PDF viewer! This is a common unnecessary practice.*

The problem with complying with this paragraph is that developers must assure that the plug-in required is accessible. Chapter Eight of this book provides more information on software accessibility.



## **TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (M)**

One must test either Web or software standards for any applet, object or plug-in used separately. Check with the provider of the plug-in or applet/object for the status of the products accessibility.

## **Understanding Paragraph (n)**

(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.

### **DISABILITIES ADDRESSED BY THIS PARAGRAPH**

The following disabilities are addressed by this paragraph:

- Potentially all disabilities referred to in the standards

### **DEALING WITH ELECTRONIC FORMS**

The most important thing to remember when providing electronic forms is that you “SHOULD” develop the forms to be accessible for both simple and sophisticated screen readers and assistive technologies. This means that the following can be used to make the form accessible:

- Predefined values in text area
- Alternative text on inputs
- Labels for easy navigation

Stay away from solutions that do not work with all screen readers.

### **TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (N)**

It is easy to test your page for compliance to paragraph (n). AccVerify can quickly test an entire site or page for compliance.

## **Understanding Paragraph (o)**

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

### **DISABILITIES ADDRESSED BY THIS PARAGRAPH**

The following disabilities are addressed by this paragraph:

- Potentially all disabilities referred to in the standards

### **DEALING WITH NAVIGATION LINKS**

Most sites are developed with structure. By having structure you most likely have sections of your Web site designed specifically for navigation of the entire site. One of the easiest ways to deal with this paragraph is to put a bookmark on the page for where the content starts and a link on the page navigation section. This allows the user to select the link to skip directly to the content bookmark that you placed on the page.

### **TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (O)**

This test is a visual checkpoint. When validating that the skip navigation links works you should visually verify that:

- The bookmark that you send the user to does not skip ANY Non-Text content that is not related to navigation
- The bookmark that you send the user to does not skip ANY Text content that is not related to navigation

## **Understanding Paragraph (p)**

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

### **DISABILITIES ADDRESSED BY THIS PARAGRAPH**

The following disabilities are addressed by this paragraph:

- Potentially all disabilities referred to in the standards

### **DEALING WITH TIMED RESPONSES**

A Timed response is best described as a page that requires some sort of user input, based on script or coding, that only gives a user a specified amount of time to perform some action before:

- The page expires
- Clearing Form Input
- Redirecting the user to another page

The completion of the action required in the time allotted may be difficult for a user with a disability covered by this paragraph and general Section 508 rule.

Any content that requires a timed response must alert the user that they are running out of time and allow them via a prompt to request more time.

### **TESTING YOUR PAGE FOR COMPLIANCE TO PARAGRAPH (P)**

This test should be performed on all pages with a timed response, it could be both a visual and / or automated test.

# 3. Introduction to Accessibility Testing

---

## *In this chapter*

- Introduction to common accessibility errors
  - Introduction to quality assurance
  - Introduction to accessibility testing with software tools
  - Using Verification, Repair, and Monitoring tools with explanations of what they can and cannot do
  - Accessibility testing and content management / current test architectures
- 

Applying accessibility testing to your current Web site quality assurance practices is the easiest way to ensure an accessible site. This chapter will introduce you to common errors and the quality assurance process.

## **Common Accessibility Errors**

The accessibility guidelines and standards provide an excellent guide for making your Web site accessible. Whether you are following W3C Priority One Guidelines or 508 Standards it is important to understand the most

common errors that Web developers make when developing a Web site. This is important from a testing perspective because if you understand the most frequent errors you can develop a proper testing methodology and plan. Common errors are divided into two groups for the purpose of this book, Design errors and Guideline / Standards errors.

Note: Guideline refers to the W3C/WCAG guidelines. The guidelines are a proposal and not a standard that Web site developers must follow if they want to achieve accessibility. Standards refer to the Section 508 *Electronic Information Technology Accessibility Standards; Final Rule*. (Actual US Federal Standards that government sites are required to meet.)

The most common design errors that impact accessibility of a Web site are:

- Use of Frames
- Overuse of java based menu systems that might look impressive but tend to be overwhelming
- Assumption that placing forms and field descriptions in tables makes the form more usable

The most common accessibility errors that prevent a site from being utilized effectively by peoples with disabilities are:

- Use of Frames that are not easily navigated
- Not using alternative text for non-text based elements
  - Forms

- Images
  - Objects
  - And other non-text elements
- Using priority two or three recommendations to try to satisfy 508 or W3C priority one rules
- Not providing a site map or contents
- Non-accessible Image maps

Sites that can address the items listed above will find themselves close to compliance or in compliance. This will have a profound impact on the overall accessibility of Internet content.

## **An Introduction to Quality Assurance**

Quality Assurance for Web sites is an essential activity for businesses or organizations that develop content or Web site applications to be used by others. It is safe to assume that any organization that has a Web site for public or internal use has in place some sort of Web site quality assurance practices. This internal QA group serves as the consumers in-house representative. When performing their tasks the QA group looks at the Web site from the perspective of the Web site visitor.

Like Software testing Web site testing starts with the standards and requirements of the product or Web site and its functionality. This means that if a Web site is initially designed to perform a set function or to provide information in a specific format then it must complete that task as specified.

The Objectives of Web Site Testing:

- The testing of a Web site combines both content editing and functionality testing with the intent of finding errors.
- The tester should strive to develop test cases that have the highest likelihood of producing errors. Remember a successful test is one that uncovers errors. (e.g. Because an object has been developed to function its best under Internet Explorer, a good test case would be to test it on all other browsers to assure the same results are obtained.)

In the case of accessibility there would be a test plan in place designed to ensure effective accessibility error detection. The task of error detection does not end with the initial “Basic” test. We must look further then the basic requirements.

Types of tests that could apply to Web site Accessibility Testing:

- Condition Testing - This test type is best defined as a true/false test. For Example, it may be a test of a specific operating system (“OS”), for which the Condition being tested. Another example might be testing performance through different Web browsers. Condition testing is generally relevant for the testing of dynamic data and not static web sites.
- Data Flow Testing - Based on the flow of inputs to a Web site application, different screens and general content will be presented. This output to the user



needs to be verified and validated for compliance to the accessibility standards. Data Flow testing is again best suited for dynamic web sites. This is more then likely to be used in a Web based application, (I.e. an online reservation system.)

- Boundry - Boundry testing can be defined as testing the limits of specific inputs. As with the previous test type, a boundry test is most frequently designed for dynamic sites where specific user inputs, OS, or User Agents could have a direct impact on the content delivered.
- Verification - Verification is the process of determining that you have met your organizations requirements and or standards for the Web site as defined. (In the case of Accessibility this may be Section 508 or W3C Priority one). Note: This test type is designed to assure that requirements are met- not if they were right in the first place.
- Validation - This test process is a check for the initial requirements. It looks at the requirements and then the final Web site to determine that while the desired responses have been produced, in response to the test, the requirements have also met their objectives. For example, an organization may have the Web site reviewed by an inside or outside group to determine usability of the site even though it complies to the accessibility rules. This assures not only compliance but usability of the end product.

- **Change Testing** - Change testing for Web sites as with software quality assurance is very important. Whether a site is dynamic or static the content will be changed or content will be added. Upon changes a page or site need to be retested. Because of the nature of authoring and Content Management tools it is possible that one change can create countless accessibility errors. For example, in an authoring tool, adding a link to an image map may clear all other alt text for the entire image map. In a content management system, changing one template could make the whole site inaccessible. Alternatively, the modification or removal of an image asset could make one or many pages inaccessible.

### ***Beta Testing***

One of the most important steps you could perform is to beta test your Web site. This means to test you page or site before going live into production. Whether you are a large organization or a small company you should make the time to do this. Once you have completed the development of the page or site, and once you have completed accessibility testing you should have the product beta tested.

This process has an outside individual look at the page to see if it performs like it was designed to. This user is not part of the products or sites qa or development team, and does not necessarily need to be a tester. Once the product / Web site is beta tested it should then be released to production.

## Introduction to Accessibility Testing

Developers or auditors of Web sites, at key checkpoints in an overall process, will determine whether objectives and or requirements are being met by testing. Accessibility testing (commonly referred to as verification) has two main components - automated and visual. In order to complete a successful audit or test of content you must understand how you can implement a successful test plan.

**Automated testing** is accomplished with either a Web site crawler (Agent) or a file scanner. The agent will imitate a user agent, obtain the content for the Web pages, and then perform an evaluation of the content to see if it complies with the guidelines.

**Visual testing** is used for verifying guideline checkpoints that do not have the structure or predictability available to allow for automated pass or fail detection.

A preferred solution would combine the automated testing and the visual testing, thereby only requiring a visual verification of pages where elements that require visual verification were found. Small organization with predominantly static content will find the task of creating an accessible Web site to be relatively simple. Mid to large sized organization will find the task to be a little more challenging.

### *Page versus Site*

When testing site content there are two instances the company should address. The first instance is content creation. The Second is existing site content. As with any

development effort, Web content or software, it is always less costly to find and correct errors before they make it to the production environment (actually become published). This means that a company needs to perform testing at both page and site level.

### **Page**

The author who develops the actual page should test it for compliance before publishing the content. This should become a final part of any process that deals with the publishing of electronic content.

### **Site**

From a site wide perspective companies should implement unattended services, either hosted or placed on their internal servers that will constantly monitor Web sites or portions of the Web sites and alert responsible parties if there is a problem that brings them out of compliance.

## ***Advanced Content Delivery***

It is becoming more common for Web content to be dynamically generated delivered via content management or publishing servers. In this case you have to consider a different type of testing, template testing. Dynamic sites use templates to deliver content. It is important in this case to test all templates before they are placed into production.

For more information on Accessibility and Dynamic Content please see chapter Five. Chapter five goes into detail on Dynamic Content and also shows how an error in templates can create catastrophic Accessibility errors on your Web site.

## Using Tools

It is becoming more common for Web site

Authors/managers to implement the use of accessibility verification, remediation and repair tools to assure site accessibility. These solutions can assist in decreasing IT overhead and human resources for development projects. This section will deal with the types of tools available and how they should be applied to your testing process.

### *Verification tools*

An accessibility verification tool is a software solution or a hosted service solution that allows you to test a page that you are working on or a group of pages of a (Logical or Physical) Web site for compliance with the accessibility standards. This technology can be instrumental in developing accessible content quickly and at a greatly decreased cost. The solutions are available in many different forms:

- **Desktop Software** - Runs on a client desktop and runs with or without user interaction. Desktop software is generally suited for testing of smaller workgroup sites Web based and local pages. It is generally not used to disseminate information to a team. *An example of a desktop software product would be HiSoftware's AccVerify Software*
- **Server Software** - Server based solutions generally run without requiring any user intervention except for initial configuration. Additionally, once configured or for initial configuration there is general a "Web-Based" interface so that after installation there is no requirement for access to the console of the server.

- **Hosted Services** - A Hosted service is similar to the Server software; the main difference is that there is no hardware requirement since the server software is installed and maintained and the service providers' facilities.

As part of your evaluation, look for solutions that provide example files demonstrating all potential accessibility issues that can be tested. This will aid in ensuring that the checks implemented comply with the standards.

Verification software should also support the following types of verification:

- **FTP** - Anonymous File Transfer Protocol
- **FTP Authentication** - File Transfer Protocol that requires a user be authenticated with a username/password pair.
- **HTTP** - A User agent capable of issuing requests and receiving responses from an anonymous web server.
- **HTTPS** - A User agent capable of issuing requests and receiving responses from a web server that supports the Secure Socket Layer.
- **HTTP(s) Authentication** - A User agent capable of issuing requests and receiving responses from an web server that supports http authentication and optionally the Secure Socket Layer.

- **Local File Scanning** - The ability to scan a local file system and identify and verify all documents that match the configured web based document extensions.
- **Local File Selection** - The ability to select local files from a users computer or server.
- **Browser Emulation (Essential for Dynamic Sites)** - Web content may differ based on the site programmer's requirements coded for each agent. Therefore, the software should have the ability to interact with a Web server or test a Web server, and in doing so identify the user agent as an agent other than it in order to test how the server would react for each agent
- **Firewall support** - The ability to send requests and receive responses via a corporate firewall.
- For training reasons there should be support for historical viewing of your accessibility status

If you want a quick and free trial of an accessibility tool that runs around the clock visit:

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

This site allows a free verification of five pages immediately. E-mail will be sent with links to reports of the accessibility status of your site!

## ***Remediation (Repair) tools***

Remediation tools are always controversial to HTML developers. The reason is the author fears that the tools will invalidate their HTML by performing bad modifications. However, repair tools can be valuable in the management of Web site accessibility.

- The remediation solution you choose should be able to work with files that are developed or stored on Windows, Unix (Linux), or Apple based systems.
- The remediation system should include full support for spell checking any alternative text representations entered to comply with accessibility.
- The remediation system should allow Web teams to collaborate on repairs for single or multiple pages.
- The remediation tool should learn the meanings of input elements, objects, and any element requiring alternative text or element content.
- The remediation tool should allow values to be edited in a library environment outside of the repair process.
- The remediation tool should allow for automated repairs where possible.

If a repair (remediation) tool is implemented properly it can prove to be essential in the process of making a site accessible. Additionally, a good repair tool decreases the training requirement for your Web team as well as decreasing the likelihood of errors.



## ***Monitoring tools***

Remember that because Web site content is dynamic - it is by nature always changing. That means that an important piece of any accessibility strategy is a solution that will let you ensure that your sites remain in compliance on an ongoing basis. From a site wide perspective companies should implement unattended services either hosted or placed on their internal servers that will, once configured, constantly monitor Web sites or portions of the Web sites and alert responsible parties if there is a problem that brings them out of compliance.

## ***What automated tools cannot do***

An automated testing tool can be used to test your site or groups of documents in an unattended manner once they are configured. It is very important to remember that “NO” tool alone can validate the absolute accessibility of your web site.

However a good tool can identify a majority of what needs to be verified visually. Additionally a good tool will let you know what pages do not need to be verified visually, based on the absence of elements that require visual verification. Remember: You will still need to assure that all visual checkpoints identified by the solution are accessible.

## ***Evaluating Accessibility Testing and Repair Tools***

As with any testing requirement for Web-based content you will have choices to make when selecting your accessibility testing solution. You should evaluate the solution you

choose to make sure that it can complete the task that you need completed. In addition to this you should verify that the tool has:

**A Common Design** - Decreases the level of training required to implement the solution.

**Complete Program Help & User Documentation** - Regardless of the tool there will be some time required to come up to speed on the use of the product. When you evaluate a tool make sure that complete documentation is available.

**The capability to automate** - Your testing tool should allow for client side or server side automation. This is important because the process of testing Accessibility can become resource intensive if the number of pages is greater than a few hundred. Additionally the solution should allow you to test pages without user interaction.

The Following Items can represent Automation capabilities:

- A Published, Public, and Documented Applications Programmers Interface (API)
- Scripting Support - The Ability to create and run test scripts.
- A Scheduling Subsystem (Generally with Server Side Products)

- Integration with the Operating Systems Scheduling Sub-System (e.g. The Windows Task Scheduler)
- Automated Accessibility Alerts

## **Existing Content Management Systems**

It is clear that with training and content engineering you can have a positive impact on the overall accessibility of your Web site. Your content management system (if applicable) is a good starting point.

By using a repair or verification tool that supports automation or by using a server-based tool as part of the content preparation, you can quickly and automatically prevent inaccessible content from being published. You can also use verification tools to assure that templates that are used to display data are also accessible. Accessibility verification and repair can be built into the pre-publishing quality assurance checks and balances of your content management system.

## **Existing Testing Architecture**

Most organizations already have in-place a Web site quality assurance testing process. Additionally this process may include the use of generalized or specialized tools and/or services.

*Accessibility testing could be retrofitted into this process if the same quality assurance team will be*

*responsible for accessibility testing or if the same test management system is being used.*

This means that if you are currently using site-testing tools, it is important that your accessibility-testing tool can be integrated with the testing application via its API or via the accessibility testing solution's API.

Testing tools, which incorporate accessibility verification, will allow you to implement site assessments, accessibility project plans, testing requirements, test management, and defect assignment and tracking.

### ***What is an API?***

API is the Acronym for Application Programmers Interface. If a product has an API it means that some or all of its functions can be accessed programmatically. This allows for automation programming and for integration with your existing systems or processes.

# 4. Developing a Strategy for Accessibility Compliance

---

## *In this chapter*

- Performing a site assessment
  - Setting goals
  - Setting design guidelines
  - Training developers
  - Scheduling your site retrofitting project
  - Introduction to unit testing
  - Introduction to conformance testing
  - Introduction to compatibility testing
  - Introduction to accessibility maintenance
  - A step-by-step approach to accessible content
  - Introduction to Intranets
  - Introduction to accessibility planning for Intranets
- 

The practice of designing accessible Web sites is not new, however, the adoption of this concept by the mainstream is! This new practice can produce new and significant training requirements.

Organizations have been pushing the design elements of their Web sites for a while, attempting to develop the richest

and most appealing Web content possible. Accessibility now has to be retrofitted into most sites as organizations become aware that they have left behind over 50 million potential customers or site content viewers. While this was not the original intent of the designers it is a problem that must be addressed.

## Looking Forward

In the previous chapters we have clearly defined accessibility guidelines and those whom they impact. After reading the previous sections you should have a better understanding of the level of effort required to build a Web site that is accessible by everyone.

An accessibility strategy will provide your organization with the ability to view accessibility from a project management perspective. This will enable your organization to:

- Allocate resources appropriately
- Track site progress
- Educate employees
- Identify problem areas
- Integrate accessibility into quality assurance and content delivery processes
- Keep a historical view of your Web site accessibility work

Most corporations and government agencies that have a Web presence have Web content or application development teams. Many of these organizations have multiple teams each with specific responsibilities across different content and design aspects of any given site or application. Each of these teams will need to be

responsible for their teams accessibility concerns and requirements. For the purposes of this book these teams will be categorized into five main groups:

- **Content Development**

A content development team consists of the people who generate the actual content that will be presented by the site. Additionally they most likely are responsible for placing content and any graphics and/or objects together on the core site. In this definition, content refers to text / writing not the graphics or functional components of the site. This applies to dynamic or static content.

- **Multimedia Development**

A Multimedia development team is responsible for the creation of rich content. Some types of rich content are:

- PowerPoint Slide Shows
- Streaming Audio or Audio Files
- Video and Streaming Video
- Flash Presentations

- **Layout and Design**

A Layout and Design team is responsible for the creation of the overall site navigation and/or layout.

- **Site Management**

A site management team is responsible for the core architecture of the Web site and its compliance to the organizations standards.

- **Graphics Development**

A graphics development team is responsible for all graphical representations or imagery for the site.

These groups are logical breakdowns of what is required to create a Web based information portal. Depending on your organizational structure there may be crossover roles and responsibilities between Web groups and designers.

## **The Right Leadership**

As with any project make sure you put the right leadership team in place. Your Accessibility project team should include individuals from every group that contributes to the development and maintenance of your Web site. In the average organization the following groups would most likely be represented:

- **Marketing** - For their knowledge of who uses the site and for their knowledge of site usage statistics.
- **Development** - For their knowledge of what skills and technologies will be needed to complete the task.
- **Public relations** - Involve your Public Relations team to determine whether or not the accessibility effort should be made public.
- **End Users** - From internal users to an outside beta tester group, actual users will help to make any accessibility project be successful.
- **Executive management** - Corporate or Executive representation on the team will help to keep the project funded.
- **Operations** - This group is generally responsible for overall Web site performance and testing. This group



will be critical in the development of a long term site accessibility strategy.

- **Professional development / training** - This group should be part of the project from the beginning so they can better understand training requirements.

The level of participation for each group will vary, however, they all will play an important role in the project.

## A Phased Approach

The following are the four basic phases of developing and maintaining an accessible Web site:

- Developing a plan - Performing a site assessment, setting your organizations goals, and setting design guidelines and standards.
- Implementing your guidelines - Training developers and retrofitting current content. All new content development should be performed based on your organizations new guidelines and standards.
- Accessibility testing - Complete testing to assure that you have complied with your new guidelines and standards.
- Accessibility maintenance - Evaluate results, automate verification of accessibility guidelines and standards.

It is highly recommended that organizations complete each of the above phases to assure that organizations goals for Web site accessibility are achieved. This phased approach is a very basic strategy based on fundamentals. Your organization may require a more comprehensive strategy or a simpler strategy. The process of planning a project could

fill multiple texts- this process represents the need for a real plan and serves as an introduction to those individuals who have not worked within a formal project process in the past.

*For example: An organization may have a terrific plan, create guidelines, do initial testing but never perform the maintenance step. In this example a site could return to a inaccessible state very quickly.*

### ***The Site Assessment***

For a small Web site the assessment may be managed by looking at each page to see where there are flaws. For larger sites, with more than several pages of content, this method becomes cumbersome, and even unmanageable. The use of an automated assessment tool is recommended.

The first step of any assessment is to determine the overall accessibility condition of your Web site. Using readily available accessibility assessment tools organizations can receive an instant snapshot of their Web sites accessibility.

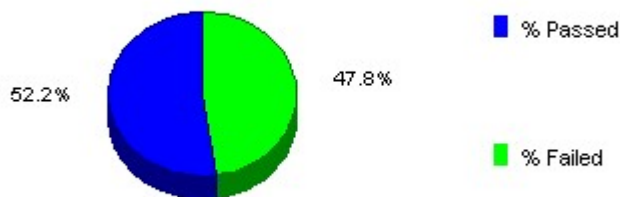


Image 3.1 - Graphical overview of a sites accessibility standard compliance produced by AccVerify Professional.

An assessment tool can further show development teams important information about which accessibility rules are being violated.

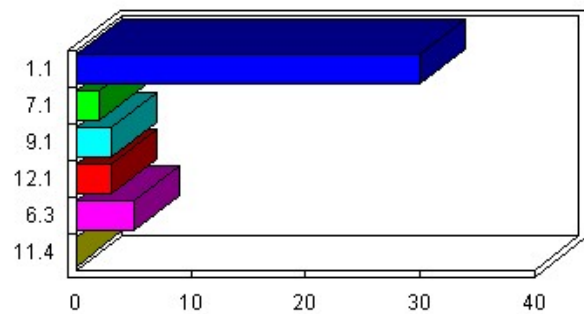


Image 3.2 - Graphical overview of a sites accessibility standard compliance down to exact detail on where errors exist produced by AccVerify Professional.

Regardless of the tool used, it is important to get a valid assessment, so that the development or management teams will be able to determine the scope and nature of the work that will be required by the team from a remediation perspective, as well as how many pages in the site will require visual verification.

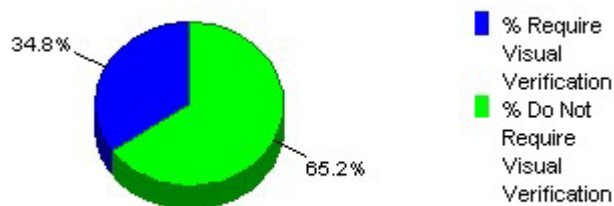
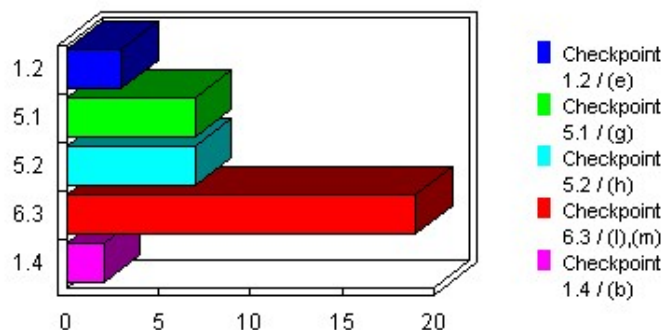


Image 3.3 - Graphical overview of the percent of site pages that will require a visual overview to assure compliance of accessibility standards, produced by AccVerify Professional.

In order to properly allocate company resources organizations will also need an overview of the individual visual verification required.



3.4 - Graphical overview of the actual elements in your site that require visual verification to assure compliance of accessibility standards, produced by AccVerify Professional.

When considering accessibility, it is important to understand how site development teams use Web elements. Your assessment tool should provide you with the following information:

## ACCESSIBILITY STATISTICS SUMMARY

With the following information organizations can adequately assign tasks required to make the accessibility project successful.

### *Image Summary*

The image summary of a Web site provides an immediate view of how many references to images exist and how many do not comply with the W3C or Section 508 Guidelines. In the example below, ten of the image references do not comply with the guidelines.

### **Image Summary**

Images: 13

Images without Alt attribute: 10

Images with Alt attribute: 3

Images with blank Alt attribute: 0

Images with null Alt attribute: 0

Image Counts by file extension:

Image File Extension	Count
.gif	12
.jpg	1

### ***Form Summary***

The form summary of a Web site provides an immediate view of how many forms exist as well as how many do not comply with the W3C or Section 508 Guidelines. In the example below, one of the Form Input Image elements does not have alternative text.

### **Form Summary**

Forms: 3

Forms with Labeled Controls: 0

Forms with Inputs using the Alt attribute: 0

Forms without Input elements: 0

Forms with Input Images not using the Alt attribute: 1

Forms without any use of TabIndex: 3

Forms without any use of AccessKey: 3

### ***Frame Summary***

The frame summary of a Web site provides an immediate view of how many frames exist and of those how many do not comply with the W3C or Section 508 Guidelines. In the

example below, none of the frames are in compliance with the W3C and Section 508 Guidelines.

### **Frame Summary**

Frames: 3

Frames without Title Attribute: 3

IFrames: 1

IFrames without element content: 1

### ***Script Summary***

The script summary of a Web site provides an immediate view of how many script elements exist and of those how many do not comply with the W3C or Section 508 Guidelines.

### **Script Summary**

Script Elements: 2

Script Elements without NOSCRIPT: 2

### ***Link Summary***

The link summary of a Web site provides immediate views of how many links are found containing the content "Click Here" or that contain links to external documents that must be accessible or must comply with accessibility guidelines.

### **Link Summary**

Links with the phrase "Click Here" in the link text: 0

Links to DOC files: 0

Links to MP3 files: 0

Links to MPG files: 0  
Links to PDF files: 0  
Links to PPT files: 0  
Links to SWF files: 0  
Links to XLS files: 0

### ***Table, Object & Applet Summary***

These Web site summaries provide valuable information on compliance, or lack there of, for these essential design and/or interactive content elements.

#### **Table Summary**

Tables: 8  
Tables with Summary attribute: 0  
Tables with Caption: 0  
Tables with Summary and Caption: 0  
Tables with ID attribute: 8  
Tables with Cells with ID attribute: 0  
Tables with Cells that use Scope: 0  
Data Tables: 7

#### **Object Summary**

Objects: 6  
Objects without element content: 4

#### **Applet Summary**

Applets: 1  
Applets without either the Alt attribute or element content: 1

## ***Setting Goals***

Setting formal goals for accessibility compliance within your organization is critical to the project success. The goals portion of your project depends heavily on:

- The level of accessibility that you want to accomplish
- The current accessibility based on your assessment
- Defining training requirements based on the assessment
- Company resources available for the project

## ***Setting Design Guidelines***

The level of accessibility compliance that is highly recommended for all organizations is W3C Priority One and US Section 508. This is an achievable goal and it will allow organizations to satisfy both US and international guidelines. The other guidelines and/or recommendations available today are not as widely accepted. It would be unwise to set organizational goals higher until the expanded recommendations become more widely accepted.

## **USING THE ASSESSMENT**

Once you have a valid Web site accessibility assessment you can begin to map the steps that will be required to make your site accessible. This should begin with the training and education of the specific development teams responsible for your entire Web content.

### **Content Development**

If content development is dynamic in nature, then it may be that a high percentage of content delivery methods need to be made accessible, which will require the training of developers in these



methodologies. Developers will need to be trained in the creation and testing of accessible templates, as well as accessible data that will populate the templates on the end-user queries. For example, in the case of a product catalog, a development team may need to automate the entry of alt text for images displayed. Dynamic site developers need to be trained about all accessibility concerns based on current development practice. Training should also be provided to match technology updates and changes.

### **Multimedia Development**

The use of multimedia for your Web content should be carefully assessed as significant development time may be required to implement accessibility. The training regiment should educate multimedia developers in the accessibility issues associated with multimedia Web elements, and should review tools and technologies that are available to address or remediate these issues.

### **Site Layout and Design**

Accessibility should become an element that is part of every step of the Web site design process. The assessment is the first level of training and going forward you may wish to assign a member of the team to validate the accessibility of any proposed design modifications.

### **Site Management**

Site managers, who currently deal with site flaws and errors, need to be trained on the accessibility requirements. Accessibility testing should be

implemented as part of the Quality Assurance practices that already exist within organizations. Developing and assuring accessible content prior to publication, as well as monitoring of live Web sites can assist site managers in assuring that their sites are accessible. Verification of compliance will be most effective if it is incorporated into the standard organizational operating procedures and regiment.

### **Graphics Development**

Graphic developers need to be trained on the accessibility requirements for graphical content. Training on how to write text equivalents and deliver a base text equivalent with every image should be provided.

## **Implementing Organizational guidelines**

With the accessibility goals and guidelines developed the next phase of the project begins. In this phase developers and quality assurance teams will receive training and the process of retrofitting the company Web site will begin. As policy, companies should either issue a moritorium on all new Web site development or set standards to assure that all new pages, templates, and content is developed in compliance with the new standards.

### ***Training Developers***

Although most organizations have addressed at least some accessibility issues in their Web content, others have addressed none, and few have addressed all. With this in mind organizations must first make an assessment of what

is being done right and what is not being developed so that it is accessible.

The Accessibility assessment is the most important phase of determining a company's accessibility strategy. If you are not sure about the problem and its size you will not be able to provide the proper resources and planning necessary to come up with a solution.

***Retrofit current content and develop all new content according to guidelines***

The site assessment allows for developing project goals. Once a company has a project goals defined it is easy to determine what project steps are required, milestones that will be achieved, and what the plan will require. In determining resources consider several factors, which determine the actual number of resources that will be required for the project.

- A. Total amount of content or the number of static pages
- B. Number of templates for dynamic content
- C. Time required to fix one page or template (in hours)
- D. Growth of Web site (How many new pages are added each week or month)
- E. Total resources available to the project
- F. Priority of project
- G. Project start date
- H. Required project completion date

The analysis of the items listed above will allow common project management systems to determine the number of

resources required. If your company does not use a project management system perform the following calculation:

$A+B \times C$  = total hours required for site remediation

H-G= total days allotted for the project

Base an average work day on six (6) hours then divide total hours by six (6) to come up with number of resource days required to complete the project.

**For example:** If 40 resource days are required to complete the project and if you only have 20 actual days to complete the project. Divide the resource days (40) by Requirement days (20) to come up with the number of resources required. In this example the number of resources required to complete the project on time would be two.

Once the project is clearly defined and resources are allocated a company is ready to begin the accessibility remediation project.

## Accessibility Testing

The testing section of the phased approach applies two distinct testing types. “Developer” and “User” testing.

The **developer** is responsible for following the guidelines and assuring that templates and/or static pages comply to Accessibility Guidelines before they are delivered to final Quality Assurance or User testing. In the software testing world the equivalent would be Unit or “White Box” testing.

The **user** testing is normally the responsibility of the quality assurance team. This team can be a formal testing team or for smaller organizations, simply another developer that is trained in testing web pages. This team is responsible for conformance testing and compatability testing.

### ***Unit Testing***

Developers should test their work utilizing an accessibility checklist, either manually or automatically generated that allows them to verify that every page or template produces accessible output. For this phase we recommend working with testing technologies that allow a developer to test the page or template during the development process. It is important that the tool provide both detailed error reports and page checklist reports. Remember, this effort should be formalized, the cost of fixing errors found after delivery to quality assurance or production will cost from three to five times more to correct.

### ***Conformance Testing***

Conformance testing verifies that your site, page or template implementation conforms to accessibility standards and/or guidelines. As with unit testing it is recommended that you use automated testing tools to more effectively determine your Web sites compliance.

### ***Compatibility Testing***

Compatibility testing assures compatibility of a Web based application or Web site when viewed by different browsers, and operating systems. Although compatibility testing can be performed manually, we recommend that an automated testing tool be used to handle this stage of testing. It is important that the tool provides the capability of using test

configurations and scripts to minimize the number of resource hours required for this testing.

### ***Assistive Technology Testing***

The quality assurance team should also be responsible for the verification of Web site content presentation via assistive technology. In this phase a tester will actually test the page against assistive technologies to assure that the true meaning of the content is clearly communicated and/or presented to the site visitor.

For Example:

- Use the IBM Home Page Reader to review a site for accessibility for those with sight related disability.
- Print your page out on a back and white printer to assure compliance to guidelines related to the use of color.

### **Accessibility maintenance**

Once the previous phases have been completed a plan should be in place that provides for ongoing testing and development to assure compliance with organizational guidelines and standards. The previous chapter introduced accessibility monitoring. In the maintenance phase you will want to institute site monitoring and:

- Continued training - The process of continued professional development for site developers and quality assurance testers.
- Defect reporting and remediation procedures

The phased approach offers the best opportunity for Web site compliance to accessibility guidelines and standards. Companies can look forward to the continued accessibility of their Web content once they develop the above disciplines.

## **Introduction to Intranets**

An Intranet is a private network of computers. This network can be utilized by authorized users of the network owner. Intranets are utilized for internal distribution of information, communication, and/or application sharing.

The most common uses of Intranets are:

- Information distribution and publishing
- Corporate communications
- Corporate management
- Corporate Application sharing

One of the general goals of the Intranet is to decrease individual workstation and maintenance cost while increasing knowledge sharing. This task has clearly been accomplished by many organizations.

## ***Internet Sites vs. Intranet Sites***

Internets and Intranets have many common characteristics and are based on identical technologies. The main difference is that a company owns their Intranet while the Internet is publicly available. Some companies expand Intranet capabilities by making portions of their resources available through the Internet in a secure manner; this is referred to as an Extranet. Intranet and Extranet site access is generally controlled by a user access security scheme.

## *Creating an accessible Intranet*

The previous sections of this book have covered the accessibility issues, standards, and challenges. However, the book would be incomplete if it did not cover Intranet accessibility. Companies that are committed to developing accessible information and providing accessible solutions for their employees to complete their tasks **must** make their Intranets accessible.

## *Intranet Priorities*

Intranets almost always have more content and applications than an Internet site. The task of making an Intranet site accessible can seem impossible! In fact it is easier. The reason that it is easier to correct accessibility issues for Intranets is because they are private. This allows companies to have greater control over the project. Follow the same phased approach that was described above to develop your Intranet plan and goals. Intranet accessibility projects should be divided and ordered by priority as follows:

- Develop an Intranet accessibility policy
- Make accessible all Intranet applications that employees require to communicate with other employees.
- Make accessible all information repositories required for employees to perform their job.
- Retrofit all other Intranet documents and applications.



The above tasks should be planned. Organizations should set realistic goals and develop metrics that can be used to measure their success. It is highly recommended that you make this process of Intranet accessibility a very public process. Ask for and accept feedback from your employees.

### **Take Small Steps for large gains in Accessibility**

A Fortune 100 site, recently requested a review by HiSoftware's AccessibilityWATCH. The site had a failure rate of over 96% of all their pages. When the company was presented with this information, it appeared to be an overwhelming remediation task and one that they did not believe the company would approve.

Using the Assessment technologies mentioned above the company was able to separate the errors and by doing this some interesting information was discovered.

- If all Images were fixed 62% of all pages would pass the accessibility requirements.
- If all input elements were fixed then 72% of all pages would pass
- If all objects and applets were fixed then 77% of all pages would pass
- If all Frame pages were fixed then 92% of all pages would pass

Knowing this information provided them with the belief that the problem was not as large or overwhelming as it had initially seemed and that they could, over time, provide a

project plan that would have dramatically improve their Web site accessibility levels.

Remember Rome was not built in a day and you should take small steps to begin with so that you can:

- Properly address training
- Plan out your accessibility remediation project
- See and chart the positive results from your efforts

# 5. Accessibility & Dynamic Content

---

## *In this chapter*

- An Introduction to common types of dynamic content
  - Introduction to database generated content
  - What browser emulation is and why it needs to be performed as part of your test process
  - Case Study I - Common Issues with dynamic content
  - Case Study II - Submitting your site to a popular search engine
- 

## **Introducing Dynamic Content**

The term “dynamic content” is used to describe many different types of content delivered from different source types. This section of the book will deal with the testing and evaluation of dynamic Web site content. The following types of dynamic content will be discussed.

- Content delivered specified on the capabilities of a particular browser.
- Content delivered based on users site login information.

- Content delivered based on a users query to a database.
- Content delivered based on specific server side programming.
- Content delivered because of server side processing.

Dynamic content introduces special issues. Simply testing the template may not assure accessibility of the final product. The only way to be assured that specific accessibility standards are met is to test the sites “live.”

This means if you have a site that presents different data, based on the browser that is used to access the site, you should be testing with every relevant browser. Solutions that provide accessibility testing should have support for testing sites on the Web and for testing browser emulation.

## **Common forms of Dynamic Content**

There are many different types of dynamic content and content management and delivery servers on the market today. For the purpose of this book the most basic will be covered as a means of introducing key concepts that you will need to understand in order to properly evaluate and repair accessibility errors.

### ***HTML Templates***

An underlying component of most dynamic content is the HTML template. Templates should be tested with verification tools. Templates are HTML files that have the

basic site look and feel. When the user requests information from the server, the response to the users request and the template are combined on the server and are presented back to the user. In this case, to correct accessibility you may just have to correct the template. If the template is combined with additional HTML to make the final page and there are errors in the page HTML or the programming that creates the HTML, you will also need to correct this.

A simple process should be followed when validating and correcting accessibility of a site that uses templates:

1. Validate that the templates are accessible
2. If they are not correct them so that they are accessible.
3. Validate that any programming that generates the final page produces accessible HTML.
4. If there are programming errors that prevent your site from being accessible correct them.
5. Run an accessibility assessment against your dynamically generated Web site
6. Correct errors as necessary

This process will be the most efficient when testing and repairing accessibility for a site that uses templates.

### ***Server Side Includes***

A server side include is similar to templates. When you use server side includes you are simply instructing the Web server to create one page, which contains the main page and the included content.

For Example: The navigation bar across the top of the HiSoftware Web site is accomplished with a server side include.

Server Side Includes should be evaluated in the same way as you handled templates.

1. Validate that the include files are accessible
2. If they are not correct them so that they are accessible.
3. Run an accessibility assessment against your Web site
4. Correct errors as necessary

**Note:** Testing the server side includes first is the only way to achieve results that will allow you to plan your site accessibility project.

### ***Web pages generated from a database***

Another common type of dynamic content is HTML produced via a database query selected by a user or program. In this case the HTML is generally produced as a new page to the user. Another type of dynamic content produced based on database results would be the implementation of a custom Web component that allows the user to use their browser as a general table browsing utility. Both of these types of generic database generated results present their own distinct challenges.

The most difficult type to remediate or test, from an accessibility perspective is the Web component. For the purpose of this discussion a Web component is defined as a Java Applet or ActiveX control. When using Web

components you must make sure that the control that is downloaded to the user system is accessible under the desktop software accessibility guidelines. If the Web component is not accessible you must provide alternative access to the information.

## **Web Site Structure**

It has become easy for companies to deliver dynamic content versus static content. Products like Microsoft FrontPage allow users to create many different types of shared content. Unfortunately companies have not developed the discipline required to effectively manage their content. It is recommended that you consider site asset management as part of your accessibility testing and remediation process.

### Recommended Asset categories:

- Images - All Images for a site or folder
- Templates - All Templates used to generate content for your site
- Scripts - Programmatic components of your site
- Database Access - All database scripts
- Shared - All Shared components or files
- Includes - All HTML Files used as includes
- CSS - Style sheets

Web site developers typically create a method to manage all Web site content. The challenge is to develop a content creation methodology that can be used and understood by anyone responsible for maintaining the site.

## ***Add Structure to your Web site***

Create a file system structure for your Web site if you do not already have one. For Example: all customer surveys could be stored in the survey directory. The files system will allow you to easily manage all like assets.

Develop a naming convention for your html files. The following list is an example of a naming convention that uses a prefix to the file that identifies the asset type and allows for simplified accessibility testing:

- inc -include files
- tmp - template files
- db - a database access file
- sh - a shared file

With structure and naming conventions it becomes very easy to test specific files for accessibility. Remember, if your site does not have structure you should implement it as part of your accessibility testing.

For Example: A page that is used as a Server Side include is Navbar.htm. To make this file automatically recognizable as an include file change its name to use the new convention: incNavbar.htm

## **Who should be testing Dynamic Content**

Unless you are a “one-person shop” the developer of the dynamic site should not be responsible for assuring that its accessibility is in compliance with global accessibility standards.



There should be a designated person, or team, charged with testing the dynamic site to assure that it complies with accessibility standards.

Some common myths or inaccurate generalizations about the testing of dynamic content:

- If I test the site template and it is accessible then the page generated will be accessible.
- I do not need to test my page because it is data driven.
- If I use HTML 4.0 my site is accessible.

While there can be truth to any one of these statements it is necessary to test them to prove them right or wrong.

## **Web based Applications**

Web based applications are also a form of dynamic content. A Web based application could be a news portal, driver's license renewal application or any other application that runs through the Web browser. Web based application are generally HTML based or Plug-in based.

For Example: The Web based administration system for AccessibilityWATCH.COM is 100% HTML. Though it is an application, its content is HTML and therefore it should comply with Web accessibility guidelines.

Web based applications that are developed as Java Applets or other Web components will most likely have to comply with the Software Accessibility Guidelines or paragraph (m) of the Section 508 standards.

## Browser Emulation

When a Web browser or other type of user agent submits a request for a page from a Web server, part of the request from the browser includes information about the browser itself, including the name of the agent, its version, etc (an example of a user agent is Internet Explorer). The Web server, in most cases, returns a valid Web page to the agent or Web browser. While the newest Web browsers support this content, Web developers must ensure the information is accessible to individuals who utilize older browsers.

Below is an example of some pseudo-code a developer might use in creating a Web page based on this idea.

```
<%  
var browser =  
Server.CreateObject ("MSWC.BrowserType")  
ie4 = (browser.version>=4) &&  
      (browser.browser=="IE")  
ns4 = (browser.version>=4) &&  
      (browser.browser=="Netscape")  
if (ie4)  
  { %> <P>Do something for Internet  
    Explorer 4 and greater <% }  
else if (ns4)  
  { %> <P>Do something else for Netscape 4  
    and greater <% }  
else  
  { %> <P>Give this content here for  
    everything else <% }  
%>
```

The above JavaScript code, when used in an Active Server Page (ASP) on a Web server, delivers content to the Web browser or agent based on the identity of the user agent. Depending on the design of the Web site and different options used in writing and delivering the script, the above code might not be visible to a user viewing the source either through a browser program such as Internet Explorer or Netscape, or through saving the page locally to test it. Those users might just see the code for their respective browser.

In testing a Web site for accessibility, the user wants to be sure that the Web site is accessible for all users that could view the page. This functionality is crucial. Often times, the group that reports on Web site accessibility is not the same group that develops the Web site. In these cases, the group responsible for the accessibility of the site might not be aware of the existence of browser-specific functionality and could possibly miss testing code that was written specifically for a certain brand and version of a Web browser.

Pages missed in testing can have failing elements that could go undetected by conventional means.

### **Case Study I - Common Issues with Dynamic Content as seen in a major e-commerce site**

ABC Company is a major online portal that resells software. The company resells software products, and is paid a fee for providing secure transactions and fraud protection for the vendors they represent. The Company provides product

catalog services and ordering from information stored in an Oracle database.

Software vendors/clients can add information to their Web presence on the portal, including product logos and descriptive text via a Web-based Interface or update application. A disabled user (in this case, a blind user), working with the reseller's Web based application, would face many challenges.

### ***Entering Product Information***

In order to bring products to market, the first step of using the vendor's application is for the user to log into the product management system. A quick test of the accessibility of this screen shows major section 508 and/or W3C errors.

*In the example data below*, the Priority One errors that the user will encounter could prevent them from simply logging into the system and minimally make it very difficult to navigate. This is because of the following accessibility error:

#### **Checkpoint 12.1 / (i) - Failed**

Title each frame to facilitate frame identification and navigation.

- 12.1 Rule: All FRAME elements are required to contain the 'title' attribute.
  - **Failure** - FRAME Element at Line: 9, Column: 3
  - **Failure** - FRAME Element at Line: 11, Column: 5
  - **Failure** - FRAME Element at Line: 12, Column: 5

Data 5.1 – Case Study Example

As with the login screen the inner pages have multiple accessibility problems.

Alternative Text is missing for Images.

- Labels and or Alternative Text are missing from Input Elements.

**Note:** It is easy to say that the customer of the commerce site could assign a person without disabilities to perform the tasks required to create or update the catalog.

*However,* if the customer was the government and there was a competitor that provided the same service, however in an accessible manner, the government would be compelled by law to go with the competitor.

After the customer of the e-commerce vender enters the product content via the interface, the data, which is stored in an Oracle database, is presented when the purchaser (customer) selects the product to purchase from the e-commerce customers Web site.

### ***Selling the Product***

The first screen that the customer sees is the product description page. The page consists of the following important items:

- The companies branding is visible on the page.
- Product image that has a big “Ranked # 1” text on it.
- An image that proudly displays that the store is secure, virus free and that this is guaranteed.
- There is a compelling input image that instructs the user to “Buy it now!” (The product that is.)

They also provide text that says to click on “buy it now” to order.

Now the blind user cannot:

- See company branding.
- See product image or that it is ranked #1.
- See the proud statements of how secure the store is and how it is virus free.
- Figure out how to buy the product or benefit from the compelling “Buy it now” image!

**Note:** You probably just lost a customer and it would have taken next to nothing to make this page totally accessible so that any user could have had access to all of the information.

Now let’s say the customer figured out how to get to the next page. The next page has the license agreement that must be approved before going further.

The customer is presented with four images each of which links to another page or action. They must choose the link that lets them accept the license agreement.

*This page has no alternative text, so again the customer would be forced with guessing to complete this stage of the order.*

The customer would need to click through all of the images until they found the correct image and were able to continue the order.

The next page has:

- Inaccessible tables to list the products.
- Inaccessible form inputs to select quantity.
- Inaccessible form inputs to choose shipping.
- Inaccessible images to choose platform (operating system) that the product will run on.

Note: the blind customer would find even more difficulties in entering address information or credit card information.

### ***Let's fix the problem***

It appears that there are three real problems.

1. Templates are not accessible.
2. Customer Data is not accessible.
3. Dynamically generated content is not accessible.

#### **FIXING #1**

The Web Development team should assure that all templates are accessible. The minimal they should strive for is US Section 508 Compliance but thinking globally means they should develop to the W3C Standards.

#### **FIXING #2**

All customer site modifications and data entry applications should be modified to allow for the entry of accessible

information for images and other accessibility elements that need accessible content.

**Note:** Most likely this fix will require modifications to the database storing the information.

### **FIXING #3**

The developers of the store catalog system need to modify their templates plus, the page generation code to allow for usage of the new customer data that can now be entered via the enhanced Web based application that manages customer products.

### ***When should the e-commerce company start?***

The e-commerce company should start immediately. There are many reasons but the most compelling are:

1. Perform changes based on their schedule.
2. Perform changes before losing customers.
3. Likely rise in revenue.
4. Great opportunity for press and free marketing.
5. Although it is only federal law at this point, most experts predict that within 2-4 years accessibility will be specified as part of the ADA.



## Case Study II - Submitting your Web site to a popular search engine

One of the world's leading search portals allows users to submit Web sites to their index. This submission is done by the user through entering a submission code, Web site URL and a valid e-mail address.

Below are the accessibility issues that would be faced by disabled users.

File: MajorSearchEngine.htm - **failed**

Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content). This includes: images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, ascii art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video.

- 1.1 Rule: All IMG elements are required to contain either the 'alt' or the 'longdesc' attribute.
  - **Failure** - IMG Element at Line: 104, Column: 1
  - **Failure** - IMG Element at Line: 157, Column: 44
  - **Failure** - IMG Element at Line: 234, Column: 18
- 1.1 Rule: All INPUT elements are required to contain the 'alt' attribute or use a LABEL.
  - Note: INPUT Element found at Line: 82, Column: 1 does not contain the 'alt' attribute, however, it is of type 'hidden' and does not represent an accessibility problem.
  - Note: INPUT Element found at Line: 83, Column: 1 does not contain the 'alt' attribute, however, it is of type 'hidden' and does not represent an accessibility problem.

- **Failure** - INPUT Element at Line: 111, Column: 48
- **Failure** - INPUT Element at Line: 125, Column: 29
- **Failure** - INPUT Element at Line: 133, Column: 29
- **Failure** - INPUT Element of Type IMAGE at Line: 134, Column: 1

The Impact of this accessibility issues it that a person with no vision could not add their URL to the index

Other accessibility problems for submission include:

- Alternative Text is not available for the submission code (which is an image); this is required for submission to the index.
- Alternative Text is not used nor is a label for the text box used to enter the submission code.
- Alternative Text is not used nor is a label for the text box used to enter the URL
- Alternative Text is not used nor is a label for the text box used to enter the users e-mail text.
- Alternative Text is not used nor is a label for the input element, of type image, is used to submit the information.

This seems relatively simple, however, it becomes more complex if the user cannot read the image and enter it in as the submission code. If this is the case, the user CANNOT add a page via the form, even if they were able to get by the other errors on this site.

The application also provides no instruction for submitting to the index if you cannot read the image, which is the code required for submission.

The Engine uses this practice to “Combat Spam” to their engine. This prevents automatic submission tools from submitting to their index. It is recommended that the search engine company perform the following steps to repair the accessibility errors:

- Correct all page accessibility errors.
- Provide a submission code that can be read by all users
- Provide a link on the page for users with weak or no vision where they can submit pages to the index or apply for a code that can be used to complete normal submission.



# 6. The Unseen meaning

---

## *In this chapter*

- An introduction to metadata for improved content accessibility
  - A Guide to writing text equivalents
  - Writing text equivalents for images.
  - Common practices and errors when writing alternative text
- 

This section discusses non-visual components, which can significantly impact the accessibility of a page.

## **Metadata for Accessibility**

Metadata is an excellent way to communicate information about a document or resource that directly relates to accessibility.

## *An introduction to metadata*

Metadata is information about a document. The information you use to describe a document can help users identify whether the document is helpful to them and how to locate the document readily.

The precursor to metadata is the card catalog. For each item in a library collection, there are three entries in the card catalog: title, author, and subject. A card indicates the

location of an item in the collection, and provides additional information about it, such as the publisher, format, genre, date of publication, and volume size. While the card catalog serves as a database for the library, metadata goes a step further - metadata becomes part of the electronic file and remains in the source code no matter where the file moves. When you use metadata correctly, you can provide helpful information about a file to the user. The use of metadata has been recommended by W3C®, or World Wide Web Consortium, as a Priority 2 Checkpoint for Web accessibility.

### ***Metadata for Accessibility***

Checkpoint 13.2 of the W3C accessibility guidelines states that you should provide metadata to add semantic information to pages and sites.

You should include author name, company rights, type of content, etc...

Metadata is stored in the form of Meta tags. There are some common Meta tags that are recognized readily by indexing services on the Internet. In addition, you can create custom Meta tags that are helpful to members of your organization.

The most important part of using metadata is that you apply it uniformly to your document collection and that you use it accurately. When you use metadata inappropriately, in an attempt to gain higher visibility on the Internet, you can jeopardize your ranking with search engines. Additionally from an accessibility standpoint you are basically providing misleading or inaccurate information regarding a resource.

## ***The structure of the Meta tag***

The Meta tag is placed in the HEAD section of an HTML document.

Example:

```
<HEAD>  
  <Meta name="Author" content="Yonaitis">  
</HEAD>
```

You can place as many metadata elements in the document as are required to adequately describe the resource.

## ***Choosing Meta tags***

Although many organizations have recommended the use of metadata to boost usability and enhance accessibility, it is difficult to find specific information about the Meta tags that should be used. Below are some of the most commonly utilized Meta tags.

### **Keywords**

Each document should have keywords. The words you use to describe your document should occur in the document body text.

### **Description**

Each document should have a description. The words you use to describe your document should occur in the document body text.

**Author**

Each document should have at least one author. The author can be an organization, one or more individuals, or both.

**Language**

The language Meta tag indicates the primary language of a document. There is a list of common language codes. For example, the language code for United States English is en-us.

**Robots**

The robots Meta tag value provides instructions to crawlers on how to crawl, or index, the document and other documents linked to it.

**Rating**

The rating tag reveals information about the content of your document to users. This helps users to screen information that might not be appropriate for all viewers.

**Copyright**

The copyright Meta tag includes the name of the copyright holder and a statement of permissible use.

***Adding Meta tags to documents***

You can add metadata directly to the head section of documents using a Web editor, like Microsoft® FrontPage®. This manual method works for document authors who need



to add metadata to describe the contents of a particular document.

Some metadata values should be consistent across your document collection. This helps avoid inconsistencies and data entry mistakes by document authors. There are software tools to help you add metadata systematically to individual documents, groups of documents, or an entire document collection.

## Writing Text Equivalents

Accessibility with regard to Web content can have two meanings:

1. A user can gain access to the content; this is the more traditional meaning.
2. A user can understand (gather the meaning of) the content. This is the challenge!

Text equivalents are necessary for all non-text elements.

### *Tips for writing text equivalents*

When you write text equivalents, you are providing information about non-text Web elements to people with disabilities that would otherwise be prevented from having access to the information.

To minimize confusion for the user, consider the following recommendations:

- If you are linking to a document: Note what kind of document you are linking to, remember if the

document requires a plug-in to view you MUST provide information on where the user can get the plug-in and the version required.

- Fully describe the meaning and purpose of any image.
- When you refer to numbers, be clear about the units of measure.
- When you describe tables or graphs, describe the general characteristics of both the x and y-axis if applicable.

### ***Branding and Accessibility Guidelines***

There has been much discussion on the use of the word “Logo” in alternative text. This brings up a very interesting intersection between branding and what is best for the site visitor that can not see images because of a technology limitation or physical disability.

A company will want it to be clear that every page is part of their Web property. In this case the logo and meaning are very important. It is recommended that the Web site also provide a long description that fully describes the logo and what the site visitor should be taking away with them as the meaning of the logo.

### ***Prevent Meaningless Link Text***

One of the major problems that users of screen readers encounter is the use of repetitive or meaningless link text, which is used to direct users to additional site content.

This is a problem because the screen reader will gather all of the links on a page. Then the links will be presented in a manner that is easy for the user to select the links to go to

the additional resources. When the screen reader does this it does not bring in all of the associate text around the link.

Because of this common conventions that designers use for visual navigation can cause problems to those with blind users or those with limited vision.

### **Example of the problem**

The list below is of some content that appears on a web site and how the use of “Click Here” can make it difficult for a user of assistive technology to navigate the site.

The Italic Text Represents the Hyperlink Text

Product “a” is a Fruit Substitute recommended by  
many leading physicians  
For more information on product “a” *click here*

Product “b” is a Meat Substitute recommended by  
many leading physicians  
For more information on product “b” *click here*

Product “c” is a Grain Substitute recommended by  
many leading physicians  
For more information on product “c” *click here*

In the above example the screen reader would bring up the link text “Click Here” for all of the products, a-c. This would not have meaning to someone who could not see the placement of the link.

The proper way build the links would have the link text be more descriptive.

## The Italic Text Represents the Hyperlink Text

Product “a” is a Fruit Substitute recommended by many leading physicians

*For more information on product “a” click here*

Product “b” is a Meat Substitute recommended by many leading physicians

*For more information on product “b” click here*

Product “c” is a Grain Substitute recommended by many leading physicians

*For more information on product “c” click here*

In the above link text the full line that has click here in it is now the link, allowing for easy identification of what the link actually provides and where it goes.

## Accessibility and Images

Images are used for their meaning and in most cases add meaning because of their location or function on the page. With this in mind you need to provide full and qualified alternative text representation for the image so that people with disabilities, who cannot view the image, for any reason, understand the full meaning of the image. In developing exclusively to the standards, you may miss the meaning and the importance of the rules. If you do this you can get by using verification tools but your site may NOT be accessible.

## ***Common but FLAWED practices***

The following practices should not be followed, as they do the exact opposite of the desired result for using alt text with images.

### **Marketing Play**

Many developers will use the alt text properties of images to enter marketing keywords about their Web site rather than to describe the image or its purpose. This is accomplished by entering keywords versus descriptive text. Example: there is an image of a company logo, the alt text should read alt="My Company Logo" instead the alternative text says alt="Keyword 1, Keyword 2, Keyword 3, Keyword Phrase, Keyword Phrase 2, Keyword Phrase 3..."

This practice should cease and corrections should be made to any text that has been entered incorrectly.

### **Image Name as Alternative Text**

In the past there were many editors, or dynamic generation systems, which input the image name as alternative text.

Example:

```

```

This clearly has no meaning to the average person viewing the page. All editors need to be informed on correctly adding accessible alt text. Any incorrect alternative text should also be updated.

## Incomplete Text

This is also a common problem. In order to satisfy accessibility requirements some organizations are rushing to add alternative text without taking the time to make it meaningful.

### Example:

There is an image that is a link to a priority service; the image is 100% text and a gradient background.

The image text reads:

*Be more competitive with Priority service – Go there now.*

The alternative text reads:

*Alt="Priority Service"*

Although alternative text is provided in this scenario, it is incorrect, as it does not clearly represent the significance of the image. It would take minimal effort to modify the alternative text to accurately represent the image, and its intended meaning.

Once modified, the alternative text reads:

*Alt=" Be more competitive with Priority Service - go there now!"*

It is easy to see how this difference in content has conveyed new meaning to the image. The effort associated in altering the text, and making the image "accessible" was minimal. One point to note is that

when the accessibility tasks are done correctly originally there is no related cost to creating and maintaining an accessible site.

### **Make it Readable**

Many WYSIWYG or HTML Editors provide no default facility for spell checking element properties like ALT. With this in mind, remember to verify the spelling of your alternative text. Note: It is difficult to read “car” from “cra”. Make sure that your text is readable and has correct spelling.

### ***Common Error to Avoid - Text as Image***

The use of Alternative text for images can be confusing; especially around an image that is simply formatted text. It is common for designers to do this to protect and or control the look of text.

The Problem occurs when the site developer or accessibility expert puts the design meaning versus the screen meaning for the alt text. In the example below you have the HiSoftware Publishing Tag line as an image.

**HiSoftware Publishing - Peel  
open the cover of one of  
these useful books**

Design: HiSoftware Publishing Tag Line

Alt Text: HiSoftware Publishing - Peel open the cover of one of these useful books

So if we misinterpret the alternative text requirement and put the designers meaning versus what the intent of the image to the site visitor is we will not satisfy the requirement.

Follow a Simple rule: If there is text in the image then the same text should be in the alternative text attribute

### ***Good Practices***

Although, it is easy to list the bad practices, it is more important to focus on the simple steps associated with providing accessible alternative text for images. The book will now introduce some good practices and suggestions on providing meaningful alternative text for images. Below is a list of recommended practices:

- Spell check your alternative text.
- Use punctuation, if applicable.
- Match text to the meaning of image.
- Always read back the alternative text to ensure it will convey the intended meaning.

Some of the most common questions, in regards to providing accessible alternative text, are listed below:

#### When should I address alternative text?

The application of alternative text for images should be performed as soon as they are placed on a page. This will ensure accuracy and efficiency.

In my organization we submit content to a graphics designer and they provide an image, I cannot always



adequately articulate what the designer meant in alt text  
what do you recommend?

This is a good question; it addresses a very important issue for organizations that use a different design and content group to create the final page.

The obvious answer is to ask the designer why this image and why use it.

But more importantly perhaps the company's delivery policy of graphics needs to be changed to have the designer send not only the image but also the base alternative text.

Secondly, the page designer may add more meaning through the placement of the image as well as the function it performs as either a link or image map.

My image is transparent and cannot be seen, it is used  
merely for spacing of content as a design tool. Do I need  
alternative text?

The best practice is to use alternative text with a value of a space because it effectively translates its use.

Example:

```

```

My image cannot be described briefly. Should I use  
alternative text greater than 80 characters to describe my  
image?

When an image cannot be described briefly, use the `longdesc` attribute. The `longdesc` attribute allows you to indicate the URL that contains a long description of the image.

This might be useful for images that convey detailed information to the user, such as schematic diagrams and maps.

# 7. Tutorial: Correcting Accessibility Errors

---

## *In this chapter*

- Learn how to make IMAGES and FORMS accessible
  - Learn how to make APPLETS and OBJECTS accessible
  - Learn to create a D-Link and an IMAGE with the “longdesc” Attribute
  - Learn how to create accessible inline frames
  - Learn the proper way to create a bulleted list that uses images for bullets
  - Learn how to create redundant text links for Server-Side image maps
  - Learn how to work with color
  - Learn how to produce accessible image maps
  - Learn why you need to avoid BLINK and MARQUEE
  - Learn how to make a TABLE accessible
  - Learn how to correct common SCRIPT accessibility errors
  - Learn how to follow Section 508 paragraph (o)
  - Full examples for every accessibility task
-

This section of the book provides guidelines and examples designed to teach the reader how to develop accessible content. The section is organized with the following conventions:

Section + Rule (Section 508 / W3C)

Current HTML - original HTML example

Corrected HTML - accessible HTML

**Note:** If you have more questions on the rules themselves or on what disabilities are addressed by a specific rule, please see Chapter Two.

For this tutorial we will look at different pages on the Internet, these pages are designed to cover specific accessibility defects. Those developing a site to meet Section 508 standards or W3C guidelines will find this section valuable.

## Using the tutorial

The tutorial can be used as a workshop to introduce you to Section 508 or W3C/WCAG accessibility guidelines. It can also be used as a general reference on how to correct specific accessibility issues.

If you are new to creating accessible content it is recommend that you use it as a workshop. Once you have completed the examples you should be ready to apply what you have learned to your Web site.

## Section 1 - Section 508 (a) W3C Guidelines 1.1

The demo file for this section can be downloaded from the internet

<http://www.hisoftware.com/uaccess/fpdemo1.htm>

Remember to follow along in your Web page editor you must first download the example. To do this: Open the above page in Internet Explorer and save it locally by completing the following steps.

- 1 Open the above page in your browser
- 2 Select File |Save As
- 3 Type in fpdemo1.htm in the filename input box, make sure you selected to save as complete web page so you will have all demo files.
- 4 Click on the Save button

*If you purchased the eBook these files were included in a zip file called demos.zip. Extract this file to your hard drive.*

Now open the saved file in your Web page editor. The accessibility issues that will be covered by section one are:

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

- All IMG elements are required to contain either the 'alt' or the 'longdesc' attribute.
- All INPUT elements are required to contain the 'alt' attribute or use a LABEL.

- All OBJECT elements are required to contain element content.
- All APPLET elements are required to contain both element content and the alt attribute.
- All IFRAME elements are required to contain element content.
- All Anchor elements found within MAP elements are required to contain the 'alt' attribute.
- All AREA elements are required to contain the 'alt' attribute.

## ***1. Simple Image Requiring Alternative Text***

The first Image represents “conserve”. The images purpose is simply to add the meaning of conserve. The rule states all IMG elements are required to contain either the 'alt' or the 'longdesc' attribute.



This image requires alternative text.

### **Current HTML - EXAMPLE 1**

In the current HTML the alt attribute is missing.

```
<IMG border="0" src="../images/j0293240.gif"
width="165" height="122">
```

### **Corrected HTML - EXAMPLE 1**

In the corrected HTML the alt attribute is used.

```
<IMG border="0" src="../images/j0293240.gif"
width="165" height="122" alt="Conserve Icon">
```

## ***2. Same image: also serves as a navigation link***

This example demonstrates an image with a navigation purpose. The purpose is to provide a navigation link to the HiSoftware home page. The rule states all IMG elements are required to contain either the 'alt' or the 'longdesc' attribute.

### **Current HTML - EXAMPLE 2**

In the current HTML the alt attribute is missing.

```
<IMG border="0" src="../images/j0293240.gif"
width="165" height="122">
```

### **Corrected HTML - EXAMPLE 2**

In the corrected HTML the alt attribute is used and the alt text also shows the purpose of the image.

```
<a href="http://www.hisoftware.com/">
<IMG border="0" src="../images/j0293240.gif"
alt="Conservation Symbol and link to HiSoftware
Home Page"width="165" height="122"></a>
```

**Note:** To create effective alternative text for your non-text elements makes sure you:

- Follow the guide in this book for developing alternative text
- Use a spell checker



### ***3. An image requiring a long description***

This example demonstrates the use of “longdesc.” It would be difficult to describe this image with alternative text only. In this case a long description should be used. The rule states that all IMG elements are required to contain either the 'alt' or the 'longdesc' attribute.



#### **Current HTML - EXAMPLE 3**

In the current HTML the alt attribute is missing and there is no longdesc attribute.

```
</font></p>
```

#### **Corrected HTML - EXAMPLE 3**

In the corrected HTML the alt attribute and longdesc attribute are used. In addition to this a d-Link was used.

```

<A href="AboutSpace.html">[D]</A>
```

#### ***4. Object MS Office Spreadsheet***

Example four demonstrates the proper accessibility coding for an ActiveX object. The object is of a Microsoft Office Web component. The rule states All OBJECT elements are required to contain element content.

##### **Current HTML - EXAMPLE 4**

The element content is missing from the current HTML.

```
<object classid="clsid:0002E551-0000-0000-C000-000000000046" id="Spreadsheet1" width="319" height="249">
<param name="DataType" value="XMLDATA">
</object>
```

##### **Corrected HTML - EXAMPLE 4**

In the corrected HTML the element content is present.

```
<object classid="clsid:0002E551-0000-0000-C000-000000000046" id="Spreadsheet1" width="319" height="249">
<param name="DataType" value="XMLDATA">
```

*Your browser does not support this object.*  
*This is an object and it represents data that is also available at*  
<http://www.hisoftware.com/uaccess/demo1.htm>

```
</object>
```

## 5. Applet

Example five demonstrates the proper accessibility coding for a Java applet. The rule states that all APPLET elements are required to contain both element content and the alt attribute.

### Current HTML - EXAMPLE 5

In the current HTML the element content and alt attribute are missing.

```
<applet width="128" height="128"  
code=""  
codebase="http://Example.js.class/">  
</applet>
```

### Corrected HTML - EXAMPLE 5

In the corrected HTML the element content and the alt attribute are present.

```
<applet width="128" height="128"  
code="" alt="My Applet Description"  
codebase="http://Example.js.class/">
```

*This is a navigation applet. Alternative navigation  
can be found at:*

<http://www.hisoftware.com/uaccess/navigate.html>

```
</applet>
```

## 6. Input Elements (also 508 (n))

Example six demonstrates the proper accessibility coding for a standard HTML form. The rule states that all INPUT elements are required to contain the 'alt' attribute or use a LABEL.

### Current HTML - EXAMPLE 6

In the current HTML the alternative text for the input element is not present.

```
<Form method="POST" action="-WEBBOT-">  
<Input type="text" name="T1" size="20">  
</form>
```

### Corrected HTML - EXAMPLE 6

In the corrected HTML the alternative text for the input element is present.

```
<Form method="POST" action="-WEBBOT-">  
<Input type="text" name="T1" size="20" alt="First  
Name"> </form>
```

**Adding LABELS:** In the corrected HTML the alternative text for the input element is present and LABELS have been added.

```
<Form method="POST" action="-WEBBOT-">  
<Label for="T1"> User Name: </label>  
<Input type="text" id="T1" name="T1" size="20"  
alt="First Name"> </form>
```

**Note:** Using both Label and Alternative text gives you increased usability of your form.

## ***7. Inline Frame (IFRAME)***

Example seven demonstrates the proper accessibility coding for Inline frames. Inline frames prove an excellent TABLE or FRAME alternative. The rule states that all IFRAME elements are required to contain element content.

### **Current HTML - EXAMPLE 7**

In the current HTML the element content for the IFRAME element is not present.

```
<iframe name="I1" SRC="sample.htm">
</iframe>
```

### **Corrected HTML- EXAMPLE 7**

In the corrected HTML the element content is present.

```
<iframe name="I1" SRC="sample.htm">
```

Your browser does not support inline frames or is currently configured not to display inline frames.

Content can be viewed at actual source page:

<http://www.hisoftware.com/sample.htm>

```
</iframe>
```

## ***8. Image Map Example (Section 508 (f))***

Example eight demonstrates the proper accessibility coding for an Image Map. The rule states that all AREA elements are required to contain the 'alt' attribute.

### **Current HTML - EXAMPLE 8**

In the current HTML the alternative text for the area elements and alternative text for the image are not present.

```
<map name="FMap0">
  <area href="#Go Here for Links!!!" shape="rect"
  coords="13, 5, 172, 75">
  <area href="#Go Here for Links!!!" shape="rect"
  coords="8, 92, 119, 187">
  <area href="#Go Here for Links!!!" shape="rect"
  coords="125, 91, 191, 180">
</map>

```

### **Corrected HTML - EXAMPLE 8**

In the corrected HTML the alternative text for the area elements and alternative text for the image are present.

```
<map name="FMap0">
  <area href="#Go Here for Links!!!" shape="rect"
  coords="13, 5, 172, 75" alt="My Alternative Text">
  <area href="#Go Here for Links!!!" shape="rect"
  coords="8, 92, 119, 187" alt="My Alternative Text">
  <area href="#Go Here for Links!!!" shape="rect"
  coords="125, 91, 191, 180" alt="My Alternative
  Text">
</map>

```

## ***9. Images As Bullets Example***

Example nine demonstrates the proper accessibility coding for a bullet list, where the bullets are actually images. The rule states All IMG elements are required to contain either the 'alt' or the 'longdesc' attribute.

### **Current HTML - EXAMPLE 9**

In the current HTML there are five products listed, these products have an image before the product name to create a fashionable bulleted list.

```
<p>HiSoftware Accessibility Products</p>
<p>
Understanding Accessibility Book</p>
```

```
<p>
AccVerify</p>
```

```
<p>
AccRepair</p>
```

```
<p>
AccMonitor</p>
```

```
<p>
AccessibilityWATCH</p>
```

### **Corrected HTML - EXAMPLE 9**

The guidelines state that the purpose of the image is very important, the purpose in this case has nothing to do with the visual representation, and it is merely a bullet. This is definitely an exception to previously stated rules. The

exception is: describing the image in the alternative text would not be effective instead we will describe the purpose.

HiSoftware Accessibility Products

<p>

Understanding Accessibility Book</p>

<p>

AccVerify</p>

<p>

AccRepair</p>

<p>

AccMonitor</p>

<p>

AccessibilityWATCH</p>

In this case the asterisks clearly describe the value and purpose of the images as bullets.



## **Section 2 - Section 508 (b) W3C Guidelines 1.1**

This section does not contain demonstration files but simply lists tips for Section 508 (b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.

### ***Audio Files***

There are different types of audio files-all require text equivalents. Consider the following audio files:

- Stand-alone audio files, such as music that plays during a multimedia presentation or at the request of the user
- Interactive audio files, such as an alert tone, that only plays when the user has performed a specific task

### **CREATING ACCESSIBLE AUDIO FILES**

You have several options if you need to provide equivalent text for sound and audio files.

#### **Static Text Equivalent**

You can display static, or non-moving, text during the sound. This technique is most effective for sounds that require user input, such as stand-alone audio or interactive audio.

Stand-alone audio is a file that the user plays by some action or command. For example, Clicking on an element plays an audio file.

Interactive audio is sound that occurs because the user has performed an action or command. For example, any error or interactive action that results in an audio warning or prompt based on user action must also have a visual equivalent.

### **Link to Text Transcript**

You can provide a link to a text transcript of an audio file. This technique is most effective for stand-alone audio files, such as music.

When you provide a link to the text transcript, position the link near the top of the page, frame, or table cell. When you create a text transcript, use a description that accounts for both spoken and non-spoken sound, and be sure to differentiate between them.

For example, suppose you are viewing a page about United States history. Near the top of the page there are two links:

- Listen to the earliest recording of the national anthem.
- See a text transcript of the earliest recording of the national anthem.

However, if the page automatically plays the national anthem, there should be some indication to the user that sound is being used. The best technique for sounds that plays automatically is static text done by scripting.

## Section 3 - Section 508 (c) W3C Guidelines 2.1

The demo file for this section can be downloaded from the internet

<http://www.hisoftware.com/uaccess/fpdemo3.htm>

Remember to follow along in your Web page editor you must first download the example. To do this: Open the above page in Internet Explorer and save it locally by completing the following steps.

1. Open the above page in your browser
2. Select File |Save As
3. Type in fpdemo3.htm in the filename input box, make sure you selected to save as complete web page so you will have all demo files.
4. Click on the Save button

*If you purchased the eBook these files were included in a zip file called demos.zip. Extract this file to your hard drive.*

Now open the saved file in your Web page editor and we are ready to begin. The accessibility issue that will be covered by section three is:

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

## ***1. Colored Text as the only way to determine a required step***

Example one demonstrates an incorrect use of color. Color is used to communicate user navigation. This example has the user perform an action based on the color of text. This could be a desirable design practice, however it could be meaningless to someone who had vision disabilities that prevented him or her from identifying the color.

### **Current HTML - EXAMPLE 1**

In the current HTML the color is the only way to determine how to go to the correct product.

```
Thank you for your interest in our product. For
those with:</font>
<p><font size="2" face="Verdana">Product A Follow
Instructions that are in Red</font></p>
<p><font size="2" face="Verdana">Product B Follow
Instructions that are in Green</font></p>
<p><font color="#FF0000" size="2"
face="Verdana">Register your Product
Here</font></p>
<p><font color="#008000" size="2"
face="Verdana">Register your Product
Here</font></p>
```

### **Information**

This HTML could be corrected by simply providing more descriptive text. Developing accessible pages does not prevent you from using color. You must simply take into account that a Web site visitor may be blind or colorblind. If this is the case then visual prompts will not allow the user to complete the process that you have designed.

## Corrected HTML - EXAMPLE 1

In the corrected HTML color is no longer the only way to determine the correct action.

Thank you for your interest in our product. For those with:</font>

<p><font size="2" face="Verdana">Product A Follow Instructions marked A that are in Red</font></p>

<p><font size="2" face="Verdana">Product B Follow Instructions Marked B that are in Green</font></p>

<p><font color="#FF0000" size="2" face="Verdana">Register your Product AHere</font></p>

<p><font color="#008000" size="2" face="Verdana">Register your Product B Here</font></p>

## ***2. Required form fields***

Example two demonstrates a bad use of color. This example uses colored text to communicate which fields of a form are required for successful submission.

### **Current HTML - EXAMPLE 2**

In the current HTML color is the only way to determine which fields are required to submit the form.

```
<form method="POST" action="--WEBBOT--">
  <p>Required fields are RED</p>
  <p>User Name <input type="text" name="T1"
size="20"></p>
  <p>User Address <input type="text" name="T2"
size="20"></p>
  <p><font color="#FF0000">User E-Mail
Address</font><input type="text" name="T3"
size="20"></p>
  <p><input type="submit" value="Submit"
name="B1"></p>
</form>
```

### **Corrected HTML - EXAMPLE 2**

In the corrected HTML color is not the only way to determine which fields are required to submit the form.

```
<form method="POST" action="--WEBBOT--">
  <p>Required fields are RED and Marked with an
*</p>
  <p>User Name
<input type="text" name="T1" size="20"></p>
  <p>User Address
<input type="text" name="T2" size="20"></p>
```

```
<p><font color="#FF0000">* User E-Mail  
Address</font>  
<input type="text" name="T3" size="20"></p>  
<p>  
<input type="submit" value="Submit" name="B1">  
</p>  
</form>
```

### **Apply what you have learned**

In the above example the completion of the form can be accomplished without any dependence on color. However the form itself is not accessible. Use Alternative text and labels to make the form accessible!

## Section 4 - Section 508 (e) W3C Guidelines 1.2

The demo file for this section can be downloaded from the internet

<http://www.hisoftware.com/uaccess/fpdemo4.htm>

Remember to follow along in your Web page editor you must first download the example. To do this: Open the above page in Internet Explorer and save it locally by completing the following steps.

1. Open the above page in your browser
2. Select File |Save As
3. Type in fpdemo4.htm in the filename input box, make sure you selected to save as complete web page so you will have all demo files.
4. Click on the Save button

*If you purchased the eBook these files were included in a zip file called demos.zip. Extract this file to your hard drive.*

Now open the saved file in your Web page editor and we are ready to begin. The accessibility issue that will be covered by section four is:

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



## ***1. Provide Redundant Text Links for server Side Image Maps***

Example one demonstrates how to use redundant text links to make a server side image map accessible.

### **Current HTML**

```
<A  
href="http://www.hisoftware.com/scripts/imagemap/a  
ccessmap">  
<IMG src="../../images/ACCVerifybut88.gif"  
ismap width="88" height="31"></A>
```

### **Information**

Remember when correcting this image map you need to use alternative text for the image.

### **Corrected HTML**

```
<A  
href="http://www.hisoftware.com/scripts/imagemap/a  
ccessmap">  
<IMG src="../../images/ACCVerifybut88.gif"  
alt="Get AccVerify Approved(Text links follow)"  
ismap width="88" height="31"></A></font>  
<p> [<A href="../../access/vindex.html">  
AccVerify Home</A>]&nbsp;</font></p>  
<p> [<A href="../../access/approved.html">  
The Approved Logo Meaning</A>]</p>
```

## Section 5 - Section 508 (g&h) W3C Guidelines 5.1 & 5.2

The demo file for this section can be downloaded from the internet

<http://www.hisoftware.com/uaccess/fpdemo5.htm>

Remember to follow along in your Web page editor you must first download the example. To do this: Open the above page in Internet Explorer and save it locally by completing the following steps.

1. Open the above page in your browser
2. Select File | Save As
3. Type in fpdemo4.htm in the filename input box, make sure you selected to save as complete web page so you will have all demo files.
4. Click on the Save button

*If you purchased the eBook these files were included in a zip file called demos.zip. Extract this file to your hard drive.*

Now open the saved file in your Web page editor and we are ready to begin. The accessibility issues that will be covered by section five are:

(g) / 5.1

- All Cells in the first row must be column header cells
- All Cells in the first column must be row header cells
- Tables used for data should have a caption

(h) / 5.2

- All column header cells are required to contain the scope attribute
- All row header cells are required to contain the scope attribute
- All column header cells are required to contain the id attribute
- All row header cells are required to contain the id attribute
- All data cells are required to contain the headers attribute
- All data cells are required to contain the axis attribute
- When row grouping elements are used, all rows are required to be grouped

## ***1. Row and Column Headers shall be identified for Data Tables***

Example one demonstrates how to add headers to rows and columns.

### **Current HTML - EXAMPLE 1**

The current HTML example has not identified row and column headers.

```
<table border="1" cellpadding="4" cellspacing="0"
width="100%">
  <tr>
    <td width="25%">Time of day</td>
    <td width="25%">Food</td>
    <td width="25%">Drink</td>
    <td width="25%">Time allotted for meal</td>
  </tr>
  <tr>
    <td width="25%">6 am</td>
    <td width="25%">Eggs</td>
    <td width="25%">Juice</td>
    <td width="25%">30 minutes</td>
  </tr>
  <tr>
    <td width="25%">11am</td>
    <td width="25%">Sandwich</td>
    <td width="25%">Water</td>
    <td width="25%">30 minutes</td>
  </tr>
</TABLE>
```

## Corrected HTML - EXAMPLE 1

To correct the HTML we add Row and Column headers

```
<TABLE cellSpacing=0 cellPadding=4 width="100%"  
border=1>
```

```
<TR>
```

```
<TH width="25%">Time of day</TH>
```

```
<TH width="25%">Food</TH>
```

```
<TH width="25%">Drink</TH>
```

```
<TH width="25%">Time allotted for  
meal</TH></TR>
```

```
<TR>
```

```
<TD width="25%">6 am</TD>
```

```
<TD width="25%">Eggs</TD>
```

```
<TD width="25%">Juice</TD>
```

```
<TD width="25%">30 minutes</TD></TR>
```

```
<TR>
```

```
<TD width="25%">11am</TD>
```

```
<TD width="25%">Sandwich</TD>
```

```
<TD width="25%">Water</TD>
```

```
<TD width="25%">30  
minutes</TD></TR></TABLE>
```

## ***2. All data tables should have a caption***

Example two demonstrates how to add a CAPTION to a data table.

### **Current HTML - EXAMPLE 2**

The current HTML is missing the CAPTION element.

Remember a CAPTION will be visible on the screen.

```
<TABLE cellSpacing=0 cellPadding=4 width="100%"
border=1>
<TR>
  <TH width="25%">Time of day</TH>
  <TH width="25%">Food</TH>
  <TH width="25%">Drink</TH>
  <TH width="25%">Time allotted for
meal</TH></TR>
<TR>
  <TD width="25%">6 am</TD>
  <TD width="25%">Eggs</TD>
  <TD width="25%">Juice</TD>
  <TD width="25%">30 minutes</TD></TR>
<TR>
  <TD width="25%">11am</TD>
  <TD width="25%">Sandwich</TD>
  <TD width="25%">Water</TD>
  <TD width="25%">30 minutes</TD></TR>
<TR>
  <TD width="25%">3pm</TD>
  <TD width="25%">Steak</TD>
  <TD width="25%">Water</TD>
  <TD width="25%">1 Hour</TD></TR></TABLE>
```

## Corrected HTML - EXAMPLE 2

The corrected HTML now has the CAPTION added right below the TABLE element.

```
<TABLE cellSpacing=0 cellPadding=4 width="100%"
border=1>
  <CAPTION>Table about Meals and times allotted
  for eating</CAPTION>

  <TR>
    <TH width="25%">Time of day</TH>
    <TH width="25%">Food</TH>
    <TH width="25%">Drink</TH>
    <TH width="25%">Time allotted for
meal</TH></TR>
  <TR>
    <td width="25%">6 am</td>
    <TD width="25%">Eggs</TD>
    <TD width="25%">Juice</TD>
    <TD width="25%">30 minutes</TD></TR>
  <TR>
    <td width="25%">11am</td>
    <TD width="25%">Sandwich</TD>
    <TD width="25%">Water</TD>
    <TD width="25%">30 minutes</TD></TR>
  <TR>
    <td width="25%">3pm</td>
    <TD width="25%">Steak</TD>
    <TD width="25%">Water</TD>
    <TD width="25%">1 Hour</TD></TR></TABLE>
```

### ***3. All column & row header cells are required to contain the scope attribute***

Example three demonstrates how to add the scope attribute.

#### **Current HTML - EXAMPLE 3**

The current HTML example does not contain the scope attribute.

```
<TABLE cellSpacing=0 cellPadding=4 width="100%"
border=1>
<TR>
  <TD width="25%">Time of day</TD>
  <TD width="25%">Food</TD>
  <TD width="25%">Drink</TD>
  <TD width="25%">Time allotted for
meal</TD></TR>
<TR>
  <TD width="25%">6 am</TD>
  <TD width="25%">Eggs</TD>
  <TD width="25%">Juice</TD>
  <TD width="25%">30 minutes</TD></TR>
<TR>
  <TD width="25%">11am</TD>
  <TD width="25%">Sandwich</TD>
  <TD width="25%">Water</TD>
  <TD width="25%">30
minutes</TD></TR></TABLE>
```



### Corrected HTML - EXAMPLE 3

The HTML is modified and now contains the scope attribute.

```
<TABLE cellSpacing=0 cellPadding=4 width="100%"
border=1>

<TR>
  <TH scope="col" width="25%">Time of day</TH>
  <TH scope="col" width="25%">Food</TH>
  <TH scope="col" width="25%">Drink</TH>
  <TH scope="col" width="25%">Time allotted for
meal</TH></TR>
<TR>
  <TD width="25%">6 am</TD>
  <TD width="25%">Eggs</TD>
  <TD width="25%">Juice</TD>
  <TD width="25%">30 minutes</TD></TR>
<TR>
  <TD width="25%">11am</TD>
  <TD width="25%">Sandwich</TD>
  <TD width="25%">Water</TD>
  <TD width="25%">30
minutes</TD></TR></TABLE>
```

#### ***4. All DATA cells are required to contain the header attribute if they are HEADER cells***

Example four demonstrates the correct use of the header attribute.

##### **Current HTML - EXAMPLE 4**

The current HTML example does not contain header attributes.

```
<TABLE cellSpacing=0 cellPadding=4 width="100%"
border=1>
<TR>
  <TD width="25%">Time of day</TD>
  <TD width="25%">Food</TD>
  <TD width="25%">Drink</TD>
  <TD width="25%">Time allotted for
meal</TD></TR>
<TR>
  <TD width="25%">6 am</TD>
  <TD width="25%">Eggs</TD>
  <TD width="25%">Juice</TD>
  <TD width="25%">30 minutes</TD></TR>
<TR>
  <TD width="25%">11am</TD>
  <TD width="25%">Sandwich</TD>
  <TD width="25%">Water</TD>
  <TD width="25%">30
minutes</TD></TR></TABLE>
```

#### Corrected HTML - EXAMPLE 4

The corrected HTML has the required header attributes.

```
<TABLE cellSpacing=0 cellPadding=4 width="100%"
border=1>
<TR>
  <TH id="header5" width="25%">Time of
day</TH>
  <TH id="header6" width="25%">Food</TH>
  <TH id="header7" width="25%">Drink</TH>
  <TH id="header8" width="25%">Time allotted for
meal</TH></TR>
<TR>
  <TD headers="header5" width="25%">6 am</TD>
  <TD headers="header6"
width="25%">Eggs</TD>
  <TD headers="header7"
width="25%">Juice</TD>
  <TD headers="header8" width="25%">30
minutes</TD></TR>
<TR>
  <TD headers="header5"
width="25%">11am</TD>
  <TD headers="header6"
width="25%">Sandwich</TD>
  <TD headers="header7"
width="25%">Water</TD>
  <TD headers="header8" width="25%">30
minutes</TD></TR></TABLE>
```

## ***5. All DATA cells are required to contain the AXIS attribute***

Example five demonstrates the correct use of the axis attribute.

### **Current HTML - EXAMPLE 5**

The current HTML example does not contain axis attributes.

```
<TABLE cellSpacing=0 cellPadding=4 width="100%"  
border=1>  
<TR>  
  <TD width="25%">Time of day</TD>  
  <TD width="25%">Food</TD>  
  <TD width="25%">Drink</TD>  
  <TD width="25%">Time allotted for  
meal</TD></TR>  
<TR>  
  <TD width="25%">6 am</TD>  
  <TD width="25%">Eggs</TD>  
  <TD width="25%">Juice</TD>  
  <TD width="25%">30 minutes</TD></TR>  
<TR>  
  <TD width="25%">11am</TD>  
  <TD width="25%">Sandwich</TD>  
  <TD width="25%">Water</TD>  
  <TD width="25%">30  
minutes</TD></TR></TABLE>
```

## Corrected HTML - EXAMPLE 5

The corrected HTML example contains the axis attributes.

```
<TABLE cellSpacing=0 cellPadding=4 width="100%"
border=1>

<TR>
  <TH id="header9" width="25%">Time of
day</TH>
  <TH id="header10" width="25%">Food</TH>
  <TH id="header11" width="25%">Drink</TH>
  <TH id="header12" width="25%">Time allotted for
meal</TH></TR>
<TR>
  <TD axis="header9" width="25%">6 am</TD>
  <TD axis="header10" width="25%">Eggs</TD>
  <TD axis="header11" width="25%">Juice</TD>
  <TD axis="header12" width="25%">30
minutes</TD></TR>
<TR>
  <TD axis="header9" width="25%">11am</TD>
  <TD axis="header10"
width="25%">Sandwich</TD>
  <TD axis="header11" width="25%">Water</TD>
  <TD axis="header12" width="25%">30
minutes</TD></TR></TABLE>
```

***6. When row grouping elements are used, all rows are required to be grouped***

Example six demonstrates the correct use of row grouping.

**Current HTML - EXAMPLE 6**

The current HTML example does not use row grouping

```
<TABLE cellSpacing=0 cellPadding=4 width="100%"
border=1>
<TR>
  <TD width="25%">Time of day</TD>
  <TD width="25%">Food</TD>
  <TD width="25%">Drink</TD>
  <TD width="25%">Time allotted for
meal</TD></TR>
<TR>
  <TD width="25%">6 am</TD>
  <TD width="25%">Eggs</TD>
  <TD width="25%">Juice</TD>
  <TD width="25%">30 minutes</TD></TR>
<TR>
  <TD width="25%">11am</TD>
  <TD width="25%">Sandwich</TD>
  <TD width="25%">Water</TD>
  <TD width="25%">30
minutes</TD></TR></TABLE>
```

## Corrected HTML - EXAMPLE 6

The corrected HTML example uses row grouping as required.

```
<TABLE cellSpacing=0 cellPadding=4 width="100%"  
border=1>  
<thead>  
<TR>  
  <TH width="25%">Time of day</TH>  
  <TH width="25%">Food</TH>  
  <TH width="25%">Drink</TH>  
  <TH width="25%">Time allotted for  
meal</TH></TR>  
</thead>  
<TBODY>  
<TR>  
  <TD width="25%">6 am</TD>  
  <TD width="25%">Eggs</TD>  
  <TD width="25%">Juice</TD>  
  <TD width="25%">30 minutes</TD></TR>  
<TR>  
  <TD width="25%">11am</TD>  
  <TD width="25%">Sandwich</TD>  
  <TD width="25%">Water</TD>  
  <TD width="25%">30 minutes</TD></TR>  
</TBODY>  
</TABLE>
```

## Section 6 - Section 508 (i) W3C Guidelines 1.1 & 12.1

The demo file for this section can be downloaded from the internet

Initial Example:

<http://www.hisoftware.com/uaccess/frames/FPDemoFrames.htm>

Fixed Example:

<http://www.hisoftware.com/uaccess/frames/FPDemoFramesFixed.htm>

Remember to follow along in your Web page editor you must first download the example. To do this: Open the above page in Internet Explorer and save it locally by completing the following steps.

1. Open the above page in your browser
2. Select File | Save As
3. Type in fpdemoframes.htm or fpdemoframesfixed.htm in the filename input box, make sure you selected to save as complete web page so you will have all demo files.
4. Click on the Save button

*If you purchased the eBook these files were included in a zip file called demos.zip. Extract this file to your hard drive.*

Now open the saved file in your Web page editor and we are ready to begin.



The accessibility issues that will be covered by section six are:

- All Frame Elements are required to contain the title attribute
- All Frameset Elements are required to contain the noframes element

## ***1. All Frame Elements are required to contain the title attribute***

Example one demonstrates how to use FRAME title attributes.

### **Current HTML - EXAMPLE 1**

The current HTML does not implement the title attribute

```
<FRAMESET  
cols="40%,60%">  
<FRAME src="left_frame.htm">  
<FRAME src="right_frame.htm" >  
</FRAMESET>
```

### **Corrected HTML - EXAMPLE 1**

The corrected HTML implements the title attribute

```
<FRAMESET  
cols="40%,60%"  
title="Our product line">  
<FRAME src="left_frame.htm"  
title="Frame Left:  
Navigating our web">  
<FRAME src="right_frame.htm"  
title="Frame Body: Body text">  
</FRAMESET>
```

## ***2. All Frameset Elements are required to contain the noframes element***

Example two demonstrates how to use the NOFRAMES element.

### **Current HTML - EXAMPLE 2**

The current HTML does not implement the NOFRAMES element.

```
<FRAMESET
cols="40%,60%"
title="Our product line">
<FRAME src="left_frame.htm">
<FRAME src="right_frame.htm" >
</FRAMESET>
```

### **Corrected HTML - EXAMPLE 2**

The corrected HTML implements the NOFRAMES element.

```
<FRAMESET
cols="40%,60%"
title="Our product line">
<FRAME src="left_frame.htm">
<FRAME src="right_frame.htm" >
<NOFRAMES>
  This page uses frames, but your
  Search engine or Web browser may not
  support them.
  <A href="products.html" title="Products link">
    Select here to go to the
    products page</A>
</NOFRAMES>
</FRAMESET>
```

## HTML With NOFRAMES Element Added and Title Attribute

The below corrected HTML makes the Frameset completely accessible.

```
<FRAMESET
cols="40%,60%"
title="Our product line">
<FRAME src="left_frame.htm"
title="Frame Left:
Navigating our web">
<FRAME src="right_frame.htm"
title="Frame Body: Body text">
<NOFRAMES>
This page uses frames, but your
Search engine or Web browser may not
support them.
<A href="products.html"
title="Products link">
    Select here to go to the
    products page</A>
</NOFRAMES>
</FRAMESET>
```

## Section 7 - Section 508 (j) W3C Guidelines 7.1

The demo file for this section can be downloaded from the internet

<http://www.hisoftware.com/uaccess/FPDemo6.htm>

Remember to follow along in your Web page editor you must first download the example. To do this: Open the above page in Internet Explorer and save it locally by completing the following steps.

1. Open the above page in your browser
2. Select File |Save As
3. Type in fpdemo6.htm in the filename input box, make sure you selected to save as complete web page so you will have all demo files.
4. Click on the Save button

*If you purchased the eBook these files were included in a zip file called demos.zip. Extract this file to your hard drive.*

Now open the saved file in your Web page editor and we are ready to begin. This section covers the most common forms of screen flickering. If you want more information on this requirement refer to Chapter Two of this book. The accessibility issues that will be covered by section seven are:

- Documents are required not to contain the BLINK element
- Documents are required not to contain the MARQUE element

## ***1. Do not use the BLINK Element***

Example one demonstrates alternatives to the BLINK element.

### **Current HTML - EXAMPLE 1**

The current HTML uses the BLINK element to allow the user agent to blink (flash the Message) repeatedly signifying the importance of the text to the user.

```
<blink>Pay attention to this</blink>
```

### **Corrected HTML - EXAMPLE 1**

The corrected HTML does not use the BLINK element instead it implements other formatting to show the importance of the text.

```
<strong><u>Pay Attention to  
This</u></strong>
```

## ***2. Do not use the MARQUEE element***

Example two demonstrates alternatives for the MARQUEE element.

### **Current HTML - EXAMPLE 2**

The current HTML uses the MARQUEE element to scroll a message across the screen; this is intended to catch the users attention.

```
<marquee align="bottom" bgcolor="#C0C0C0"
direction="right" scrolldelay="91" width="100%"
height="16">
use
of Marquee can cause you to fail Accessibility testing
</marquee>
```

### **Corrected HTML - EXAMPLE 2**

The corrected HTML uses alternative methods of making the text grab the users attention.

```
<em><strong><u>
use of Marquee can cause you  to fail Accessibility
Testing
</u></strong></em>
```

## Section 8 - Section 508 (l)&(m) W3C Guidelines

### 6.3

The demo file for this section can be downloaded from the internet

<http://www.hisoftware.com/uaccess/FPDemo7.htm>

Remember to follow along in your Web page editor you must first download the example. To do this: Open the above page in Internet Explorer and save it locally by completing the following steps.

1. Open the above page in your browser
2. Select File |Save As
3. Type in fpdemo7.htm in the filename input box, make sure you selected to save as complete web page so you will have all demo files.
4. Click on the Save button

*If you purchased the eBook these files were included in a zip file called demos.zip. Extract this file to your hard drive.*

Now open the saved file in your Web page editor and we are ready to begin. The accessibility issues that will be covered by section eight are:

Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.



- Anchor elements are required not to use javascript for the link target when the NOSCRIPT element is not present in the document body.
- AREA elements are required not to use javascript for the link target when the NOSCRIPT element is not present in the document body.
- When SCRIPT Elements are used, the NOSCRIPT element is required in the document body.

## ***1. If you use the SCRIPT element use the NOSCRIPT element***

Example one demonstrates the proper usage of the NOSCRIPT element

### **Current HTML - EXAMPLE 1**

The current HTML uses the script element that can be used to open another browser window. This is used to present the user with an offer.

```
<SCRIPT language="JavaScript">
<!-- function popwin(){
window.open('http://www.hisoftware.com/acctest.htm',
'', 'width=400,height=200, scrollbars=1');
} // --> </script>
```

### **Corrected HTML - EXAMPLE 1**

The corrected HTML uses the NOSCRIPT element for the SCRIPT element. This assures that users who do not have a browser that supports scripts receive the same offer.

```
<SCRIPT language="JavaScript">
<!-- function popwin(){
window.open('http://www.hisoftware.com/acctest.htm',
'', 'width=400,height=200, scrollbars=1');
} // --> </script>
<noscript> This pages used JavaScript to display a
pop-up window that points a user to <a
href="http://trial.accessibilitywatch.com/scripts/tryacc
wmh.exe" class="thenav">WebSite
Accessibility Testing and Repair Solutions trial</a>
This link is for a trial of the web site accessibility
testing service</noscript>
```

## ***2. Do not use script in ANCHOR elements***

Example two demonstrates how to correct ANCHOR elements that use script.

### **Current HTML - EXAMPLE 2**

The current HTML uses a JavaScript function to navigate within an ANCHOR element. This is unwise and should not be done.

```
<a  
  href="javascript:displayWindow('sample.htm',360,14  
  5)">
```

### **Corrected HTML - EXAMPLE 2**

The corrected HTML uses a direct link to the page.

```
<a href="http://www.hisoftware.com/sample.htm">
```

### **NOTES - EXAMPLE 2**

Using JavaScript in your anchors could work for most visitors, however, the complexity of the NOSCRIPT element should deter you from doing this. It is recommended to NOT USE JavaScript in ANCHOR elements.

### ***3. Do not use script in AREA elements***

Example three demonstrates how to correct AREA elements that use script.

#### **Current HTML - EXAMPLE 3**

The current HTML uses a JavaScript functions to for links in AREA elements. This is unwise and should not be done.

```
<map name="FMap0">
  <area
href="javascript:displayWindow('hit.htm',360,145)"
shape="rect" coords="61, 24, 132, 92">
  <area
href="javascript:displayWindow('miss.htm',360,145)"
shape="rect" coords="16, 101, 179, 215"></map>
</p>
```

#### **Corrected HTML - EXAMPLE 3**

The corrected HTML uses a direct link to the page.

```
<map name="FMap1">
  <area href="hit.htm" coords="61, 24, 132, 92"
shape="rect">
  <area href="miss.htm" coords="16, 101, 179, 215"
shape="rect"></map></p>
```

## Section 9 - Section 508 (0)

The demo file for this section can be downloaded from the internet

<http://www.hisoftware.com/uaccess/FPDemo8.htm>

Remember to follow along in your Web page editor you must first download the example. To do this: Open the above page in Internet Explorer and save it locally by completing the following steps.

1. Open the above page in your browser
2. Select File |Save As
3. Type in fpdemo8.htm in the filename input box, make sure you selected to save as complete web page so you will have all demo files.
4. Click on the Save button

*If you purchased the eBook these files were included in a zip file called demos.zip. Extract this file to your hard drive.*

Now open the saved file in your Web page editor and we are ready to begin. The accessibility issue that will be covered by section nine are:

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

## ***1. Skip Navigation Links***

Example one demonstrates how to use a HYPERLINK to a BOOKMARK on your page to skip navigation links.

### **Current HTML - EXAMPLE 1**

The current HTML has a set of navigation links and main content. There is no method to skip the navigation links to gain direct access to the content.

```
<p><a href="http://www.hisoftware.com/">Home</a>
| <a
href="http://www.hisoftware.com/press/">News</a> |
<a
href="http://www.hisoftware.com/support/">Support<
/a></p>
<p>This is the pages main content... Section 508
(o) States that there needs to be a method to skip
the navigation links above
And allow the user to get directly to this content!
</p>
```

### **Corrected HTML - EXAMPLE 1**

The corrected HTML has a set of navigation links and main content. The page satisfies paragraph (o) of the Section 508 Standards by creating a BOOKMARK where the content starts and then by placing a Hyperlink to go to this bookmark before any of the navigation links are encountered.

```
<p><a href="#test">Skip Navigation Links</a> | <a
href="http://www.hisoftware.com/">Home</a>
```

```
| <a  
href="http://www.hisoftware.com/press/">News</a> |  
<a href="http://www.hisoftware.com/support/">  
Support</a></p>  
<p>  
<a name="test"></a>
```

This is the pages main content... Section 508 (o) states that there needs to be a method to skip the Navigation links above and allows the user to get directly to this content!





# 8. Use Accessible Solutions

---

## *In this chapter*

- Discussion on how software accessibility standards relate to Web standards
  - Introduction to the Section 508 and the Voluntary Product Accessibility Template
  - Software and features that support accessibility
  - Remarks and explanations for software compliance under Section 508
  - Introduction to EITACC Desktop Software Guidelines
  - Introduction to Microsoft Accessibility Guidelines
- 

## **Software Standards**

Software standards for plug-ins or components are as important as your Web site accessibility. This is true because:

- Many Web sites use multimedia objects and presentations that require third party plug-ins
- ActiveX controls are common
- Java Applets have become common
- Web sites are being used as documentation repositories

The Section 508 guideline clearly states that Web pages requiring an applet or plug-in on the client system must make sure that the applet or plug-in itself is accessible. In many cases the required plug-in or object is automatically downloaded to the client. If not a link must be provided as to where the user can obtain the plug-in. This is stated in paragraph (m) of the Section 508 standard.

This presents a problem in the development of dynamic Web sites. Simply put, if you present the user with information that requires a separate component to run or view information, then that client program or component must be accessible. Many components or custom developed Web based applications are not.

Some examples:

- An older PDF document would not be accessible in the Acrobat 5.0 viewer. The document does not have the correct properties that allow assistive technologies to get at the content.
- A new PDF Document created with Adobe Acrobat 5.0 can be utilized by assistive technologies, however, if you have an older viewer it cannot take advantage of the new formatting.
- A Custom Object is developed to show a companies schedule of events. The custom object itself is not accessible. If this information was not available in any other way then the site would not be accessible because the object was not accessible.

The important concept for a Web site developer to remember is:

*The Web site, its content, and all third party or custom components must be accessible*

The rest of this chapter will introduce you to the software accessibility standards.

## **Section 508**

When you are selecting software solutions to implement accessibility to your site you should consider the accessibility of the software itself, as well as its output. (This is a requirement for US Federal agencies under Section 508.)

### ***The Voluntary Product Accessibility Template***

The IT Industry Council has developed a form called a Voluntary Product Accessibility Template (VPAT.) This form is a self-certification that allows software vendors to identify the accessibility components of their software applications, and to provide that information to their customers. Software companies should be able to provide you with their voluntary product accessibility statement. This will give you a good idea whether or not the software is accessible. Remember it is your responsibility to verify that the solution is accessible.

The following items should be reviewed when selecting a tool:

- (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. It must also move among interactive interface elements as the input focus changes. The focus shall be programmatically exposed so that assistive technologies 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 technologies. 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.

### ***Software Output***

The above sections deals with software characteristics, however many software applications generate on-screen output. Remember that all output generated, whether it is HTML, XHTML, PDF, or any other format, must be accessible.

## ***Supporting Features***

If necessary the software solution VPAT should list the supporting features of an accessibility checkpoint.

### **Criteria**

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.

## **SUPPORTING FEATURES**

All features and functions of the software have text equivalents.

In the above example the software developer noted that all features and functions have text equivalents as required by (a). If a software developer delivers a VPAT that is not complete you may want the company to validate in writing the support for these essential accessibility features.

## ***Remarks and Explanations***

If necessary the software solution VPAT should list additional remarks or explanations required to describe the accessibility of their product.

### ***Criteria***

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.

## **Remarks and explanations**

The functions are accessible to the user through the toolbar menu items and access keys. HiSoftware uses standard Windows® access keys, as well as additional access keys that are included in AccMonitor documentation.

In the above example the Software developer fully lists the extent of compliance to the accessibility feature required.

## **Introduction to EITACC Desktop Software Standards**

Complete Information on this standard can be found at:

[http://trace.wisc.edu/docs/eitaac\\_desktop\\_software\\_standards/desktop\\_software\\_standards.htm](http://trace.wisc.edu/docs/eitaac_desktop_software_standards/desktop_software_standards.htm)

The Standard covers:

- Keyboard access
- Object Information
- Icons
- Sounds
- Display
- System Startup and Shutdown

Developers will find the type of information that they require to comply with this standard clearly detailed on the site listed above.

## Introduction to Microsoft Accessibility Guidelines

The complete Microsoft guidelines for designing accessible software applications can be found at:

<http://www.msdn.microsoft.com/library/default.asp?url=/library/en-us/dnacc/html/accessibilitydsgn.asp>

This is perhaps one of the most complete guide available on the internet. In addition to saying what needs to be covered with Accessible software design Microsoft also:

- Rates the important
- Lists the impacted disabilities
- Provides a firm foundation for those testing software

This is a must read for software developers!



# 9. Accessibility Resources for Further Reading

This section of the book will introduce the reader to additional resources available pertaining to accessibility. Resources will be divided by topic:

- Section 508
- Related Web Resources
- W3C/WAI
- Software Solutions
- Accessibility Services

## Section 508

*Architectural and Transportation Barriers Compliance Board  
Electronic Information Technology Accessibility Standards; Final  
Rule*

<http://www.access-board.gov/sec508/508standards.htm>

*Federal IT Accessibility Initiative:*

<http://www.section508.gov>

Center for IT Accommodation

[http://www.gsa.gov/Portal/content/offerings\\_content.jsp?contentOID=22804&contentType=1004&PMKC=1&S=1](http://www.gsa.gov/Portal/content/offerings_content.jsp?contentOID=22804&contentType=1004&PMKC=1&S=1)

Department of Justice Section 508 Web Site

<http://www.usdoj.gov/crt/508/508home.html>

Microsoft and Section 508 Web Site

<http://www.microsoft.com/enable/microsoft/section508.htm>

## Related Web Resources

*Trace Research & Development Center:*

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

NCAM - Media Access Equality

<http://ncam.wgbh.org/>

Microsoft - Accessibility Home

<http://www.microsoft.com/enable/>

IBM Accessibility Center

<http://www-3.ibm.com/able/index.html>

*Software Accessibility Guidelines*

<http://www-3.ibm.com/able/accesssoftware.html>

EITACC Desktop Software Standards

[http://trace.wisc.edu/docs/eitaac\\_desktop\\_software\\_standards/desktop\\_software\\_standards.htm](http://trace.wisc.edu/docs/eitaac_desktop_software_standards/desktop_software_standards.htm)

Microsoft Accessible Software Guidelines

<http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000544>

Microsoft FrontPage Accessibility Page

<http://www.microsoft.com/frontpage/using/accessibility/default.htm>

## W3C / WAI

*Web Content Accessibility Guidelines 1.0*

<http://www.w3.org/TR/WCAG/>

*Techniques for Web Content Accessibility Guidelines 1.0*

<http://www.w3.org/TR/WCAG10-TECHS/>

*Core Techniques for Web Content Accessibility Guidelines 1.0*

<http://www.w3.org/TR/WCAG10-CORE-TECHS/>

*HTML Techniques for Web Content Accessibility Guidelines 1.0*

<http://www.w3.org/TR/WCAG10-HTML-TECHS/>

*CSS Techniques for Web Content Accessibility Guidelines 1.0*

<http://www.w3.org/TR/WCAG10-CSS-TECHS/>

## Software Solutions

AccVerify SE For Microsoft FrontPage - a free accessibility checker for Microsoft FrontPage

<http://www.hisoftware.com/msacc/index.html>

*HiSoftware Accessibility Solutions Home*  
<http://www.hisoftware.com/access/Index.html>

Microsoft Reader Software - eBook Reader  
<http://www.microsoft.com/reader/default.asp>

Adobe Acrobat eBook Reader  
<http://www.adobe.com/products/ebookreader/register.html>

## **Accessibility Services**

*HiSoftware Accessibility Services Home*  
<http://www.hisoftware.com/services/services.htm>

# Appendix A

## Section 508 standards and W3C® Priority 1 checkpoints

When you consider accessibility criteria from different organizations, be sure to compare the criteria to the standards you need to meet. The two most common sets of accessibility criteria are *Electronic Information Technology Accessibility Standards; Final Rule*, which is issued by the U.S. Access Board, and *Web Content Accessibility Guidelines 1.0*, which has been published by W3C®. Both organizations have published accessibility criteria that are considered to be essential to accessible Web content; however, their criteria are not the same, yet they do overlap in some cases.

This section provides you with W3C<sup>®</sup> checkpoint equivalents for Section 508 standards.

508:

A. A text equivalent<sup>6</sup> for every non-text element shall be provided.

W3C:

1.1 Priority 1<sup>7</sup>

508:

B. Equivalent alternatives<sup>1</sup> to multimedia presentations shall be synchronized with the audio presentation.

W3C:

1.4 Priority 1

508:

C. Color shall not be used as a single method for indicating important information on a Web page.

W3C:

2.1 Priority 1

---

<sup>6</sup> An equivalent element or page conveys all essential information of the original.

<sup>7</sup> The W3C issues the following statement about Priority 1 checkpoints: "A Web content developer *must* satisfy this checkpoint. Otherwise, one or more groups will find it impossible to access information in the document. Satisfying this checkpoint is a basic requirement for some groups to be able to use Web documents.

508:

D. Documents shall be readable without browser support for style sheets<sup>8</sup>.

W3C:

6.1 Priority 1

508:

E. Redundant text links shall be provided for each active region of a server-side image map<sup>9</sup> on a web page.

W3C:

1.2 Priority 1

508:

F. Client-side image maps<sup>10</sup> shall be used instead of server-side image maps, except where regions cannot be defined with an available geometric shape..

W3C:

9.1 Priority 1

---

<sup>8</sup> A style sheet is the default template for one or more pages. The style sheet includes information about the appearance and layout of a page, and is often applied to a group of pages to attain consistency.

<sup>9</sup> A server-side image map requires a script, or set of instructions, from the server to display the active regions of the image map.

<sup>10</sup> A client-side image map displays the active regions of an image map without a script, or set of instructions, from the server.

508:

G & H. Tables shall be created according to the development rules for the markup language. When tables are coded incorrectly, or when table code describes non-tabular material, assistive technologies cannot read the content accurately.

W3C:

5.1 and 5.2 Priority 1

508:

I. Frames<sup>11</sup> shall be titled with text to assist the user in navigating them.

W3C:

12.1 Priority 1

508:

J. Until users can control screen flickering, Web pages shall avoid causing the screen to flicker. (In accordance with 1194.21<sup>12</sup>, screen items shall not flash frequencies between 2Hz and 55Hz.).

W3C:

7.1 Priority 1

---

<sup>11</sup> A frame is part of a Web page that can be controlled independently. For example, a Web page might include frames, one containing links and another containing the unique text that appears on the page.

<sup>12</sup> Section 1194.21 governs software applications and operating systems.



508:

K. A text-only alternative page<sup>13</sup> shall be provided only as a last resort in complying with accessibility standards.

W3C:

11.4 Priority 1

508:

L. When a Web relies on scripts<sup>14</sup> to display information or process user input, functional text shall be provided.

W3C:

6.3 Priority 1 (in part)

508:

M. When a Web page contains proprietary file formats that require a plug-in for display, the Web page shall provide a link to a plug-in that meets software accessibility provisions outlined in 1194.21.

W3C:

Does not exist as a W3C<sup>®</sup> checkpoint

---

13 A Failsafe measure allows the Web page creator to provide a link to a text-only version of a page when it is not possible to make a Web page accessible.

14 A script is a set of directions written in one program to be interpreted by another program. For example, a script can provide your browser with a sequence of instructions on how to display a Web page with animations and sounds.

508:

N. Electronic forms<sup>15</sup> shall be made accessible to people with disabilities.

W3C:

Does not exist as a W3C<sup>®</sup> checkpoint

508:

O. A method shall be used on Web pages to allow for easy tracking of page content. This method shall provide users of assistive technology the option to skip repetitive navigation links.

W3C:

Does not exist as a W3C<sup>®</sup> checkpoint

508:

P. When a script causes a time-out<sup>16</sup> on pages with forms, the user shall be given the option to indicate that additional time is necessary to complete the form.

W3C:

Does not exist as a W3C<sup>®</sup> checkpoint

---

<sup>15</sup> A form processes input from the user. For example, when you sign up for a Web-based e-mail account, you fill out an electronic form.

<sup>16</sup> A time-out is a dropped connection between the user host and the server that delivers page information.

# Appendix B

## Summary of Section 508 standards for software applications and operating systems

The following standards have been interpreted from the document, *Electronic Information Technology Accessibility Standards; Final Rule*. Software standards have been included to provide you with a general idea of their content. The following standards should be read in their entirety and interpreted, just as you would with any law requiring compliance on a technical level.

- A. When software is created for systems with keyboards, all functions (or their results) that are identifiable by text shall be accessible by the keyboard.
- B. Applications shall not disrupt or disable any accessibility features that are developed according to industry standards.
- C. Applications shall use a visual indicator, or focus, to indicate where action occurs upon a keystroke or mouse click. The indicator shall be readable by other applications.
- D. The controls of an application shall be readable by assistive technologies. This is accomplished through the program code. Examples of controls include checkboxes, menus, and toolbars.

- E. When images are used as icons, the image shall not change in meaning throughout the course of the application. (Screen readers allows users to name icons based on their function or meaning.)
- F. Textual information in an application shall be provided using the functions of the operating system. This ensures that the text be compatible with assistive technologies.
- G. Applications shall not override display attributes, such as color and contrast settings, that are selected by the user.
- H. Since some assistive technologies do not support animation, a software application or operating system shall supply animated elements additionally in non-animated form.
- I. Color shall not be used as a primary method of conveying information. Color is permitted as an identifier; however, an alternative method of labeling the information shall be provided.
- J. When users of software applications and operating systems have the opportunity to select colors, they shall also be provided the ability to modify the contrast settings.
- K. Software applications and operating systems shall not cause screen items to flash or blink frequencies between 2 and 55 Hz.
- L. Electronic forms shall be made accessible to people with disabilities. This includes allowing the user to request additional time.

# Glossary

**alt text**

Alternative text, sometimes called alt text, is a method of providing equivalent text for non-text elements, such as images, objects, and applets. W3C<sup>®</sup> has recommended that alt text be no longer than 150 characters.

**applet**

An applet is a little application. An applet can accompany a Web page, so that tasks can be performed without having to send a request back to the server.

**ASCII art**

ASCII art is an image created using text characters and symbols. An example of this is a smiley face, :-).

**Assistive technologies**

Assistive technologies describe methods that enhance the way users access Web content. These methods can include:

- screen readers that read text aloud or print a Braille version of the text
- screen magnifiers
- audio browsers that can interpret HTML attributes and styles to read the text aloud with inflection
- text-only browsers that display only text characters
- voice input software

**client-side image map**

A client-side image map displays the active regions of an image map without a script, or set of instructions, from the server. The client-side image map is preferable to the server-side image map for accessibility.

**crawler**

A crawler visits documents and their contents to gather information.

**dynamic content**

The dynamic content of a Web page is the information that allows a page to be responsive to user actions. For example, when the user points to text, it changes color.

**element content**

The element content is the information that is available to the Web user when the Web element is not supported by the user agent or browser.

**equivalent**

Something is equivalent when it conveys the same information as the original item. Accessibility standards have equivalent text, and sometimes, equivalent pages.

**frame**

A frame is part of a Web page that can be controlled independently. For example, a Web page might include frames, one containing links and another containing the unique text that appears on the page.

**image map**

An image map is a graphic that contains links.

**longdesc**

A long description, created by using the longdesc attribute in HTML, is a method of providing equivalent text for non-text elements, such as images, objects, and applets. The text value for a long description is the URL where the long description is available. Because some browsers do not support the longdesc attribute, you should use both alternative text and long descriptions for Web elements that require a long description.

**script**

A script is a set of directions written in one program to be interpreted by another program. For example, a script can provide your browser with a sequence of instructions on how to display a Web page with animations and sounds.

**server-side image map**

A server-side image map requires a script, or set of instructions, from the server to display the active regions of the image map. The client-side image map is preferred for accessibility.

**style sheet**

A style sheet is the default template for one or more pages. The style sheet includes information about the appearance and layout of a page, and is often applied to a group of pages to attain consistency.





# Index

Access Board .....	9
contacting .....	14
accessibility	
definition .....	2
future .....	7
software .....	12
<b>alt text, definition</b> .....	186
<b>applet, definition</b> .....	186
<b>ASCII art, definition</b> .....	186
assistive technologies	
definition and examples .....	2
<b>assistive technologies, definition</b> .....	186
<b>client-side image map, definition</b> .....	187
Copyright .....	ii
<b>crawler, definition</b> .....	187
<i>d-links</i>	
<i>equivalent text</i> .....	119
<b>dynamic content, definition</b> .....	187
<b>element content, definition</b> .....	187
<b>equivalent, definition</b> .....	187
<b>frame, definition</b> .....	187
<b>image map, definition</b> .....	187
<b>longdesc, definition</b> .....	188

Rehabilitation Act Amendments of 1998.....	7
Rehabilitation Act of 1973.....	7
<b>script, definition .....</b>	<b>188</b>
section 508	
effective date .....	11
obtaining a copy .....	10
software and operating systems.....	185
subpart B overview.....	9
subparts A, B, C, D.....	8
technical support .....	11
<b>server-side image map, definition.....</b>	<b>188</b>
software .....	12
<b>style sheet, definition .....</b>	<b>188</b>
Workforce Investment Act of 1998.....	7
writing text equivalents .....	104

# Register your book

This book is published in print form and as an eBook. The book covers constantly changing requirements. Because of this if and when changes are delivered to this edition you can automatically receive eBook updates.

In addition to receiving updates you can optionally request to be informed about any new editions on this product. You can register your book at the HiSoftware Corporate web site.

<http://www.hisoftware.com/register.htm>

This is an Accessible Electronic form.