SECTION 508

Compliance and Verification

Version 2.0
June 2018

GovCIO

CONTENTS

Introduction 5

Section 508: A Brief History 6

Section 508 2.0 Overview 6

GovCIO Accessibility Capabilities 8

Background 8

Understanding 8

Section 508 Compliance Management 8

Section 508 Compliance Training 9

Section 508 Compliance Design and Development 9

Support Documentation Section 508 Compliance (E208/602) 10

Support Services Section 508 Compliance (E208/603) 10

Authoring Tools Section 508 Compliance (504) 10

Section 508 Compliance Testing 11

Conformance Levels 13

VPAT (Voluntary Product Accessibility Template) 14

Conformance to Section 508 Standards 15

Level A Compliance 15

Level AA Compliance 19

Functional Performance Criteria 22

Criteria List 22

Reporting Conformance to Functional Performance Criteria Using the VPAT 23

Interoperability with Assistive Technologies (502) 24

Platform Accessibility Features 26

Accssibility Services 26

Platform Accssibility Features 27

Reporting Software compatibility with Assistive Technologies Using the VPAT 27

Documentation and Support Services 30

Support Documentation (602) 30

Support Services (603) 30

Reporting Conformance to Support Documentation and SERVICES Criteria on the VPAT 31

Testing for Section 508 Compliance 32

Automated Testing Tools 32

Testing for Interoperability with Assistive Technologies 34

Testing of Techniques 34

Understanding and Meeting Level A and AA Standards 36

1. Perceivable 36

1.1 Text Alternatives 36

1.1.1 Non-text Content 36

1.2 Time-based Media 45

1.2.1 Audio-only and Video-only (Prerecorded) 45

1.2.2 Captions (Prerecorded) 47

1.2.3 Audio Description or Media Alternative (Prerecorded) 49

1.2.4 Captions (Live) 51

1.2.5 Audio Description (Prerecorded) 52

1.3 Adaptable 54

1.3.1 Info and Relationships 54

1.3.2 MEANINGFUL SEQUENCE 57

1.3.3 SENSORY CHARACTERISTICS 59

1.4 Distinguishable 61

Make it easier for users to see and hear content including separating foreground from background. 61

1.4.1 Use of Color 61

1.4.2 Audio Control 63

1.4.3 Contrast (Minimum) 64

1.4.4 Resize Text 69

1.4.5 Images of Text 72

2. Operable 75

2.1 – Keyboard Accessible 75

2.1.1 Keyboard 75

2.1.2 – No Keyboard Trap 77

2.2 – Enough Time 79

2.2.1 – Timing Adjustable 79

2.2.2 – Pause, Stop, Hide 82

2.3 – Seizures and Physical Reactions 86

2.3.1 – Three Flashes or Below Threshold 86

2.4 – Navigable 88

2.4.1 – Bypass Blocks 88

2.4.2 – Page Titled 90

2.4.3 – Focus Order 91

2.4.4 – Link Purpose 94

2.4.5 – Multiple Ways 96

2.4.6 – Headings and Labels 98

2.4.7 – Focus Visible 99

3. Understandable 101

3.1 Readable 101

3.1.1 – Language of Page 101

3.1.2 – Language of Parts 102

3.2 Predictable 105

3.2.1 – On Focus 105

3.2.2 – On Input 106

3.2.3 – Consistent Navigation 107

3.2.4 – Consistent Identification 109

3.3 Input Assistance 112

3.3.1 – Error Identification 112

3.3.2 – Labels or Instructions 114

3.3.3 – Error Suggestion 116

3.3.4 – Error Prevention (Legal, Financial, Data) 118

4. Robust 121

4.1 Compatible 121

4.1.1 – Parsing 121

4.1.2 – Name, Role, Value 122

INTRODUCTION

As of 2018, Section 508 requires conformance to WCAG 2.0 standards. WCAG 2.0 guidance is made up of 4 principals, decomposed into 12 guidelines, and 61 success criteria (SC). The SC are the requirements and they are assigned levels: A, AA, and AAA. Only the 41 Level A and AA SC are required for Section 508 compliance. All WCAG 2.0 principals, guidelines, and SC are listed below for reference. The level AAA standards are shaded in gray.

1. Perceivable

- 1.1 - Text Alternatives
  - 1.1.1 - Non-text Content [A]
- 1.2 - Time-based Media
  - 1.2.1 - Audio-only and Video-only (Prerecorded) [A]
  - 1.2.2 - Captions (Prerecorded) [A]
  - 1.2.3 - Audio Description or Media Alternative (Prerecorded) [A]
  - 1.2.4 - Captions (Live) [AA]
  - 1.2.5 - Audio Description (Prerecorded) [AA]
  - 1.2.6 - Sign Language (Prerecorded) [AAA]
  - 1.2.7 - Extended Audio Description (Prerecorded) [AAA]
  - 1.2.8 - Media Alternative (Prerecorded) [AAA]
  - 1.2.9 - Audio-only (Live) [AAA]
- 1.3 - Adaptable
  - 1.3.1 - Info and Relationships [A]
  - 1.3.2 - Meaningful Sequence [A]
  - 1.3.3 - Sensory Characteristics [A]
- 1.4 - Distinguishable
  - 1.4.1 - Use of Color [A]
  - 1.4.2 - Audio Control [A]
  - 1.4.3 - Contrast (Minimum) [AA]
  - 1.4.4 - Resize text [AA]
  - 1.4.5 - Images of Text [AA]
  - 1.4.6 - Contrast (Enhanced) [AAA]
  - 1.4.7 - Low or No Background Audio [AAA]
  - 1.4.8 - Visual Presentation [AAA]
  - 1.4.9 - Images of Text (No Exception) [AAA]

2. Operable

- 2.1 - Keyboard Accessible
  - 2.1.1 – Keyboard [A]
  - 2.1.2 - No Keyboard Trap [A]
  - 2.1.3 - Keyboard (No Exception) [AAA]
- 2.2 - Enough Time
  - 2.2.1 - Timing Adjustable [A]
  - 2.2.2 - Pause, Stop, Hide [A]
  - 2.2.3 – No Timing [AAA]
  - 2.2.4 – Interruptions [AAA]
  - 2.2.5 – Re-authenticating [AAA]
- 2.3 - Seizures
  - 2.3.1 - Three Flashes or Below Threshold [A]
  - 2.3.2 - Three Flashes [AAA]
- 2.4 - Navigable
  - 2.4.1 - Bypass Blocks [A]
  - 2.4.2 - Page Titled [A]
  - 2.4.3 - Focus Order [A]
  - 2.4.4 - Link Purpose (In Context) [A]
  - 2.4.5 - Multiple Ways [AA]
  - 2.4.6 - Headings and Labels [AA]
  - 2.4.7 - Focus Visible [AA]
  - 2.4.8 – Location [AAA]
  - 2.4.9 - Link Purpose (Link Only) [AAA]
  - 2.4.10 - Section Headings [AAA]

3. Understandable

- 3.1 - Readable
  - 3.1.1 - Language of Page [A]
  - 3.1.2 - Language of Parts [AA]
  - 3.1.3 - Unusual Words [AAA]
  - 3.1.4 – Abbreviations [AAA]
  - 3.1.5 - Reading Level [AAA]
  - 3.1.6 – Pronunciation [AAA]
- 3.2 - Predictable
  - 3.2.1 - On Focus [A]
  - 3.2.2 - On Input [A]
  - 3.2.3 - Consistent Navigation [AA]
  - 3.2.4 - Consistent Identification [AA]
  - 3.2.5 - Change on Request [AAA]
- 3.3 - Input Assistance
  - 3.3.1 - Error Identification [A]
  - 3.3.2 - Labels or Instructions [A]
  - 3.3.3 - Error Suggestion [AA]
  - 3.3.4 - Error Prevention (Legal, Financial, Data) [AA]
  - 3.3.5 – Help [AAA]
  - 3.3.6 - Error Prevention (All) [AAA]

4. Robust

- 4.1 - Compatible
  - 4.1.1 – Parsing [A]
  - 4.1.2 – Name, Role, Value [A]

SECTION 508: A BRIEF HISTORY

Section 508, an amendment to the Rehabilitation Act, was passed by Congress in 1998. It required Federal agencies to make electronic and information technology accessible to people with disabilities. It was enacted to eliminate barriers in information technology, to make available new opportunities for people with disabilities, and to encourage development of technologies that will help achieve these goals.

The US Access Board was charged with overseeing the implementation of Section 508 across the federal government. Original guidance focused on subsections of the law including:

- 1194.21 for software applications
- 1194.22 for web-based intranet and information and applications

That guidance was consistent with many of the Web Content Accessibility Guidelines (WCAG), published by the Web Accessibility Initiative (WAI). The WAI is an initiative of the World Wide Web Consortium (W3C), the main international standards organization for the World Wide Web (abbreviated WWW or W3). The current version, WCAG 2.0, was published in December 2008 and became an ISO standard, ISO/IEC 40500:2012 in October 2012. WCAG 2.1 is a "Proposed Recommendation" as of May 2018.

In January 2017, the Access Board approved Section 508 updates that adopt WCAG 2.0 standards as the federal government standards. The Revised 508 Standards became law in 2018. Chapter 2 of the Revised 508 Standards specifies which electronic content must be accessible.

SECTION 508 2.0 OVERVIEW

Section 508 applies to information and communication technology (ICT). 508 compliance guidance is published in Appendixes A – C of the ICT Final Standards and Guidelines. The guidance in those appendixes indicates that compliance requires conformance to WCAG 2.0 standards levels A and AA.

The guidance also includes functional performance criteria for information and communication technologies not covered by WCAG standards. The functional performance criteria are covered in a separate section below.

The following page provides the tables of contents of each of the three appendixes for references.

Appendix A - Application and Scoping Requirements

508 Chapter 1: Application and Administration
- E101 General
- E102 Referenced Standards
- E103 Definitions

508 Chapter 2: Scoping Requirements
- E201 Application
- E202 General Exceptions
- E203 Access to Functionality
- E204 Functional Performance Criteria
- E205 Electronic Content
- E206 Hardware
- E207 Software

Appendix B - Application and Scoping Requirements

255 Chapter 1: Application and Administration
- C101 General
- C102 Referenced Standards
- C103 Definitions

255 Chapter 2: Scoping Requirements
- C201 Application
- C202 Functional Performance Criteria
- C203 Electronic Content
- C204 Hardware
- C205 Software
- C206 Support Documentation and Services

Appendix C - Functional Performance Criteria and Technical Requirements

Chapter 3: Functional Performance Criteria
- 301 General
- 302 Functional Performance Criteria

Chapter 4: Hardware
- 401 General
- 402 Closed Functionality
- 403 Biometrics
- 404 Preservation of Information Provided for Accessibility
- 405 Privacy
- 406 Standard Connections
- 407 Operable Parts
- 408 Display Screens
- 409 Status Indicators
- 410 Color Coding
- 411 Audible Signals
- 412 ICT with Two-Way Communication
- 413 Closed Caption Processing Technologies
- 414 Audio Description Processing Technologies
- 415 User Controls for Captions and Audio Descriptions

Chapter 5: Software
- 501 General
- 502 Interoperability with Assistive Technology
- 503 Applications
- 504 Authoring Tools

Chapter 6: Support Documentation and Services
- 601 General
- 602 Support Documentation
- 603 Support Services

Chapter 7: Referenced Standards
- 701 General
- 702 Incorporation by Reference

Some of these numbers are included in section titles in this document for reference. For example, the number 603 in the section title Support Services (603) refers to section 603 of chapter 6 of Chapter 6 in Appendix C. This provides a cross-reference between this document, other reference materials, and Section 508 legislation.

GOVCIO ACCESSIBILITY CAPABILITIES

This section describes corporate capabilities and resources available to practitioners to make compliance a consistent practice. Compliance is easier and more effective when built into products from the start. This results in the highest quality deliverables and reduces the need for rework late in project life cycles when the cost of change is highest.

BACKGROUND

GovCIO's first Section 508 compliance work began in 1999 when the original standard was released with a June 2000 deadline for federal agencies to comply. We retrofitted hundreds of websites and web applications, including the cabinet level websites DOL.gov, ED.gov, and HUD.gov. GovCIO has developed accessible websites and applications used by millions.

Based on our experience, it takes the same amount of time to build an accessible application or website once team members are trained on accessibility features. And accessibility features improve usability for all users, not just those with visual, hearing, or mobility impairments. Our accessibility capabilities, including development and compliance testing, are described below.

UNDERSTANDING

User interface (UI) components and application information must be presentable, understandable, and navigable to as many users as possible, including users with sight, hearing, and mobility impairments, some of whom may be customizing style sheets, or using one of a wide variety of assistive technologies such as screen magnifiers, screen readers, and speech-to-text applications.

The U.S. Access Board Section 508 Standards were revised in January 2017. Revisions include incorporating WCAG 2.0 at Level A and AA success criteria to websites and non-web electronic documents and software, and addressing access for people with cognitive, language, and learning disabilities, among other changes. The Section 508 Standards apply to federal agency public-facing content, including websites, documents and media, blog posts, and social media sites, as well as non-public-facing content such as web-based intranets, among other forms of electronic content.

SECTION 508 COMPLIANCE MANAGEMENT

We manage our Section 508 Compliance design, development, and testing efforts following federal 508 compliance guidelines, templates, and best practices. This results in products accessible to the broadest possible user community. Artifacts include:

- Revised Section 508 Standards, U.S. Access Board
- HHS Section 508 Compliance Checklist (with Revised standards, for HTML, Audio/Video, PDF, and Microsoft Office software):
- How to Meet WCAG 2 (Quick Reference Guide)

We update our Section 508 Compliance processes and procedures as technologies, tools, and regulations evolve so that system accessibility is maintained throughout technological improvements and regulatory revisions. For example, we have recently shifted to the *Revised Section 508 Compliance Standards*, we have transitioned to guidance documentation based on the Revised standards, such as the new *HHS Section 508 Compliance Checklist*, which supports W3C implementation of WCAG 2.0 A and AA, and has clear guidance on Required and Encouraged fixes for HTML, PDF, Audio/Video, and Microsoft Office products such as Word or Excel.

SECTION 508 COMPLIANCE TRAINING

We train all developers and testers in Section 508 Compliance so that the team understands how to develop and test to the associated criteria. Training occurs upon joining the program, and periodically throughout the year to accommodate advancements in tools, best practices, technologies, and regulations. Training covers the artifacts listed above, plus program-related best practices that we maintain to help guide development and testing. These include Screen Reader Navigation Standard Operating Procedures, Test Tool Best Practices, and Section 508 Compliance Testing Checklists.

SECTION 508 COMPLIANCE DESIGN AND DEVELOPMENT

We design applications with Section 508 Compliance in mind, selecting development frameworks and third-party components that are Section 508 Compliant to ensure correct, built-in accessibility with minimal customization in our products. Designing for accessibility includes best practices such as:

- using simple and clear UI text and content
- using heading elements to clearly convey structure
- providing clear, concise, unambiguous instructions
- using understandable link text
- using effective color contrast within accessibility guidelines
- ensuring color is not used as the sole means of conveying information
- ensuring all form elements have associated labels
- designing for immediate assistive technology recognition of validation errors
- providing consistent navigation on each screen
- creating separate transcripts or captions for video

Developing for accessibility includes following best practices such as:

- avoiding CAPTCHA
- ensuring all elements are keyboard accessible
- ensuring correct and logical keyboard-only navigation
- ensuring effective reading and “tabbing” order for tools such as screen readers
- developing data tables that result in simple screen reader navigation
- providing and ensuring that every form control has a label
- using clear and understandable text alternatives for images
- ensuring modal dialog windows and error/warning message popups are navigable with keyboard-only and by assistive technology
- ensuring the application can be used with browser styles disabled

When designing and developing to WCAG 2.0, developers should follow the WCAG 2.0-recommended techniques, which are specific WCAG 2.0 guidelines for both developers and testers (authors and auditors) on meeting the WCAG success criteria. They include code examples and recommended testing steps. Sufficient techniques are the reliable ways to meet the success criteria. Developers also utilize WCAG 2.0 advisory techniques to improve accessibility. And they analyze the Common Failures section of the success criteria. WCAG 2.0 provides failure guidelines as well: content that has a failure does not meet WCAG success criteria, unless an alternate version is provided without the failure. Our programs maintain multiple Section 508 Compliance subject matter experts (SMEs) who assist developers with the WCAG recommended techniques. The outcome is a product that has been designed and developed to Section 508 standards.

SUPPORT DOCUMENTATION SECTION 508 COMPLIANCE (E208/602)

We ensure Section 508 Compliance for support documentation through the same overall management, training, design, development, and testing capabilities described above, though the details of the approaches may differ. This ensures that all users can utilize our documentation efficiently. Our documentation support includes both documentation internal to the application, such as online help guides and application-produced reports, and documentation external to the application such as standalone user guides. Where necessary, our support documentation lists and explains the usage of accessibility or compatibility features of the application supported by the specific document. Our capabilities include the ability to customize and verify support documentation using the internal accessibility toolkits and guidance provided within software products from vendors such as Adobe and Microsoft. These toolkits and guidance artifacts support customization to ensure the correct use of color, images, links, tables, lists, headings, form elements, among other criteria. This capability includes selecting the most compliant versions of these software products so that our customization work benefits from the latest accessibility technology. Our capability extends to Section 508 Compliance testing of the support documentation. During testing, as we do during design and development, we use the HHS and WCAG 2.0 checklists identified above to ensure compliance with WCAG 2.0 A and AA. Our testing includes navigation through the documentation using assistive technology tools such as screen readers, using the software's internal accessibility verification tools, and manual analysis. If a user requires an alternate document format, we produce the format upon request.

SUPPORT SERVICES SECTION 508 COMPLIANCE (E208/603)

We ensure that all technical support, training materials, and other documentation provided to end-users have been verified for Section 508 Compliance, and lists as well as explains, where necessary, the accessibility and compatibility features of the application and documentation supported. This allows users who use our support services to use their own accessibility tools when accessing and reviewing our support documentation and associated product. Our support services use a variety of communication methods, including phone, email, instant messaging, application landing page postings, and online meetings so that users can use a preferred method of communication.

AUTHORING TOOLS SECTION 508 COMPLIANCE (504)

During application planning and design, we analyze and select authoring tools that meet Section 508 Compliance in accordance with vendor compliance documentation so that, once integrated into our application, system, or environment, the authoring tool presents a low risk of not meeting Section 508 Compliance criteria during integration testing. During integration and testing, we verify that the authoring tool meets WCAG 2.0 A and AA Success Criteria and Conformance Requirements in accordance with the HHS and WCAG 2.0 checklists identified above, along with the target format if the target format supports accessibility. We also verify that specific Section 508 Compliance requirements for authoring tools has been met. This includes ensuring 1) that authoring tools capable of exporting PDF files that conform to ISO 32000-1:2008 (PDF 1.7) can also export PDF files that conform to ANSI/AIIM/ISO 14289-1:2016; 2) that the tool provides a mode of operation that prompts authors to create content that conforms WCAG 2.0 A and AA for supported features and, as applicable, to file formats supported by the authoring tool and 3) that any provided templates for content creation conform to WCAG 2.0 A and AA and to associated file formats. Authoring tool testing by developers and testers includes keyboard-only and accessibility tool navigation, such as the use of screen readers, and verifying any tool output file for document compliance if the format is specified as Section 508 compliant.

SECTION 508 COMPLIANCE TESTING

Our products are tested for Section 508 Compliance by both Developers and Testers using available tools, systems, guidance, best practices and templates. Our capabilities include building accessible code from the start instead of retrofitting for compliance after development, compliance testing, and assertion of compliance levels using the VPAT. We generally manage Section 508 requirements the same way we manage functional requirements. We review code using the *HHS Section 508 Compliance Checklist (Revised Section 508 Compliance)*.

WCAG 2.0 Success Criteria shows testable criteria and is followed by testers when testing an application. Testing the Success Criteria often involves both automated testing, analysis, and manual testing because accessibility test tools have limited capabilities at discovering accessibility-related issues. The average test tool can catch approximately 20-30 percent of the issues. (They may be good at finding color contrast issues, for example, but may not be good at determining whether screen readers can navigate data tables efficiently). Therefore, most accessibility issues must be found through tasks involving analysis or manual testing. Our testers typically use an agency tool such as FDA’s AMP to help manage the scope of testing. AMP provides automatic testing, that is then augmented by manual tasks such as analysis and manual test steps, as recommended in AMP, using WCAG 2.0 A and AA, as follows in accordance with WCAG 2.0:

- Level A: For Level A conformance (the minimum level of conformance), the Web page satisfies all the Level A Success Criteria, or a conforming alternate version is provided.
- Level AA: For Level AA conformance, the Web page satisfies all the Level A and Level AA Success Criteria, or a Level AA conforming alternate version is provided.

Our capabilities include third-party test tools and the use of assistive technology tools. When performing the requisite manual testing tasks, testers verify that the application can be efficiently navigated using keyboard-only (including modal dialog windows and popup messages), navigated with browser Styles disabled, and leverage additional testing tools to ensure quality testing, reducing the chance of human error. At the FDA, these tools include Web Aim WAVE test tool. They also include testing with assistive technology tools such as screen readers to directly verify efficient, simple navigation using these tools, including within modal dialog windows, data tables, and calendar components. In addition, WCAG 2.0 provides the following guidance: The tests are only for a technique, they are not tests for conformance to WCAG success criteria.

- Failing a technique test does not necessarily mean failing WCAG, because the techniques are discrete (that is, they address one specific point) and they are not required.
    - Note: Failures are particularly useful for evaluations because they do indicate non-conformance (unless an alternate version is provided without the failure).
- Content can meet WCAG success criteria in different ways other than W3C's published sufficient techniques.
- Content that passes the sufficient techniques for a specific technology does not necessarily meet all WCAG success criteria. Some success criteria have only general techniques, not technology-specific techniques.

- The content must be accessibility supported for the content's users. Some sufficient techniques require browser, assistive technology, or other support that some users might not have.

CONFORMANCE LEVELS

Conformance to a standard means satisfying the 'requirements' of the standard. In WCAG 2.0 the 'requirements' are the Success Criteria, which are categorized in three levels:

- Level A (minimum) – Addresses the most basic web accessibility features, but does not generally achieve broad accessibility.
- Level AA (mid-range) – Addresses most common barriers for disabled users and aligns to the Revised 508 Standards
- Level AAA (highest) – Addresses the highest level of web accessibility, but is not recommended as a general policy, because it is not possible to satisfy all criteria for some content.

WCAG 2.0 Level AA conformance for web sites / web applications requires:

1. **Level AA** – The web page satisfies all the Level A and Level AA Success Criteria, or a Level AA conforming alternate version is provided.
2. **Full Page** - Conformance is for full web pages only and cannot be achieved if part of a web page is excluded.
3. **Completeness** – When a web page is one of a series of web pages presenting a process (i.e., a sequence of steps that need to be completed in order to accomplish an activity), all web pages in the process conform at the specified level or better.
4. **Accessibility-Supported Technology** – Only accessibility-supported ways of using technologies are used to satisfy the SC. In other words, each page works with assistive technologies and standard methods are used to expose content to assistive technology, or accessibility features of mainstream user agents.
5. **Non-Interference** – If technologies are used in a way that is not accessibility supported, or in a non-conforming way, they do not block the rest of the page. In other words, technologies that are not accessibility-supported can be used, as long as the non-accessibility-supported material does not interfere, and all information is also available using technologies that are accessibility-supported.

When reporting conformance levels, the following terminology is commonly used to describe how well a technology product meets each SC:

- **Supports**: The functionality of the product has at least one method that meets the SC without known defects, or the product meets the SC with equivalent facilitation
- **Supports with Exceptions**: Some functionality of the product does not meet the SC
- **Does Not Support**: The majority of product functionality does not meet the SC
- **Not Applicable**: The SC is not relevant to the product
- **Not Evaluated**: The product has not been evaluated against the SC (for WCAG 2.0 Level AAA SC only)

VPAT (VOLUNTARY PRODUCT ACCESSIBILITY TEMPLATE)

Conformance is commonly reported on the Voluntary Product Accessibility Template (VPAT). The VPAT contains a table that reports compliance against each standard using the conformance levels above.

The VPAT table has three columns, as shown in the example below from VPAT version 2.1. The first column lists the SC. Note that both the WCAG (2017 Section 508) and European (EN 301 549) standards are included. In most situations, the VPAT is completed for Section 508 only and the EN 301 549 standards can be removed. The list under 'Also applies to' helps determine the applicability of the standard for column 2.

Column 2 is where the level of conformance is reported using the values listed above. Use column 3 to provide an explanation when a SC is partially met.

Criteria	Conformance Level	Remarks and Explanations
1.1.1 Non-text Content (Level A) Also applies to: EN 301 549 Criteria • 9.2.1 (Web) • 10.2.1 (non-web document) • 11.2.1.1 (Software) • 11.2.2.1 (Closed Functionality Software) • 11.6.2 (Authoring Tool) • 12.1.2 (Product Docs) • 12.2.4 (Support Docs) 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Electronic Docs: Software: Closed: Authoring Tool:	Web: Electronic Docs: Software: Closed: Authoring Tool:

CONFORMANCE TO SECTION 508 STANDARDS

This section identifies GovCIO's capability to conform with each Level A and Level AA success criterion (SC). Conformance is commonly reported on the Voluntary Product Accessibility Template (VPAT). We use the VPAT to report compliance against each standard in the A and AA conformance levels.

The VPAT table has three columns, as shown in the example below from VPAT version 2.1. The first column lists the SC. Note that both the WCAG (2017 Section 508) and European (EN 301 549) standards are included. In most situations, the VPAT is completed for Section 508 only and the EN 301 549 standards can be removed. The list under 'Also applies to' helps determine the applicability of the standard for column 2.

Column 2 is where the level of conformance is reported using the values listed above. Use column 3 to provide an explanation when a SC is partially met.

The first VPAT table reports compliance to Level A standards. The second reports compliance to Level AA standards. GovCIO has the capability to meet the compliance levels listed in the sections below using standard tools and technologies. It is common practice to embed a third-party authoring tool in the products we deliver when one is needed, instead of developing one in house. Accordingly, values from the TinyMCE text editor VPAT are used in the tables below. We deliver products to the conformance levels below unless otherwise directed by our customers, typically the COR or equivalent.

LEVEL A COMPLIANCE

The table below identifies our capability to comply with each Level A standard.

Criteria	Conformance Level	Remarks and Explanations
1.1.1 Non-text Content (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports with Exceptions	Web: Electronic Docs: Software: Authoring Tool: Fixing of labels on form elements and a mechanism setting alt text (or changing the field drawn from) necessary to make fully accessible.
1.2.1 Audio-only and Video-only (Prerecorded) (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:

Criteria	Conformance Level	Remarks and Explanations
1.2.2 Captions (Prerecorded) (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
1.2.3 Audio Description or Media Alternative (Prerecorded) (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
1.3.1 Info and Relationships (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
1.3.2 Meaningful Sequence (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
1.3.3 Sensory Characteristics (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
1.4.1 Use of Color (Level A) Aplies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
1.4.2 Audio Control (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:

Criteria	Conformance Level	Remarks and Explanations
2.1.1 Keyboard (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports with Exceptions	Web: Electronic Docs: Software: Authoring Tool: Improved labeling of forms and keyboard options for click/drag operations (reordering pages, creating exhibit pages) necessary to be fully accessible.
2.1.2 No Keyboard Trap (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
2.2.1 Timing Adjustable (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
2.2.2 Pause, Stop, Hide (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
2.3.1 Three Flashes or Below Threshold (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
2.4.1 Bypass Blocks (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) – Does not apply to non-web software • 504.2 (Authoring Tool) • 602.3 (Support Docs) – Does not apply to non-web docs	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:

Criteria	Conformance Level	Remarks and Explanations
2.4.2 Page Titled (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
2.4.3 Focus Order (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
2.4.4 Link Purpose (In Context) (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
3.1.1 Language of Page (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
3.2.1 On Focus (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
3.2.2 On Input (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
3.3.1 Error Identification (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:

Criteria	Conformance Level	Remarks and Explanations
3.3.2 Labels or Instructions (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
4.1.1 Parsing (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
4.1.2 Name, Role, Value (Level A) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:

LEVEL AA COMPLIANCE

The table below identifies our capability to comply with each Level AA standard.

Criteria	Conformance Level	Remarks and Explanations
1.2.4 Captions (Live) (Level AA) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
1.2.5 Audio Description (Prerecorded) (Level AA) Applies to: 2017 Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:

Criteria	Conformance Level	Remarks and Explanations
1.4.3 Contrast (Minimum) (Level AA) Applies to: 2017 Section 508 - 501 (Web)(Software) - 504.2 (Authoring Tool) - 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports with Exceptions	Web: Electronic Docs: Software: Authoring Tool: Contrast for link text in admin interface should be increased to improve accessibility.
1.4.4 Resize text (Level AA) Applies to: 2017 Section 508 - 501 (Web)(Software) - 504.2 (Authoring Tool) - 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
1.4.5 Images of Text (Level AA) Applies to: 2017 Section 508 - 501 (Web)(Software) - 504.2 (Authoring Tool) - 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
2.4.5 Multiple Ways (Level AA) Applies to: 2017 Section 508 - 501 (Web)(Software) – Does not apply to non-web software - 504.2 (Authoring Tool) - 602.3 (Support Docs) – Does not apply to non-web docs	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
2.4.6 Headings and Labels (Level AA) Applies to: 2017 Section 508 - 501 (Web)(Software) - 504.2 (Authoring Tool) - 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
2.4.7 Focus Visible (Level AA) Applies to: 2017 Section 508 - 501 (Web)(Software) - 504.2 (Authoring Tool) - 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports with Exceptions	Web: Electronic Docs: Software: Authoring Tool: Creation of strong default focus styles necessary to comply.

Criteria	Conformance Level	Remarks and Explanations
3.1.2 Language of Parts (Level AA) Applies to: 2017 Section 508 - 501 (Web)(Software) - 504.2 (Authoring Tool) - 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
3.2.3 Consistent Navigation (Level AA) Applies to: 2017 Section 508 - 501 (Web)(Software) – Does not apply to non-web software - 504.2 (Authoring Tool) - 602.3 (Support Docs) – Does not apply to non-web docs	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:
3.2.4 Consistent Identification (Level AA) Applies to: 2017 Section 508 - 501 (Web)(Software) – Does not apply to non-web software - 504.2 (Authoring Tool) - 602.3 (Support Docs) – Does not apply to non-web docs	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports with Exceptions	Web: Electronic Docs: Software: Authoring Tool: Ensuring that all form elements have labels and an improved system for identifying form errors are necessary to make all forms accessible.
3.3.3 Error Suggestion (Level AA) Applies to: 2017 Section 508 - 501 (Web)(Software) - 504.2 (Authoring Tool) - 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool:	Web: Electronic Docs: Software: Authoring Tool:
3.3.4 Error Prevention (Legal, Financial, Data) (Level AA) Applies to: 2017 Section 508 - 501 (Web)(Software) - 504.2 (Authoring Tool) - 602.3 (Support Docs)	Web: Supports Electronic Docs: Supports Software: Supports Authoring Tool: Supports	Web: Electronic Docs: Software: Authoring Tool:

FUNCTIONAL PERFORMANCE CRITERIA

The functional performance criteria of Chapter 3 are outcome-based provisions that address accessibility relevant to disabilities impacting vision, hearing, color perception, speech, manual dexterity, reach, and strength. The functional performance criteria are to be used where a technical requirement is silent or to determine whether an alternative means of compliance is sufficient under the provision of equivalent facilitation.

CRITERIA LIST

Where the requirements do not address one or more functions of ICT, the functions not addressed shall conform to the Functional Performance Criteria listed below.

Without Vision (302.1). Where a visual mode of operation is provided, ICT shall provide at least one mode of operation that does not require user vision.

With Limited Vision (302.2). Where a visual mode of operation is provided, ICT shall provide at least one mode of operation that enables users to make use of limited vision.

Without Perception of Color (302.3). Where a visual mode of operation is provided, ICT shall provide at least one visual mode of operation that does not require user perception of color.

Without Hearing (302.4). Where an audible mode of operation is provided, ICT shall provide at least one mode of operation that does not require user hearing.

With Limited Hearing (302.5). Where an audible mode of operation is provided, ICT shall provide at least one mode of operation that enables users to make use of limited hearing.

Without Speech (302.6). Where speech is used for input, control, or operation, ICT shall provide at least one mode of operation that does not require user speech.

With Limited Manipulation (302.7). Where a manual mode of operation is provided, ICT shall provide at least one mode of operation that does not require fine motor control or simultaneous manual operations.

With Limited Reach and Strength (302.8). Where a manual mode of operation is provided, ICT shall provide at least one mode of operation that is operable with limited reach and limited strength.

With Limited Language, Cognitive, and Learning Abilities (302.9). ICT shall provide features making its use by individuals with limited cognitive, language, and learning abilities simpler and easier.

The need to adhere to functional performance requirements may become more prevalent as new technologies are introduced that were not envisioned by the current WCAG standards. The original Section 508 standards were published before wide adoption of smartphones and tablets. Should a new technology emerge that gains broad user acceptance, these criteria could become more relevant. As of May 2018, the WCAG 2.0 standards cover the standard deliverables GovCIO provides to its federal government customers.

REPORTING CONFORMANCE TO FUNCTIONAL PERFORMANCE CRITERIA USING THE VPAT

The VPAT includes a section for reporting conformance to functional performance criteria.

Criteria	Conformance Level	Remarks and Explanations
302.1 Without Vision	Supports	
302.2 With Limited Vision	Supports	
302.3 Without Perception of Color	Supports	
302.4 Without Hearing	Supports	
302.5 With Limited Hearing	Supports	
302.6 Without Speech	Supports	
302.7 With Limited Manipulation	Supports	
302.8 With Limited Reach and Strength	Supports	
302.9 With Limited Language, Cognitive, and Learning Abilities	Supports	

INTEROPERABILITY WITH ASSISTIVE TECHNOLOGIES (502)

We must deliver software that is compatible with assistive technologies to federal agencies. Most platforms we use have accessibility features. We must enable our software to use these features where possible. *Accessibility Tech* summarized how accessibility features differ across operating systems:

> In business, the most common graphic operating systems today are Microsoft's Windows (about 89% of all users) and Apple's Mac OS. UNIX and Linux (an open-source UNIX-like operating system) are also used.
>
> One role of the graphic operating system is to provide an application program interface (API) to programmers so they can write applications consistent with the operating environment. All programs developed using a common API will have a similar interface, which makes applications easier to learn and use. The API provides a set of building blocks, which programmers assemble into an application. It is important that the API provide support for accessibility. For example, all menus and controls in a graphic user interface should be accessible via keyboard, not just mouse, and should be displayed with a font and color scheme that can be easily customized by the user. As long as the API provides the means for delivering these and other accessibility features, applications within that environment can be easily made accessible by software application developers.
>
> To date, there has been a significant disparity in the accessibility of operating systems' APIs. Microsoft addressed many of the accessibility problems of its Windows API fairly early on and provided developers with the tools to develop applications that were accessible. Most Windows applications, for example, are entirely operable via keyboard so a mouse isn't required by the user. Other graphic operating systems have failed to deliver comparable accessibility, particularly for non-mousing and non-visual users.
>
> Basic Accessibility Features
>
> Some assistive technology and other accessibility features come bundled with all operating systems, but typically these applications provide only a minimal level of accessibility, not the full set of features that many users require for equal access to the OS and its applications. The following are common built-in accessibility features across all operating systems:
>
> Keyboard customization allows users to adjust keyboard behavior so they can: A) press one key at a time in place of multi-key combinations; B) use the keyboard to control mouse movements; and C) change the length of time it takes for a keystroke to be registered.
>
> Display customization allows users to control the display contrast, font style and size, size of icons, and other display characteristics.
>
> Multimode alerts provide system notifications visually for users who can't hear auditory alerts.
>
> In addition to these basic accessibility features, both Windows and Mac OS include basic screen magnification software (Magnifier and Zoom, respectively). Windows additionally provides a basic screen reader application (text-to-speech translator) called Narrator, along with built-in speech recognition software, which allows the user to control the computer through a series of voice commands. Mac OS X includes limited text-to-speech capabilities and speech recognition (voice command) through its VoiceOver application.

Each of these products provides a basic level of access, but these applications fall far short of the more fully featured screen magnification and screen-reading applications that are available for Windows via third-party developers.

Microsoft – The Microsoft® Active Accessibility® (MSAA) standard has been available since Windows 95. These early efforts to support accessibility, combined with the market dominance of Windows, led to a disproportionate number of assistive technologies being developed for Windows.

Microsoft UI Automation is the new accessibility framework for Microsoft Windows, available on all operating systems that support Windows Presentation Foundation (WPF). UI Automation provides programmatic access to most user interface (UI) elements on the desktop, enabling assistive technology products such as screen readers to provide information about the UI to end users and to manipulate the UI by means other than standard input.

In Windows, there is an "Ease of Access Center" that provides a convenient, centralized place to locate accessibility settings and programs to make your computer easier to use. The Ease of Access Center can be found in the Control Panel by selecting Windows logo key+U and also when logging into Windows.

Microsoft’s Enable website includes detailed descriptions of accessibility features in current and previous releases of Microsoft Windows, step-by-step tutorials, and guides for users with specific disabilities.

Apple – With the release of Mac OS X, Apple improved the accessibility of its operating system. Mac OS X includes a screen magnification and VoiceOver, a screen-access technology, for the blind and visually impaired. To assist those with cognitive and learning disabilities, every Mac includes an alternative, simplified user interface that rewards exploration and learning. And, for those who find it difficult to use a mouse, every Mac computer includes Mouse Keys, Slow Keys, and Sticky Keys, which adapt the computer to the user’s needs and capabilities.

Apple has now built accessibility into its Carbon application programming interface (API), which allows Mac OS X applications to more effectively communicate with assistive technologies. Despite these efforts, however, there still are comparatively few assistive technology products available for Mac OS. For example, there is only one screen reading software option available, which has not provided the level of robustness of equivalent PC products. Additional information about Mac OS accessibility is provided on the Apple Accessibility website. This site includes specific steps for activating and using the accessibility features of Mac OS X.

Linux – Linux differs from both Windows and Mac OS in that it is an open source operating system and is supported and advanced by a dedicated community of developers. To date, the Linux developer community has produced a basic core set of accessibility features (as described above), as well as a combined screen reader/screen magnification application, Braille output software, and an on-screen keyboard. Each of these products was developed for the popular GNOME desktop, a graphic interface environment that runs on both Linux and Unix.

According to Chapter 5 of Appendix A of the standard, an exception exists for web applications that do not have access to platform accessibility services and do not include components that have access to platform accessibility services. These web applications do not need to be compatible with assistive technologies provided they conform to Level A and Level AA WCAG 2.0 standards.

PLATFORM ACCESSIBILITY FEATURES

Software developed on platforms with accessibility features must make use of those features. Chapter 5 of Appendix A requires platform software to provide user control over platform features that are defined in the platform documentation as accessibility features. It also restricts software from disrupting platform features that are defined in the platform documentation as accessibility features.

ACCSSIBILITY SERVICES

Platform software and software tools that are provided by the platform developer must provide a documented set of accessibility services that support applications running on the platform to interoperate with assistive technology. Applications that are also platforms must expose the underlying platform accessibility services or implement other documented accessibility services. The standard identifies the following accessibility services:

- **Object Information**. The object role, state(s), properties, boundary, name, and description shall be programmatically determinable.
- **Modification of Object Information**. States and properties that can be set by the user shall be capable of being set programmatically, including through assistive technology.
- **Row, Column, and Headers**. If an object is in a data table, the occupied rows and columns, and any headers associated with those rows or columns, shall be programmatically determinable.
- **Values**. Any current value(s), and any set or range of allowable values associated with an object, shall be programmatically determinable.
- **Modification of Values**. Values that can be set by the user shall be capable of being set programmatically, including through assistive technology.
- **Label Relationships**. Any relationship that a component has as a label for another component, or of being labeled by another component, shall be programmatically determinable.
- **Hierarchical Relationships**. Any hierarchical (parent-child) relationship that a component has as a container for, or being contained by, another component shall be programmatically determinable.
- **Text**. The content of text objects, text attributes, and the boundary of text rendered to the screen, shall be programmatically determinable.
- **Modification of Text**. Text that can be set by the user shall be capable of being set programmatically, including through assistive technology.
- **List of Actions**. A list of all actions that can be executed on an object shall be programmatically determinable.
- **Actions on Objects**. Applications shall allow assistive technology to programmatically execute available actions on objects.
- **Focus Cursor**. Applications shall expose information and mechanisms necessary to track focus, text insertion point, and selection attributes of user interface components.
- **Modification of Focus Cursor**. Focus, text insertion point, and selection attributes that can be set by the user shall be capable of being set programmatically, including through the use of assistive technology.
- **Event Notification**. Notification of events relevant to user interactions, including but not limited to, changes in the component’s state(s), value, name, description, or boundary, shall be available to assistive technology.

PLATFORM ACCSSIBILITY FEATURES

Platforms and platform software must conform to the requirements in ANSI/HFES 200.2, Human Factors Engineering of Software User Interfaces — Part 2: Accessibility (2008) listed below:

A. Section 9.3.3 Enable sequential entry of multiple (chorded) keystrokes;

B. Section 9.3.4 Provide adjustment of delay before key acceptance;

C. Section 9.3.5 Provide adjustment of same-key double-strike acceptance;

D. Section 10.6.7 Allow users to choose visual alternative for audio output;

E. Section 10.6.8 Synchronize audio equivalents for visual events;

F. Section 10.6.9 Provide speech output services; and

G. Section 10.7.1 Display any captions provided.

REPORTING SOFTWARE COMPATIBILITY WITH ASSISTIVE TECHNOLOGIES USING THE VPAT

The VPAT includes a section for reporting conformance to the requirements for software interoperability with assistive technologies. The level of interoperability varies by platform software as described above. Since most federal users use Windows software, the VPAT based table below is completed for Windows 10 platform software.

The Microsoft website was used to complete the table. The table reports target compliance levels for custom software developed to operate on Windows 10. It assumes that Windows 10 platform software is used. If another platform software is used, including a different version of Windows, the table will need to be updated.

Criteria	Conformance Level	Remarks and Explanations
501.1 Scope – Incorporation of WCAG 2.0 AA	See WCAG 2.0 section	See information in WCAG section
502 Interoperability with Assistive Technology	Heading cell – no response required	Heading cell – no response required
502.2.1 User Control of Accessibility Features	Supports	
502.2.2 No Disruption of Accessibility Features	Supports with Exceptions	Only the Windows assistive technologies Narrator, Screen Magnifier, High Contrast, Filter Keys and Sticky Keys are available in setup wizards. Narrator speech output collides with Cortana speech recognition. The Cortana speech interface is unsuitable for Narrator users in non-dictation scenarios. In high contrast mode, Cortana’s Reminder feature may require switching focus to be visible. Mobility Center has buttons and text that are not visible. Links in PDFs appear as plain-text in Edge (links do function). After logging to Windows with Windows Hello, Narrator does not announce that login was successful.

Criteria	Conformance Level	Remarks and Explanations
502.3 Accessibility Services	Heading cell – no response required	Heading cell – no response required
502.3.1 Object Information	Supports	
502.3.2 Modification of Object Information	Supports	
502.3.3 Row, Column, and Headers	Supports	
502.3.4 Values	Supports	
502.3.5 Modification of Values	Supports	
502.3.6 Label Relationships	Supports	
502.3.7 Hierarchical Relationships	Supports	
502.3.8 Text	Supports	
502.3.9 Modification of Text	Supports	
502.3.10 List of Actions	Supports	
502.3.11 Actions on Objects	Supports	
502.3.12 Focus Cursor	Supports	
502.3.13 Modification of Focus Cursor	Supports	
502.3.14 Event Notification	Supports	
502.4 Platform Accessibility Features	Supports	
503 Applications	Heading cell – no response required	Heading cell – no response required
503.2 User Preferences	Supports with Exceptions	Please see description for 502.2.2 above
503.3 Alternative User Interfaces	Supports with Exceptions	Please see description for 502.2.2 above
503.4 User Controls for Captions and Audio Description	Heading cell – no response required	Heading cell – no response required
503.4.1 Caption Controls	Supports	

Criteria	Conformance Level	Remarks and Explanations
503.4.2 Audio Description Controls	Supports	
504 Authoring Tools	Heading cell – no response required	Heading cell – no response required
504.2 Content Creation or Editing (if not authoring tool, enter “not applicable”)	See WCAG 2.0 section	See information in WCAG section
504.2.1 Preservation of Information Provided for Accessibility in Format Conversion	Supports	
504.2.2 PDF Export	Not Applicable	This standard applies to authoring tools. Windows is not an authoring tool.
504.3 Prompts	Not Applicable	This standard applies to authoring tools. Windows is not an authoring tool.
504.4 Templates	Supports	

DOCUMENTATION AND SUPPORT SERVICES

We must both develop accessible products and provide accessible support documentation and support services. User facing documentation and service support services, including but not limited to Help Desk services, must be accessible.

SUPPORT DOCUMENTATION (602)

Documentation that supports the use of ICT must be accessible. Examples of documentation that supports ICT are installation guides, user guides, and manuals that describe the features of a product and how it is used. Documentation may take the form of stand-alone documents or be integrated into products as on-line or context-sensitive help. Note that requirements, design, test, and other documentation are not covered by this standard.

Documentation that supports the use of ICT must list and explain how to use the product's accessibility features. Cover both built in accessibility features and compatibility with assistive technologies like those identified in the Interoperability with Assistive Technologies section of this document.

An example is the ability to navigate using the keyboard. Voice recognition, screen readers, and alternative keyboards rely on keyboard navigation. Keyboard navigation includes support for the following:

- Cursor keys (up, down, left and right arrows)
- Tab and Shift-Tab (to cycle through fields)
- Enter or Spacebar (to select or activate)
- Hot key combinations, macros, and other keyboard acceleration mechanisms

Documentation in electronic format shall conform to all Level A and Level AA Success Criteria and all Conformance Requirements in WCAG 2.0.

The VPAT can be used to report the level of document conformance to each applicable WCAG success criterion.

SUPPORT SERVICES (603)

ICT support services must accommodate the communication needs of individuals with disabilities. That means support services including, but not limited to, help desks, call centers, technical support, and training services, must be accessible. Like the requirements for support documentation, we must provide information on the accessibility and compatibility features of support services.

A best practice is for ICT support services to provide training programs about the following topics:

- accessibility requirements for individuals with disabilities
- methods of communication used by individuals with disabilities
- assistive technology commonly used with ICT products
- designing for accessibility
- solutions for accessibility and compatibility of ICT with assistive technology
- the use of people-first language; and sensitivity training concerning disability issues

To be effective, communication with individuals with disabilities should include alternate methods of communication for both in-person and remote communication. Examples of alternative methods are sign language interpreters, assistive listening systems, TTYs, real time captioning, and telecommunications relay services. Examples of telecommunication relay services are TTY speech-to-speech and video relay.

A best practice is for help desk and other ICT support services to use a variety of communication technologies. Examples of such communication technologies include Internet posting (such as message boards and website blogs), cellular telephones, two way radios, email, fax, postal mail, texting, and instant messaging.

When support services provide documentation, the documentation materials must be accessible per the standards for documentation above.

REPORTING CONFORMANCE TO SUPPORT DOCUMENTATION AND SERVICES CRITERIA ON THE VPAT

The VPAT includes a section for reporting conformance to functional performance criteria.

Criteria	Conformance Level	Remarks and Explanations
601.1 Scope	Heading cell – no response required	Heading cell – no response required
602 Support Documentation	Heading cell – no response required	Heading cell – no response required
602.2 Accessibility and Compatibility Features	Supports	
602.3 Electronic Support Documentation	See WCAG 2.0 section	See information in WCAG section
602.4 Alternate Formats for Non-Electronic Support Documentation	Supports	
603 Support Services	Heading cell – no response required	Heading cell – no response required
603.2 Information on Accessibility and Compatibility Features	Supports	
603.3 Accommodation of Communication Needs	Supports with Exceptions	We fully support use of a variety of communication technologies including, but not limited to, Internet posting (message boards, blogs, wikis, knowledge repositories), cell phones, email, fax, postal mail, texting, and instant messaging. While we can provide sign language interpreters on subcontract or other means, our contracts typically do not include provisions for this type of support. We typically collaborate with sign language interpreters contracted separately by our government customers. Similarly, we typically work with assistive technologies provided by our customers. Examples include TTY speech-to-speech and video relays.

TESTING FOR SECTION 508 COMPLIANCE

Testing enables us to assert the level of Section 508 compliance with confidence. The more robust the testing, the higher our confidence in the results. The three main testing techniques are listed below:

- Automated testing using software that checks for compliance
- Testing using one or more assistive technologies
- Testing of Techniques to include review of the finished product and/or the underlying code

A combination of these techniques yields the best results.

AUTOMATED TESTING TOOLS

Automated compliance checkers scan products with minimal user interaction and produce reports of compliance levels. The reports typically include compliance errors and warnings. Some tools are available online and others are available for download and installation.

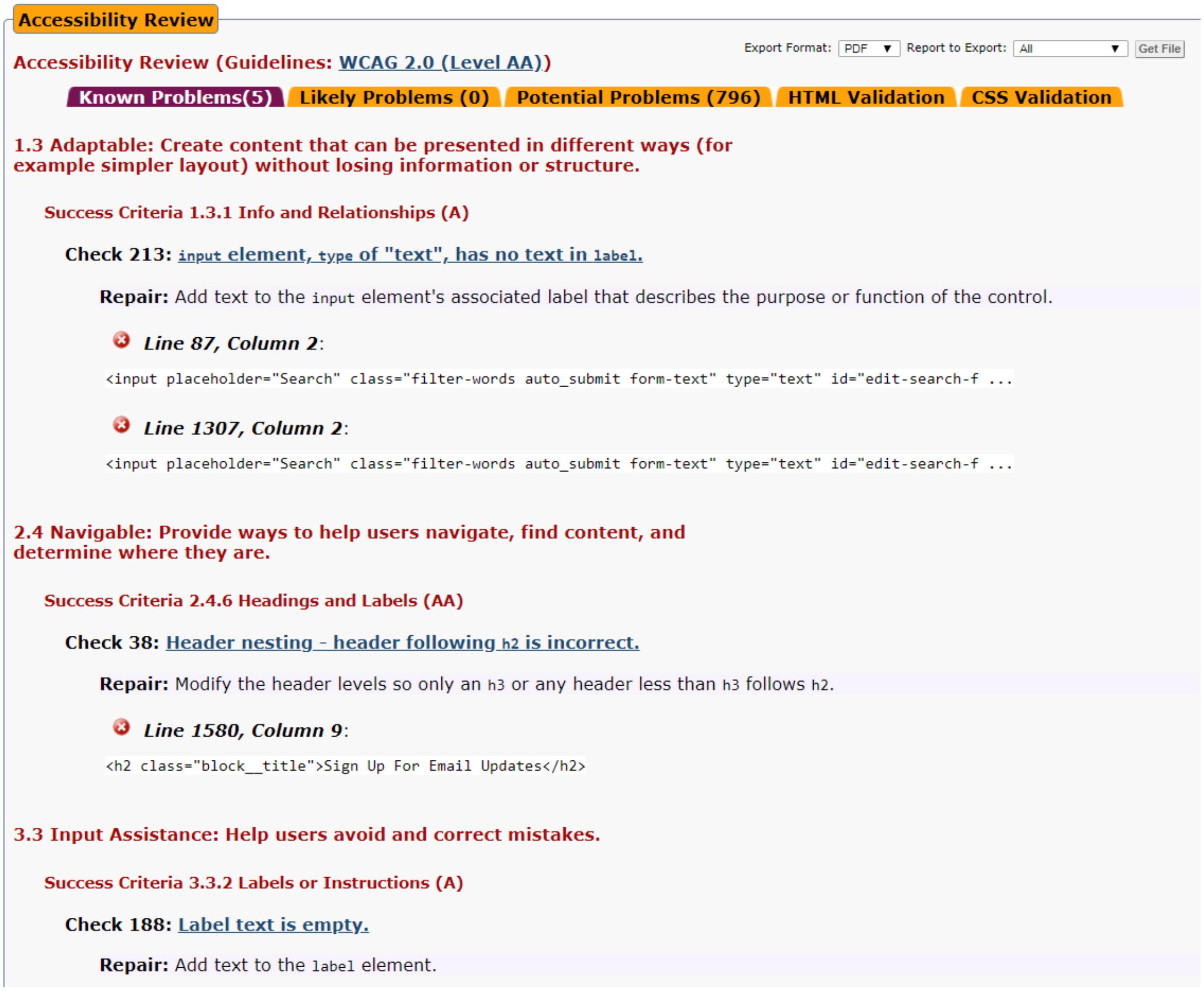
The following are free tools that can be used to assess compliance:

- AChecker – free online tool that scans websites and delivers an accessibility report against the WCAG 2.0 guidelines when you enter a URL
- Wave v 4.0 - free web accessibility evaluation tool provided by WebAIM
- Cynthia Says – free online web accessibility solution identifies content per WCAG guidelines, companion to Compliance Sherriff in list below
- FireEyes - free Firefox browser add-on tool for testing websites for 508 compliance
- Tenon Check - free Chrome browser add-on tool to test accessibility per WCAG 2.0

The following are tools available for purchase that can be used to assess compliance:

- AMP (Accessibility Management Platform) – Testing engine with training that enables developers to incorporate accessibility compliance into existing development processes
- Compliance Sherriff – Cryptzone acquired Hi Software and has replaced AccVerify and AccMonitor with its suite of Compliance Sherriff products, which includes the former Compliance Deputy product

The screen capture below shows a compliance report from the automated tool AChecker. It is open to the Known Issues tab, indicating significant compliance issues. The issues are reported in WCAG 2.0 success criterion order. The report identifies issue by line number and provides a message on how to go about repairing the code for compliance.



Accessibility Review (Guidelines: WCAG 2.0 (Level AA))

1.3 Adaptable: Create content that can be presented in different ways (for example simpler layout) without losing information or structure.

Success Criteria 1.3.1 Info and Relationships (A)

Check 213: `input` element, `type` of "text", has no text in `label`.

Repair: Add text to the `input` element's associated label that describes the purpose or function of the control.

Line 87, Column 2:

```
<input placeholder="Search" class="filter-words auto_submit form-text" type="text" id="edit-search-f ...
```

Line 1307, Column 2:

```
<input placeholder="Search" class="filter-words auto_submit form-text" type="text" id="edit-search-f ...
```

2.4 Navigable: Provide ways to help users navigate, find content, and determine where they are.

Success Criteria 2.4.6 Headings and Labels (AA)

Check 38: Header nesting - header following `h2` is incorrect.

Repair: Modify the header levels so only an `h3` or any header less than `h3` follows `h2`.

Line 1580, Column 9:

```
<h2 class="block__title">Sign Up For Email Updates</h2>
```

3.3 Input Assistance: Help users avoid and correct mistakes.

Success Criteria 3.3.2 Labels or Instructions (A)

Check 188: Label text is empty.

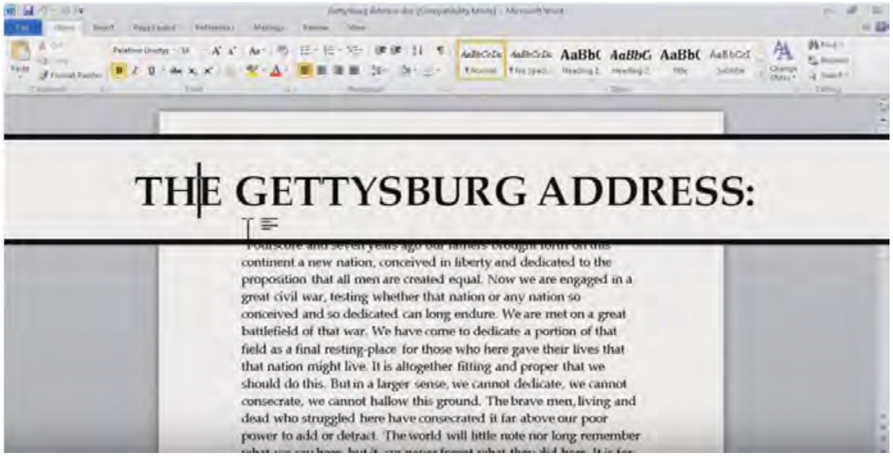
Repair: Add text to the `label` element.

TESTING FOR INTEROPERABILITY WITH ASSISTIVE TECHNOLOGIES

Screen readers and magnifiers are commonly used to test accessibility. A screen reader is an assistive technology that enables a sight challenged user to listen to the content via an automated voice or convert it to braille. A screen magnifier is an assistive technology that enables partially sighted users to significantly magnify a section of the screen by highlighting it with the mouse. Popular screen readers and magnifiers include:

- ZoomText – offers both screen magnifier and magnifier/reader products
- JAWS – provides speech and Braille output
- NVDA (Non Visual Desktop Access) – free screen reader that converts text to speech or Braille
- VoiceOver – Apple screen reader that comes on the Mac, iPhone, and iPad products
- Dolphin Readers – SuperNova magnifier available on USB, EasyReader for iOS and Android, and more

Content can be tested by obtaining a screen reader, using it to convert text to speech, listening to the results, and comparing them to the content on screen. This enables the sighted and hearing tester to verify that the content is equivalent for the sighted user and the user using the assistive technology. The screen captures below show assistive technologies in use. In the first image, a portion of a Word document is magnified. In the second, text on a smartphone is magnified.



THE GETTYSBURG ADDRESS:

Fourscore and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty and dedicated to the proposition that all men are created equal. Now we are engaged in a great civil war, testing whether that nation or any nation so conceived and so dedicated can long endure. We are met on a great battlefield of that war. We have come to dedicate a portion of that field as a final resting-place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this. But in a larger sense, we cannot dedicate, we cannot consecrate, we cannot hallow this ground. The brave men, living and dead who struggled here have consecrated it far above our poor power to add or detract. The world will little note nor long remember

It was the best time of day, the July sky cloudless, the slowly ...tting sun a

Alternative input devices are also used to interact with ICT products. Touchless keyboards enable individuals with autism, arthritis, multiple sclerosis or carpal tunnel syndrome to more easily type and mouse. Mouse emulators enable mobility challenged users to provide mouse input with an alternative device. Eye pointers enable human – computer interaction by tracking eye focus on the screen and enabling users to provide input using a series of eye blinks. Accessible ICT products enable users to interact with them using these and other assistive technologies.

TESTING OF TECHNIQUES

Much of compliance testing is manual work. Guidance on test procedures and expected results are available for all techniques in this document that are linked to external content.

Each technique has tests that help:

- authors verify that they implemented the technique properly, and
- evaluators determine if web content meets the technique.

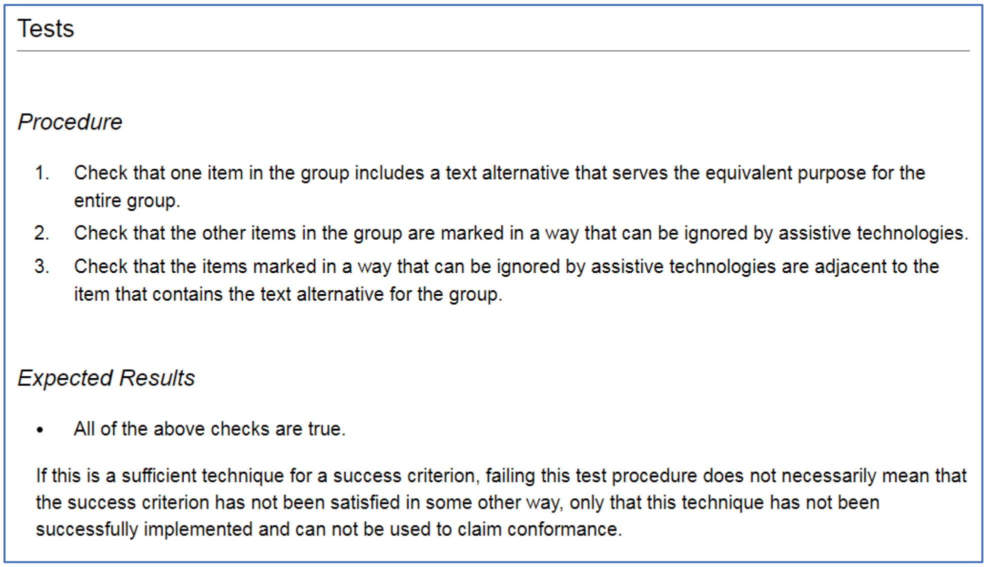
The tests are only for a technique, they are not tests for conformance to WCAG success criteria.

- Failing a technique test does not necessarily mean failing WCAG, because the techniques are discrete (that is, they address one specific point) and they are not required.
- Content can meet WCAG success criteria in different ways other than W3C's published sufficient techniques.
- Content that passes the sufficient techniques for a specific technology does not necessarily meet all WCAG success criteria. Some success criteria have only general techniques, not technology-specific techniques.
- The content must be accessibility supported for the content's users. Some sufficient techniques require browser, assistive technology, or other support that some users might not have.

Thus while the techniques are useful for evaluating content, evaluations must go beyond just checking the sufficient technique tests in order to evaluate how content conforms to WCAG success criteria.

Failures are particularly useful for evaluations because they do indicate non-conformance (unless an alternate version is provided without the failure).

The screen capture below shows the test procedures and expected results for the first technique linked in this document, G196.



Tests

Procedure

1. Check that one item in the group includes a text alternative that serves the equivalent purpose for the entire group.
2. Check that the other items in the group are marked in a way that can be ignored by assistive technologies.
3. Check that the items marked in a way that can be ignored by assistive technologies are adjacent to the item that contains the text alternative for the group.

Expected Results

- All of the above checks are true.

If this is a sufficient technique for a success criterion, failing this test procedure does not necessarily mean that the success criterion has not been satisfied in some other way, only that this technique has not been successfully implemented and can not be used to claim conformance.

UNDERSTANDING AND MEETING LEVEL A AND AA STANDARDS

This section provides guidance on how to meet each Level A and AA standard. Each principal is listed as the top node. Each guideline is nest underneath and each SC is nested under the applicable guideline. The bullets under each SC provide information on compliance based on the type of technology in use. The format is as follows:

X. Principal
  X.Y Guideline
    X.Y.Z Success Criterion (SC)
- Compliance technique
- Compliance technique
- …

Compliance techniques fall into two categories:

- General compliance techniques – for a variety of web, software, and document technologies
- Technology specific compliance techniques – for specific web, software, and document technologies

Techniques are categorized using alpha codes:

- G – general technique
- ARIA – accessible rich internet applications technique
- C – CSS technique
- FLASH – Flash technique
- H – HTML technique
- PDF – portable document format technique
- SCR – script, often JavaScript, technique
- SL – Silverlight technique
- T – text document technique

This information is also available on the W3C website as the quick reference guide How to Meet WCAG 2. Compliance techniques are provided along with advisable techniques and information on common failures. All links in this section target the W3C.org website.

Note: Other compliance techniques are acceptable. These techniques and information are provided to help with compliance. The goal is compliance with the standard, not necessarily adherence to these techniques.

1. PERCEIVABLE

Information and user interface components must be presentable to users in ways they can perceive.

1.1 TEXT ALTERNATIVES

Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.

1.1.1 NON-TEXT CONTENT

All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below.

- **Controls, Input:** If non-text content is a control or accepts user input, then it has a name that describes its purpose. (Refer to Guideline 4.1 for additional requirements for controls and content that accepts user input.)
- **Time-Based Media**: If non-text content is time-based media, then text alternatives at least provide descriptive identification of the non-text content. (Refer to Guideline 1.2 for additional requirements for media.)
- **Test**: If non-text content is a test or exercise that would be invalid if presented in text, then text alternatives at least provide descriptive identification of the non-text content.
- **Sensory**: If non-text content is primarily intended to create a specific sensory experience, then text alternatives at least provide descriptive identification of the non-text content.
- **CAPTCHA**: If the purpose of non-text content is to confirm that content is being accessed by a person rather than a computer, then text alternatives that identify and describe the purpose of the non-text content are provided, and alternative forms of CAPTCHA using output modes for different types of sensory perception are provided to accommodate different disabilities.
- **Decoration, Formatting, Invisible**: If non-text content is pure decoration, is used only for visual formatting, or is not presented to users, then it is implemented in a way that it can be ignored by assistive technology.

UNDERSTANDING

The intent of this Success Criterion is to make information conveyed by non-text content accessible through the use of a text alternative. Text alternatives are a primary way for making information accessible because they can be rendered through any sensory modality (for example, visual, auditory or tactile) to match the needs of the user. Providing text alternatives allows the information to be rendered in a variety of ways by a variety of user agents. For example, a person who cannot see a picture can have the text alternative read aloud using synthesized speech. A person who cannot hear an audio file can have the text alternative displayed so that he or she can read it. In the future, text alternatives will also allow information to be more easily translated into sign language or into a simpler form of the same language.

Non-text content can take a number of forms, and this Success Criterion specifies how each is to be handled.

For non-text content that is not covered by one of the other situations in the techniques listed below, such as charts, diagrams, audio recordings, pictures, and animations, text alternatives can make the same information available in a form that can be rendered through any modality (for example, visual, auditory or tactile). Short and long text alternatives can be used as needed to convey the information in the non-text content. Note that **prerecorded audio-only** and **prerecorded video-only** files are covered here. **Live-audio-only** and **Live-video-only** files are covered below (see 3rd paragraph following this one).

For non-text content that is a control or accepts user input, such as images used as submit buttons, image maps or complex animations, a name is provided to describe the purpose of the non-text content so that the person at least knows what the non-text content is and why it is there.

Non-text content that is time-based media is made accessible through 1.2: Time-Based Media. However, it is important that users know what it is when they encounter it on a page so they can decide what action if any they want to take with it. A text alternative that describes the time-based media and/or gives its title is therefore provided.

For Live Audio-only and live video-only content, it can be much more difficult to provide text alternatives that provide equivalent information as live audio-only and live video-only content. For these types of non-text content, text alternatives provide a descriptive label.

Sometimes a test or exercise must be partially or completely presented in non-text format. Audio or visual information is provided that cannot be changed to text because the test or exercise must be conducted using that sense. For example, a hearing test would be invalid if a text alternative were provided. A visual skill development exercise would similarly make no sense in text form. And a spelling test with text alternatives would not be very effective. For these cases, text alternatives should be provided to describe the purpose of the non-text content; of course, the text alternatives would not provide the same information needed to pass the test.

Sometimes content is primarily intended to create a specific sensory experience that words cannot fully capture. Examples include a symphony performance, works of visual art etc. For such content, text alternatives at least identify the non-text content with a descriptive label and where possible, additional descriptive text. If the reason for including the content in the page is known and can be described it is helpful to include that information.

Sometimes there are non-text exercises that are used to prove you are human. To avoid spam robots and other software from gaining access to a site a device called a CAPTCHA is used. These usually involve visual or auditory tasks that are beyond the current capabilities of Web robots. Providing a text alternative to them would however make them operable by Robots, thus defeating their purpose. In this case a text alternative would describe the purpose of the CAPTCHA, and alternate forms using different modalities would be provided to address the needs of people with different disabilities.

Sometimes there is non-text content that really is not meant to be seen or understood by the user. Transparent images used to move text over on a page; an invisible image that is used to track usage statistics; and a swirl in the corner that conveys no information but just fills up a blank space to create an aesthetic effect are all examples of this. Putting alternative text on such items just distracts people using screen readers from the content on the page. Not marking the content in any way, though, leaves users guessing what the non-text content is and what information they may have missed (even though they have not missed anything in reality). This type of non-text content, therefore, is marked or implemented in a way that assistive technologies (AT) will ignore it and not present anything to the user.

CAPTCHA - CAPTCHAs are a controversial topic in the accessibility community. As is described in the paper Inaccessibility of CAPTCHA, CAPTCHAs intrinsically push the edges of human abilities in an attempt to defeat automated processes. Every type of CAPTCHA will be unsolvable by users with certain disabilities. However, they are widely used, and the Web Content Accessibility Guidelines Working Group believes that if CAPTCHAs were forbidden outright, Web sites would choose not to conform to WCAG rather than abandon CAPTCHA. This would create barriers for a great many more users with disabilities. For this reason the Working Group has chosen to structure the requirement about CAPTCHA in a way that meets the needs of most people with disabilities, yet is also considered adoptable by sites. Requiring two different forms of CAPTCHA on a given site ensures that most people with disabilities will find a form they can use.

Because some users with disabilities will still not be able to access sites that meet the minimum requirements, the Working Group provides recommendations for additional steps. Organizations motivated to conform to WCAG should be aware of the importance of this topic and should go as far beyond the minimum requirements of the guidelines as possible. Additional recommended steps include:

- Providing more than two modalities of CAPTCHAs
- Providing access to a human customer service representative who can bypass CAPTCHA
- Not requiring CAPTCHAs for authorized users

BENEFITS

- This Success Criterion helps people who have difficulty perceiving visual content. Assistive technology can read text aloud, present it visually, or convert it to braille.
- Text alternatives may help some people who have difficulty understanding the meaning of photographs, drawings, and other images (e.g., line drawings, graphic designs, paintings, three-dimensional representations), graphs, charts, animations, etc.
- People who are deaf, are hard of hearing, or who are having trouble understanding audio information for any reason can read the text presentation. Research is ongoing regarding automatic translation of text into sign language.
- People who are deaf-blind can read the text in braille.
- Additionally, text alternatives support the ability to search for non-text content and to repurpose content in a variety of ways.

EXAMPLES

1. **A data chart**
   A bar chart compares how many widgets were sold in June, July, and August. The short label says, "Figure one - Sales in June, July and August." The longer description identifies the type of chart, provides a high-level summary of the data, trends and implications comparable to those available from the chart. Where possible and practical, the actual data is provided in a table.
2. **An audio recording of a speech**
   The link to an audio clip says, "Chairman's speech to the assembly." A link to a text transcript is provided immediately after the link to the audio clip.
3. **An animation that illustrates how a car engine works**
   An animation shows how a car engine works. There is no audio and the animation is part of a tutorial that describes how an engine works. Since the text of the tutorial already provides a full explanation, the image is an alternative for text and the text alternative includes only a brief description of the animation and refers to the tutorial text for more information.
4. **A traffic Web camera**
   A Web site allows users to select from a variety of Web cameras positioned throughout a major city. After a camera is selected, the image updates every two minutes. A short text alternative identifies the Web camera as "traffic Web camera." The site also provides a table of travel times for each of the routes covered by the Web cameras. The table is also updated every two minutes.
5. **A photograph of an historic event in a news story**
   A photograph of two world leaders shaking hands accompanies a news story about an international summit meeting. The text alternative says, "President X of Country X shakes hands with Prime Minister Y of country Y."
6. **A photograph of a historic event in content discussing diplomatic relationships**
   The same image is used in a different context intended to explain nuances in diplomatic encounters. The image of the president shaking hands with the prime minister appears on a Web site discussing intricate diplomatic relationships. The first text alternative reads, "President X of country X shakes hands with Prime Minister Y of country Y on January 2, 2009." An additional text alternative describes the room where the leaders are standing as well as the expressions on the leaders' faces, and identifies the other people in the room. The additional description might be included on the same page as the photograph or in a separate file associated with the image through a link or other standard programmatic mechanism.
7. **An audio recording of a press conference**
   A Web page includes a link to an audio recording of a press conference. The link text identifies the audio recording. The page also links to a text transcript of the press conference. The transcript includes a verbatim record of everything the speakers say. It identifies who is speaking as well as noting other

significant sounds that are part of the recording, such as applause, laughter, questions from the audience, and so on.

8. **An e-learning application**
   An e-learning application uses sound effects to indicate whether or not the answers are correct. The chime sound indicates that the answer is correct and the beep sound indicates that the answer is incorrect. A text description is also included so that people who can't hear or understand the sound understand whether the answer is correct or incorrect.
9. **A linked thumbnail image**
   A thumbnail image of the front page of a newspaper links to the home page of the "Smallville Times". The text alternative says "Smallville Times".
10. **The same image used on different sites**
    Different alternatives for an image of the world: An image of the world that is used on a travel site as a link to the International Travel section has the text alternative "International Travel". The same image is used as a link on a university Web site with the text alternative "International Campuses".
11. **An image map**
    An image of a building floor plan is interactive, allowing the user to select a particular room and navigate to a page containing information about that room. The short text alternative describes the image and its interactive purpose: "Building floor plan. Select a room for more information."

RELATED RESOURCES

Resources are for information purposes only, no endorsement implied.

- Excerpts from the NBA Tape Recording Manual, Third Edition
- Inaccessibility of CAPTCHA
- All That Malarkey: Accessible Alternatives
- 456 Berea Street: The Alt and Title Attributes

TECHNIQUES

Sufficient Techniques

Situation A: If a short description can serve the same purpose and present the same information as the non-text content:

- G94: Providing short text alternative for non-text content that serves the same purpose and presents the same information as the non-text content *using one of the techniques below:*
  - ARIA6: Using aria-label to provide labels for objects
  - ARIA10: Using aria-labelledby to provide a text alternative for non-text content
  - G196: Using a text alternative on one item within a group of images that describes all items in the group
  - FLASH1: Setting the name property for a non-text object
  - FLASH5: Combining adjacent image and text buttons for the same resource
  - FLASH28: Providing text alternatives for ASCII art, emoticons, and leetspeak in Flash
  - H2: Combining adjacent image and text links for the same resource
  - H35: Providing text alternatives on applet elements
  - H37: Using alt attributes on img elements
  - H53: Using the body of the object element

- H86: Providing text alternatives for ASCII art, emoticons, and leetspeak
- PDF1: Applying text alternatives to images with the Alt entry in PDF documents
- SL5: Defining a Focusable Image Class for Silverlight

Situation B: If a short description cannot serve the same purpose and present the same information as the non-text content (e.g., a chart or diagram):

- G95: Providing short text alternatives that provide a brief description of the non-text content *using one of the techniques below:*

  Short text alternative techniques for Situation B:

  - ARIA6: Using aria-label to provide labels for objects
  - ARIA10: Using aria-labelledby to provide a text alternative for non-text content
  - G196: Using a text alternative on one item within a group of images that describes all items in the group
  - FLASH1: Setting the name property for a non-text object
  - FLASH5: Combining adjacent image and text buttons for the same resource
  - FLASH28: Providing text alternatives for ASCII art, emoticons, and leetspeak in Flash
  - H2: Combining adjacent image and text links for the same resource
  - H35: Providing text alternatives on applet elements
  - H37: Using alt attributes on img elements
  - H53: Using the body of the object element
  - H86: Providing text alternatives for ASCII art, emoticons, and leetspeak
  - PDF1: Applying text alternatives to images with the Alt entry in PDF documents
  - SL5: Defining a Focusable Image Class for Silverlight

  Long text alternative techniques for Situation B:

  - ARIA15: Using aria-describedby to provide descriptions of images
  - G73: Providing a long description in another location with a link to it that is immediately adjacent to the non-text content
  - G74: Providing a long description in text near the non-text content, with a reference to the location of the long description in the short description
  - G92: Providing long description for non-text content that serves the same purpose and presents the same information
  - FLASH2: Setting the description property for a non-text object in Flash
  - FLASH11: Providing a longer text description of an object
  - H45: Using longdesc
  - H53: Using the body of the object element
  - SL8: Displaying HelpText in Silverlight User Interfaces

Situation C: If non-text content is a control or accepts user input:

- G82: Providing a text alternative that identifies the purpose of the non-text content *using one of the following techniques:*

  - ARIA6: Using aria-label to provide labels for objects
  - ARIA9: Using aria-labelledby to concatenate a label from several text nodes
  - FLASH6: Creating accessible hotspots using invisible buttons

- FLASH25: Labeling a form control by setting its accessible name
- FLASH27: Providing button labels that describe the purpose of a button
- FLASH29: Setting the label property for form components
- FLASH30: Specifying accessible names for image buttons
- FLASH32: Using auto labeling to associate text labels with form controls
- H24: Providing text alternatives for the area elements of image maps
- H30: Providing link text that describes the purpose of a link for anchor elements
- H36: Using alt attributes on images used as submit buttons
- H44: Using label elements to associate text labels with form controls
- H65: Using the title attribute to identify form controls when the label element cannot be used
- SL18: Providing Text Equivalent for Nontext Silverlight Controls With AutomationProperties.Name
- SL26: Using LabeledBy to Associate Labels and Targets in Silverlight
- SL30: Using Silverlight Control Compositing and AutomationProperties.Name

Situation D: If non-text content is time-based media (including live video-only and live audio-only); a test or exercise that would be invalid if presented in text; or primarily intended to create a specific sensory experience:

- Providing a descriptive label *using one of the techniques below*
- Providing a descriptive label that describes the purpose of live audio-only and live video-only content using one of techniques below
- Providing the accepted name or a descriptive name of the non-text content using one of the techniques below

Short text alternative techniques for Situation D:

- ARIA6: Using aria-label to provide labels for objects
- ARIA10: Using aria-labelledby to provide a text alternative for non-text content
- G196: Using a text alternative on one item within a group of images that describes all items in the group
- FLASH1: Setting the name property for a non-text object
- FLASH5: Combining adjacent image and text buttons for the same resource
- FLASH28: Providing text alternatives for ASCII art, emoticons, and leetspeak in Flash
- H2: Combining adjacent image and text links for the same resource
- H35: Providing text alternatives on applet elements
- H37: Using alt attributes on img elements
- H53: Using the body of the object element
- H86: Providing text alternatives for ASCII art, emoticons, and leetspeak
- PDF1: Applying text alternatives to images with the Alt entry in PDF documents
- SL5: Defining a Focusable Image Class for Silverlight

Situation E: If non-text content is a CAPTCHA:

- G143: Providing a text alternative that describes the purpose of the CAPTCHA
  AND
  G144: Ensuring that the Web Page contains another CAPTCHA serving the same purpose using a different modality

Situation F: If the non-text content should be ignored by assistive technology

Implementing or marking the non-text content so that it will be ignored by assistive technology using one of the following techniques:

- C9: Using CSS to include decorative images
- FLASH3: Marking objects in Flash so that they can be ignored by AT
- H67: Using null alt text and no title attribute on img elements for images that AT should ignore
- PDF4: Hiding decorative images with the Artifact tag in PDF documents

Advisory Techniques

General Techniques for Informative Non-Text Content:

- Identifying informative non-text content
- Keeping short descriptions short
- Describing images that include text
- Providing a longer description of the non-text content where only a descriptive label is required using a technology-specific technique (for an accessibility-supported content technology) for long description listed above
- Providing different sizes for non-text content when it cannot have an equivalent accessible alternative
- Using server-side scripts to resize images of text

General Techniques for Live Non-Text Content

- Linking to textual information that provides comparable information (e.g., for a traffic Webcam, a municipality could provide a link to the text traffic report)

General techniques to minimize the barrier of CAPTCHAs

- Providing more than two modalities of CAPTCHAs
- Providing access to a human customer service representative who can bypass CAPTCHA
- Not requiring CAPTCHAs for authorized users

HTML Techniques

- H46: Using noembed with embed
- Writing for browsers that do not support frame (future link)
- Providing alternative content for iframe (future link)
- Not using long descriptions for iframe (future link)
- Providing redundant text links for client-side image maps (future link)

CSS Techniques

- C18: Using CSS margin and padding rules instead of spacer images for layout design
- Using CSS background, :before or :after rules for decorative images instead of img elements
- Displaying empty table cells

WAI-ARIA Techniques

- Using the ARIA presentation role to indicate elements are purely presentational

Silverlight Techniques

- SL19: Providing User Instructions With AutomationProperties.HelpText in Silverlight

Metadata Techniques

- Using metadata to associate text transcriptions with a video
- Using metadata to associate text transcriptions with audio-only content *using one of the following techniques:*
    - EXAMPLE: Providing, in metadata, URI(s) that points to an audio description and a text transcript of a video.
    - EXAMPLE: Providing, in metadata, URI(s) that point to several text transcripts (English, French, Dutch) of an audio file.

FAILURES

- F3: Failure of Success Criterion 1.1.1 due to using CSS to include images that convey important information
- F13: Failure of Success Criterion 1.1.1 and 1.4.1 due to having a text alternative that does not include information that is conveyed by color differences in the image
- F20: Failure of Success Criterion 1.1.1 and 4.1.2 due to not updating text alternatives when changes to non-text content occur
- F30: Failure of Success Criterion 1.1.1 and 1.2.1 due to using text alternatives that are not alternatives (e.g., filenames or placeholder text)
- F38: Failure of Success Criterion 1.1.1 due to not marking up decorative images in HTML in a way that allows assistive technology to ignore them
- F39: Failure of Success Criterion 1.1.1 due to providing a text alternative that is not null (e.g., alt="spacer" or alt="image") for images that should be ignored by assistive technology
- F65: Failure of Success Criterion 1.1.1 due to omitting the alt attribute or text alternative on img elements, area elements, and input elements of type "image"
- F67: Failure of Success Criterion 1.1.1 and 1.2.1 due to providing long descriptions for non-text content that does not serve the same purpose or does not present the same information
- F71: Failure of Success Criterion 1.1.1 due to using text look-alikes to represent text without providing a text alternative
- F72: Failure of Success Criterion 1.1.1 due to using ASCII art without providing a text alternative

1.2 TIME-BASED MEDIA

Provide alternatives for time-based media.

1.2.1 AUDIO-ONLY AND VIDEO-ONLY (PRERECORDED)

For prerecorded audio-only and prerecorded video-only media, the following are true, except when the audio or video is a media alternative for text and is clearly labeled as such:

- **Prerecorded Audio-only:** An alternative for time-based media is provided that presents equivalent information for prerecorded audio-only content.
- **Prerecorded Video-only:** Either an alternative for time-based media or an audio track is provided that presents equivalent information for prerecorded video-only content.

UNDERSTANDING

The intent of this Success Criterion is to make information conveyed by prerecorded audio-only and prerecorded video-only content available to all users. Alternatives for time-based media that are text based make information accessible because text can be rendered through any sensory modality (for example, visual, auditory or tactile) to match the needs of the user. In the future, text could also be translated into symbols, sign language or simpler forms of the language (future).

An example of pre-recorded video with no audio information or user interaction is a silent movie. The purpose of the transcript is to provide an equivalent to what is presented visually. For prerecorded video content, authors have the option to provide an audio track. The purpose of the audio alternative is to be an equivalent to the video. This makes it possible for users with and without vision impairment to review content simultaneously. The approach can also make it easier for those with cognitive, language and learning disabilities to understand the content because it would provide parallel presentation.

Note: A text equivalent is not required for audio that is provided as an equivalent for video with no audio information. For example, it is not required to caption video description that is provided as an alternative to a silent movie.

See also: 1.2.4 – Audio Only (Live)

BENEFITS

- This Success Criterion helps people who have difficulty perceiving visual content. Assistive technology can read text alternatives aloud, present them visually, or convert them to braille.
- Alternatives for timed-based media that are text based may help some people who have difficulty understanding the meaning of prerecorded video content.
- People who are deaf, are hard of hearing, or who are having trouble understanding audio information for any reason can read the text presentation. Research is ongoing regarding automatic translation of text into sign language.
- People who are deaf-blind can read the text in braille.
- Additionally, text supports the ability to search for non-text content and to repurpose content in a variety of ways.

EXAMPLES

- **An audio recording of a speech**
  The link to an audio clip says, "Chairman's speech to the assembly." A link to a text transcript is provided immediately after the link to the audio clip.
- **An audio recording of a press conference**
  A Web page includes a link to an audio recording of a press conference that identifies the audio recording. The page also links to a text transcript of the press conference. The transcript includes a verbatim record of everything the speakers say. It identifies who is speaking as well as noting other significant sounds that are part of the recording, such as applause, laughter, questions from the audience, and so on.
- **An animation that illustrates how a car engine works**
  An animation shows how a car engine works. There is no audio and the animation is part of a tutorial that describes how an engine works. Since the text of the tutorial already provides a full explanation, the media is an alternative for text and the text alternative includes only a brief description of the animation and refers to the tutorial text for more information.
- **A video-only file with an audio track**
  A silent movie includes an audio track which includes a description of the action in the video.

RELATED RESOURCES

- uiAccess list of transcription services
- Transcripts on the Web: Getting people to your podcasts and videos

TECHNIQUES

Sufficient Techniques

Situation A: If the content is pre-recorded audio only

- G158: Providing an alternative for time-based media for audio-only content
- SL17: Providing Static Alternative Content for Silverlight Media Playing in a MediaElement

Situation B: If the content is pre-recorded video only

- G159: Providing an alternative for time-based media for video-only content
- G166: Providing audio that describes the important video content and describing it as such
- SL17: Providing Static Alternative Content for Silverlight Media Playing in a MediaElement

Advisory Techniques

- H96: Using the track element to provide audio descriptions
- Providing a transcript of a live audio only presentation after the fact
- Linking to textual information that provides comparable information (e.g., for a traffic Webcam, a municipality could provide a link to the text traffic report.)

Failures

- F30: Failure of Success Criterion 1.1.1 and 1.2.1 due to using text alternatives that are not alternatives (e.g., filenames or placeholder text)
- F67: Failure of Success Criterion 1.1.1 and 1.2.1 due to providing long descriptions for non-text content that does not serve the same purpose or does not present the same information

1.2.2 CAPTIONS (PRERECORDED)

Captions are provided for all prerecorded audio content in synchronized media, except when the media is a media alternative for text and is clearly labeled as such.

UNDERSTANDING

The intent of this SC is to enable people who are deaf or hard of hearing to watch synchronized media presentations. Captions provide the part of the content available via the audio track. Captions not only include dialogue, but identify who is speaking and include non-speech information conveyed through sound, including meaningful sound effects.

It is acknowledged that at the present time there may be difficulty in creating captions for time-sensitive material and this may result in the author being faced with the choice of delaying the information until captions are available, or publishing time-sensitive content that is inaccessible to the deaf, at least for the interval until captions are available. Over time, the tools for captioning as well as building the captioning into the delivery process can shorten or eliminate such delays.

Captions are not needed when the synchronized media is, itself, an alternate presentation of information that is also presented via text on the Web page. For example, if information on a page is accompanied by a synchronized media presentation that presents no more information than is already presented in text, but is easier for people with cognitive, language, or learning disabilities to understand, then it would not need to be captioned since the information is already presented on the page in text or in text alternatives (e.g., for images).

See also 1.2.1: Captions (Live).

BENEFITS

- People who are deaf or have a hearing loss can access the auditory information in the synchronized media content through captions.

EXAMPLES

- **A captioned tutorial**
  A video clip shows how to tie a knot. The captions read,
  "(music)
  Using rope to tie knots was an important skill
  for the likes of sailors, soldiers and woodsmen.."
  From Sample Transcript Formatting by Whit Anderson

- A complex legal document contains synchronized media clips for different paragraphs that show a person speaking the contents of the paragraph. Each clip is associated with its corresponding paragraph. No captions are provided for the synchronized media.
- An instruction manual containing a description of a part and its necessary orientation is accompanied by a synchronized media clip showing the part in its correct orientation. No captions are provided for the synchronized media clip.
- An orchestra provides captions for videos of performances. In addition to capturing dialog and lyrics verbatim, captions identify non-vocal music by title, movement, composer, and any information that will help the user comprehend the nature of the audio. For instance captions read,
  "[Orchestral Suite No. 3.2 in D major, BWV 1068, Air]
  [Johann Sebastian Bach, Composer]
  ♪ Calm melody with a slow tempo ♪"

RELATED RESOURCES

Guides to Captioning

- Captioning Key: Guidelines and Preferred Techniques
- Best Practices in Online Captioning

SMIL Resources

- Synchronized Multimedia Integration Language (SMIL 3.0)
- Accessibility Features of SMIL

Other Captioning Resources

- National Center for Accessible Media
- WebAIM: Captioning Resource List

TECHNIQUES

Sufficient Techniques

- G93: Providing open (always visible) captions
- G87: Providing closed captions
- G87: Providing closed captions *using one of the following techniques:*
  - SM11: Providing captions through synchronized text streams in SMIL 1.0
  - SM12: Providing captions through synchronized text streams in SMIL 2.0
  - H95: Using the track element to provide captions
  - FLASH9: Applying captions to prerecorded synchronized media
  - SL16: Providing Script-Embedded Text Captions for MediaElement Content
  - SL28: Using Separate Text-Format Text Captions for MediaElement Content

Advisory Techniques

- Providing a note saying "No sound is used in this clip" for video-only clips
- Using SMIL 1.0 to provide captions for all languages for which there are audio tracks
- Using SMIL 2.0 to provide captions for all languages for which there are audio tracks

Failures

- F8: Failure of Success Criterion 1.2.2 due to captions omitting some dialogue or important sound effects
- F74: Failure of Success Criterion 1.2.2 and 1.2.8 due to not labeling a synchronized media alternative to text as an alternative

- F75: Failure of Success Criterion 1.2.2 by providing synchronized media without captions when the synchronized media presents more information than is presented on the page

1.2.3 AUDIO DESCRIPTION OR MEDIA ALTERNATIVE (PRERECORDED)

An alternative for time-based media or audio description of the prerecorded video content is provided for synchronized media, except when the media is a media alternative for text and is clearly labeled as such.

UNDERSTANDING

The intent of this Success Criterion is to provide people who are blind or visually impaired access to the visual information in a synchronized media presentation. This Success Criterion describes two approaches, either of which can be used.

One approach is to provide audio description of the video content. The audio description augments the audio portion of the presentation with the information needed when the video portion is not available. During existing pauses in dialogue, audio description provides information about actions, characters, scene changes, and on-screen text that are important and are not described or spoken in the main sound track.

The second approach involves providing all of the information in the synchronized media (both visual and auditory) in text form. An alternative for time-based media provides a running description of all that is going on in the synchronized media content. The alternative for time-based media reads something like a screenplay or book. Unlike audio description, the description of the video portion is not constrained to just the pauses in the existing dialogue. Full descriptions are provided of all visual information, including visual context, actions and expressions of actors, and any other visual material. In addition, non-speech sounds (laughter, off-screen voices, etc.) are described, and transcripts of all dialogue are included. The sequence of description and dialogue transcripts are the same as the sequence in the synchronized media itself. As a result, the alternative for time-based media can provide a much more complete representation of the synchronized media content than audio description alone.

If there is any interaction as part of the synchronized media presentation (e.g., "press now to answer the question") then the alternative for time-based media would provide hyperlinks or whatever is needed to provide the same functionality.

For 1.2.3, 1.2.5, and 1.2.7, if all of the information in the video track is already provided in the audio track, no audio description is necessary.

1.2.3, 1.2.5, and 1.2.8 overlap somewhat with each other. This is to give the author some choice at the minimum conformance level, and to provide additional requirements at higher levels. At Level A in Success Criterion 1.2.3, authors do have the choice of providing either an audio description or a full text alternative. If they wish to conform at Level AA, under Success Criterion 1.2.5 authors must provide an audio description - a requirement already met if they chose that alternative for 1.2.3, otherwise an additional requirement. At Level AAA under Success Criterion 1.2.8 they must provide an extended text description. This is an additional requirement if both 1.2.3 and 1.2.5 were met by providing an audio description only. If 1.2.3 was met, however, by providing a text description, and the 1.2.5 requirement for an audio description was met, then 1.2.8 does not add new requirements.

See also 1.2.2: Audio Description (Prerecorded), 1.2.2: Extended Audio Description (Prerecorded) and 1.2.3: Media Alternative (Prerecorded).

BENEFITS

- This Success Criterion may help some people who have difficulty watching video or other synchronized media content, including people who have difficulty perceiving or understanding moving images.

EXAMPLES

- **A movie with audio description.**
  Describer: A title, "Teaching Evolution Case Studies. Bonnie Chen." A teacher shows photographs of birds with long, thin beaks.
  Bonnie Chen: "These photos were all taken at the Everglades."
  Describer: The teacher hands each student two flat, thin wooden sticks.
  Bonnie Chen: "Today you will pretend to be a species of wading bird that has a beak like this."
  Describer: The teacher holds two of the sticks to her mouth making the shape of a beak.
  Transcript of audio based on the first few minutes of " Teaching Evolution Case Studies, Bonnie Chen" (copyright WGBH and Clear Blue Sky Productions, Inc.)
- **An alternative for time-based media for a training video**
  A company purchases a Training video for use by its employees and puts it on the company's intranet. The video involves explaining use of a new technology and has a person talking and showing things at the same time. Since there is no place to insert audio description of the visual demonstrations during gaps in dialogue, the company provides an alternative for time-based media that all employees, including those who cannot see the demonstrations, can use to better understand what is being presented.

RELATED RESOURCES

- Standard Techniques in Audio Description
- Synchronized Multimedia Integration Language (SMIL) 3.0
- Accessibility Features of SMIL

TECHNIQUES

Sufficient Techniques

- Providing an alternative for time-based media including any interaction using one of the following techniques
  - Placing a link to the transcript immediately next to the non-text content
  - Providing Static Alternative Content for Silverlight Media Playing in a MediaElement
- Linking to the alternative for time-based media using one of the following techniques
  - Using the body of the object element
- G78: Providing a second, user-selectable, audio track that includes audio descriptions
- G78 **AND** SL1
- Providing a version of a movie with audio descriptions using one of the following:
  - Providing audio description in SMIL 1.0
  - Providing audio description in SMIL 2.0
  - Applying audio descriptions to Flash video
  - SL1
  - Using any player that supports audio and video
- Providing a movie with extended audio descriptions using one of the following:
  - Adding extended audio description in SMIL 1.0

    - Adding extended audio description in SMIL 2.0
    - Applying audio descriptions to Flash video
    - SL1
    - Using any player that supports audio and video
- G203

Advisory Techniques

- H96

1.2.4 CAPTIONS (LIVE)

Captions are provided for all live audio content in synchronized media.

UNDERSTANDING

The intent of this Success Criterion is to enable people who are deaf or hard of hearing to watch real-time presentations. Captions provide the part of the content available via the audio track. Captions not only include dialogue, but also identify who is speaking and notate sound effects and other significant audio.

This success criterion was intended to apply to broadcast of synchronized media and is not intended to require that two-way multimedia calls between two or more individuals through web apps must be captioned regardless of the needs of users. Responsibility for providing captions would fall to the content providers (the callers) or the "host" caller, and not the application.

BENEFITS

- People who are deaf or have a hearing loss can access the auditory information in the synchronized media content through captions.

EXAMPLES

- **A Web cast**
  A news organization provides a live, captioned Web cast.
- **A music Web cast**
  An orchestra provides Communication Access Realtime Translation (CART) captioning of each real-time Web performance. The CART service captures lyrics and dialog as well as identifies non-vocal music by title, movement, composer, and any information that will help the user comprehend the nature of the audio.

RELATED RESOURCES

- See 1.2.2 Captions (Pre-recorded)

TECHNIQUES

Sufficient Techniques

- G9: Creating captions for live synchronized media **AND** G93: Providing open (always visible) captions
- G9: Creating captions for live synchronized media **AND** G87: Providing closed captions

- G9: Creating captions for live synchronized media **AND** G87: Providing closed captions using one of the following techniques:
    - SM11: Providing captions through synchronized text streams in SMIL 1.0
    - SM12: Providing captions through synchronized text streams in SMIL 2.0

1.2.5 AUDIO DESCRIPTION (PRERECORDED)

Audio description is provided for all prerecorded video content in synchronized media.

UNDERSTANDING

The intent of this Success Criterion is to provide people who are blind or visually impaired access to the visual information in a synchronized media presentation. The audio description augments the audio portion of the presentation with the information needed when the video portion is not available. During existing pauses in dialogue, audio description provides information about actions, characters, scene changes, and on-screen text that are important and are not described or spoken in the main sound track.

For 1.2.3, 1.2.5, and 1.2.7, if all of the information in the video track is already provided in the audio track, no audio description is necessary.

1.2.3, 1.2.5, and 1.2.8 overlap somewhat with each other. This is to give the author some choice at the minimum conformance level, and to provide additional requirements at higher levels. At Level A in Success Criterion 1.2.3, authors do have the choice of providing either an audio description or a full text alternative. If they wish to conform at Level AA, under Success Criterion 1.2.5 authors must provide an audio description - a requirement already met if they chose that alternative for 1.2.3, otherwise an additional requirement. At Level AAA under Success Criterion 1.2.8 they must provide an extended text description. This is an additional requirement if both 1.2.3 and 1.2.5 were met by providing an audio description only. If 1.2.3 was met, however, by providing a text description, and the 1.2.5 requirement for an audio description was met, then 1.2.8 does not add new requirements.

BENEFITS

- People who are blind or have low vision as well as those with cognitive limitations who have difficulty interpreting visually what is happening benefit from audio description of visual information.

EXAMPLES

- **A movie with audio description.**
  Describer: A title, "Teaching Evolution Case Studies. Bonnie Chen." A teacher shows photographs of birds with long, thin beaks.
  Bonnie Chen: "These photos were all taken at the Everglades."
  Describer: The teacher hands each student two flat, thin wooden sticks.
  Bonnie Chen: "Today you will pretend to be a species of wading bird that has a beak like this."
  Describer: The teacher holds two of the sticks to her mouth making the shape of a beak.
  Transcript of audio based on the first few minutes of " Teaching Evolution Case Studies, Bonnie Chen" (copyright WGBH and Clear Blue Sky Productions, Inc.)

RELATED RESOURCES

- Standard Techniques in Audio Description
- Synchronized Multimedia Integration Language (SMIL) 3.0
- Accessibility Features of SMIL

TECHNIQUES

Sufficient Techniques

- G78: Providing a second, user-selectable, audio track that includes audio descriptions
- G78: Providing a second, user-selectable, audio track that includes audio descriptions **AND** SL1: Accessing Alternate Audio Tracks in Silverlight Media
- G173: Providing a version of a movie with audio descriptions *using one of the following techniques:*
    - SM6: Providing audio description in SMIL 1.0
    - SM7: Providing audio description in SMIL 2.0
    - FLASH26: Applying audio descriptions to Flash video
    - Using any player that supports audio and video (future link)
- G8: Providing a movie with extended audio descriptions *using one of the following techniques:*
    - SM1: Adding extended audio description in SMIL 1.0
    - SM2: Adding extended audio description in SMIL 2.0
    - FLASH26: Applying audio descriptions to Flash video
    - Using any player that supports audio and video (future link)
- G203: Using a static text alternative to describe a talking head video

Advisory Techniques

- H96: Using the track element to provide audio descriptions
- Synchronized Multimedia Integration Language (SMIL) 3.0

1.3 ADAPTABLE

Create content that can be presented in different ways (for example simpler layout) without losing information or structure.

1.3.1 INFO AND RELATIONSHIPS

Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text.

UNDERSTANDING

This SC is intended to preserve information and relationships implied by visual or auditory formatting when the presentation format changes. For example, the presentation format changes when the content is read by a screen reader or when a user style sheet is substituted for the style sheet provided by the author.

Sighted users perceive structure and relationships through various visual cues — headings are often in a larger, bold font separated from paragraphs by blank lines; list items are preceded by a bullet and perhaps indented; paragraphs are separated by a blank line; items that share a common characteristic are organized into tabular rows and columns; form fields may be positioned as groups that share text labels; a different background color may be used to indicate that several items are related to each other; words that have special status are indicated by changing the font family and /or bolding, italicizing, or underlining them; items that share a common characteristic are organized into a table where the relationship of cells sharing the same row or column and the relationship of each cell to its row and/or column header are necessary for understanding; and so on. Having these structures and these relationships programmatically determined or available in text ensures that information important for comprehension will be perceivable to all.

Auditory cues may be used as well. For example, a chime might indicate the beginning of a new section; a change in voice pitch or speech rate may be used to emphasize important information or to indicate quoted text; etc.

When such relationships are perceivable to one set of users, those relationships can be made to be perceivable to all. One method of determining whether or not information has been properly provided to all users is to access the information serially in different modalities.

If links to glossary items are implemented using anchor elements (or the proper link element for the technology in use) and identified using a different font face, a screen reader user will hear that the item is a link when the glossary term is encountered even though they may not receive information about the change in font face. An on-line catalog may indicate prices using a larger font colored red. A screen reader or person who cannot perceive red, still has the information about the price as long as it is preceded by the currency symbol.

Some technologies do not provide a means to programmatically determine some types of information and relationships. In that case then there should be a text description of the information and relationships. For instance, "all required fields are marked with an asterisk (*)". The text description should be near the information it is describing (when the page is linearized), such as in the parent element or in the adjacent element.

There may also be cases where it may be a judgment call as to whether the relationships should be programmatically determined or be presented in text. However, when technologies support programmatic relationships, it is strongly encouraged that information and relationships be programmatically determined rather than described in text.

Note: It is not required that color values be programmatically determined. The information conveyed by color cannot be adequately presented simply by exposing the value. Therefore, Success Criterion 1.4.1 addresses the specific case of color, rather than Success Criterion 1.3.1.

BENEFITS

- This Success Criterion helps people with different disabilities by allowing user agents to adapt content according to the needs of individual users.
- Users who are blind (using a screen reader) benefit when information conveyed through color is also available in text (including text alternatives for images that use color to convey information).
- Users who are deaf-blind using braille (text) refreshable displays may be unable to access color-dependent information.

EXAMPLES

- **A form with required fields**
  A form contains several required fields. The labels for the required fields are displayed in red. In addition, at the end of each label is an asterisk character, *. The instructions for completing the form indicate that "all required fields are displayed in red and marked with an asterisk *", followed by an example.
- **A form that uses color and text to indicate required fields**
  A form contains both required and optional fields. Instructions at the top of the form explain that required fields are labeled with red text and also with an icon whose text alternative says, "Required." Both the red text and the icon are programmatically associated with the appropriate form fields so that assistive technology users can determine the required fields.
- **A bus schedule table where the headers for each cell can be programmatically determined**
  A bus schedule consists of a table with the bus stops listed vertically in the first column and the different buses listed horizontally across the first row. Each cell contains the time when the bus will be at that bus stop. The bus stop and bus cells are identified as headers for their corresponding row or column so that assistive technology can programmatically determine which bus and which bus stop are associated with the time in each cell.
- **A form where the labels for the checkboxes can be programmatically determined**
  In a form, the labels for each checkbox can be programmatically determined by assistive technology.
- **A text document**
  A simple text document is formatted with double blank lines before titles, asterisks to indicate list items and other standard formatting conventions so that its structure can be programmatically determined.

RELATED RESOURCES

- WebAIM: Semantic Structure
- Heading Tags

TECHNIQUES

Sufficient Techniques

Situation A: The technology provides semantic structure to make information and relationships conveyed through presentation programmatically determinable

- ARIA11: Using ARIA landmarks to identify regions of a page

- ARIA12: Using role=heading to identify headings
- ARIA13: Using aria-labelledby to name regions and landmarks
- ARIA16: Using aria-labelledby to provide a name for user interface controls
- ARIA17: Using grouping roles to identify related form controls
- ARIA20: Using the region role to identify a region of the page
- G115: Using semantic elements to mark up structure **AND** H49: Using semantic markup to mark emphasized or special text
- G117: Using text to convey information that is conveyed by variations in presentation of text
- G140: Separating information and structure from presentation to enable different presentations
- Making information and relationships conveyed through presentation programmatically determinable using the following techniques: (future link) *using one of the following techniques:*
    - G138: Using semantic markup whenever color cues are used
    - H51: Using table markup to present tabular information
    - PDF6: Using table elements for table markup in PDF Documents
    - PDF20: Using Adobe Acrobat Pro's Table Editor to repair mistagged tables
    - H39: Using caption elements to associate data table captions with data tables
    - FLASH31: Specifying caption text for a DataGrid
    - H73: Using the summary attribute of the table element to give an overview of data tables
    - FLASH23: Adding summary information to a DataGrid
    - H63: Using the scope attribute to associate header cells and data cells in data tables
    - H43: Using id and headers attributes to associate data cells with header cells in data tables
    - FLASH21: Using the DataGrid component to associate column headers with cells
    - H44: Using label elements to associate text labels with form controls
    - H65: Using the title attribute to identify form controls when the label element cannot be used
    - PDF10: Providing labels for interactive form controls in PDF documents
    - PDF12: Providing name, role, value information for form fields in PDF documents
    - FLASH32: Using auto labeling to associate text labels with form controls
    - FLASH29: Setting the label property for form components
    - FLASH25: Labeling a form control by setting its accessible name
    - H71: Providing a description for groups of form controls using fieldset and legend elements
    - SL20: Relying on Silverlight AutomationPeer Behavior to Set AutomationProperties.Name
    - SL26: Using LabeledBy to Associate Labels and Targets in Silverlight
    - H85: Using OPTGROUP to group OPTION elements inside a SELECT
    - H48: Using ol, ul and dl for lists or groups of links
    - H42: Using h1-h6 to identify headings
    - PDF9: Providing headings by marking content with heading tags in PDF documents
    - SCR21: Using functions of the Document Object Model (DOM) to add content to a page
    - PDF11: Providing links and link text using the Link annotation and the /Link structure element in PDF documents
    - PDF17: Specifying consistent page numbering for PDF documents
    - PDF21: Using List tags for lists in PDF documents
    - H97: Grouping related links using the nav element

Situation B: The technology in use does NOT provide the semantic structure to make the information and relationships conveyed through presentation programmatically determinable

- G117: Using text to convey information that is conveyed by variations in presentation of text

- FLASH8: Adding a group name to the accessible name of a form control
- Making information and relationships conveyed through presentation programmatically determinable or available in text using the following techniques: (future link) *using one of the following techniques:*
    - T1: Using standard text formatting conventions for paragraphs
    - T2: Using standard text formatting conventions for lists
    - T3: Using standard text formatting conventions for headings

Advisory Techniques

- C22: Using CSS to control visual presentation of text
- Using CSS rather than tables for page layout (future link)
- G162: Positioning labels to maximize predictability of relationships
- ARIA1: Using the aria-describedby property to provide a descriptive label for user interface controls
- ARIA2: Identifying a required field with the aria-required property
- Providing labels for all form controls that do not have implicit labels (future link)
- G141: Organizing a page using headings

Failures

- F2: Failure of Success Criterion 1.3.1 due to using changes in text presentation to convey information without using the appropriate markup or text
- F33: Failure of Success Criterion 1.3.1 and 1.3.2 due to using white space characters to create multiple columns in plain text content
- F34: Failure of Success Criterion 1.3.1 and 1.3.2 due to using white space characters to format tables in plain text content
- F42: Failure of Success Criteria 1.3.1, 2.1.1, 2.1.3, or 4.1.2 when emulating links
- F43: Failure of Success Criterion 1.3.1 due to using structural markup in a way that does not represent relationships in the content
- F46: Failure of Success Criterion 1.3.1 due to using th elements, caption elements, or non-empty summary attributes in layout tables
- F48: Failure of Success Criterion 1.3.1 due to using the pre element to markup tabular information
- F87: Failure of Success Criterion 1.3.1 due to inserting non-decorative content by using :before and :after pseudo-elements and the 'content' property in CSS
- F90: Failure of Success Criterion 1.3.1 for incorrectly associating table headers and content via the headers and id attributes
- F91: Failure of Success Criterion 1.3.1 for not correctly marking up table headers
- F92: Failure of Success Criterion 1.3.1 due to the use of role presentation on content which conveys semantic information

1.3.2 MEANINGFUL SEQUENCE

When the sequence in which content is presented affects its meaning, a correct reading sequence can be programmatically determined.

UNDERSTANDING

The intent of this Success Criterion is to enable a user agent to provide an alternative presentation of content while preserving the reading order needed to understand the meaning. It is important that it be possible to programmatically determine at least one sequence of the content that makes sense. Content that does not meet this Success Criterion may confuse or disorient users when assistive technology reads the content in the wrong order, or when alternate style sheets or other formatting changes are applied.

A sequence is meaningful if the order of content in the sequence cannot be changed without affecting its meaning. For example, if a page contains two independent articles, the relative order of the articles may not affect their meaning, as long as they are not interleaved. In such a situation, the articles themselves may have meaningful sequence, but the container that contains the articles may not have a meaningful sequence.

The semantics of some elements define whether or not their content is a meaningful sequence. For instance, in HTML, text is always a meaningful sequence. Tables and ordered lists are meaningful sequences, but unordered lists are not.

The order of content in a sequence is not always meaningful. For example, the relative order of the main section of a Web page and a navigation section does not affect their meaning. They could occur in either order in the programmatically determined reading sequence. As another example, a magazine article contains several callout sidebars. The order of the article and the sidebars does not affect their meaning. In these cases there are a number of different reading orders for a Web page that can satisfy the Success Criterion.

For clarity:

1. Providing a particular linear order is only required where it affects meaning.
2. There may be more than one order that is "correct" (according to the WCAG 2.0 definition).
3. Only one correct order needs to be provided.

BENEFITS

- This Success Criterion may help people who rely on assistive technologies that read content aloud. The meaning evident in the sequencing of the information in the default presentation will be the same when the content is presented in spoken form.

EXAMPLES

- **Example 1:** In a multi-column document, the linear presentation of the content flows from the top of a column to the bottom of the column, then to the top of the next column.
- **Example 2:** CSS is used to position a navigation bar, the main story on a page, and a side story. The visual presentation of the sections does not match the programmatically determined order, but the meaning of the page does not depend on the order of the sections.

TECHNIQUES

Sufficient Techniques

- G57: Ordering the content in a meaningful sequence
- Marking sequences in the content as meaningful **AND** G57: Ordering the content in a meaningful sequence *using one of the following techniques:*
  - H34: Using a Unicode right-to-left mark (RLM) or left-to-right mark (LRM) to mix text direction inline
  - H56: Using the dir attribute on an inline element to resolve problems with nested directional runs

  - C6: Positioning content based on structural markup
  - C8: Using CSS letter-spacing to control spacing within a word
- C27: Making the DOM order match the visual order
- FLASH15: Using the tabIndex property to specify a logical reading order and a logical tab order in Flash
- PDF3: Ensuring correct tab and reading order in PDF documents
- SL34: Using the Silverlight Default Tab Sequence and Altering Tab Sequences With Properties

Advisory Techniques

- Using left-justified text for languages that are written left to right and right-justified text for languages that are written right-to-left (future link)
- Providing a link to linearized rendering (future link)
- Providing a style switcher between style sheets that affect presentation order (future link)

Failures

- F1: Failure of Success Criterion 1.3.2 due to changing the meaning of content by positioning information with CSS
- F32: Failure of Success Criterion 1.3.2 due to using white space characters to control spacing within a word
- F33: Failure of Success Criterion 1.3.1 and 1.3.2 due to using white space characters to create multiple columns in plain text content
- F34: Failure of Success Criterion 1.3.1 and 1.3.2 due to using white space characters to format tables in plain text content
- F49: Failure of Success Criterion 1.3.2 due to using an HTML layout table that does not make sense when linearized

1.3.3 SENSORY CHARACTERISTICS

Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, size, visual location, orientation, or sound.

Note: For requirements related to color, refer to Guideline 1.4.

UNDERSTANDING

The intent of this Success Criterion is to ensure that all users can access instructions for using the content, even when they cannot perceive shape or size or use information about spatial location or orientation. Some content relies on knowledge of the shape or position of objects that are not available from the structure of the content (for example, "round button" or "button to the right"). Some users with disabilities are not able to perceive shape or position due to the nature of the assistive technologies they use. This Success Criterion requires that additional information be provided to clarify anything that is dependent on this kind of information.

Providing information using shape and/or location, however, is an effective method for many users including those with cognitive limitations. This provision should not discourage those types of cues as long as the information is also provided in other ways.

In some languages, it is commonly understood that "above" refers to the content previous to that point in the content and "below" refers to the content after that point. In such languages, if the content being referenced is in the appropriate place in the reading order and the references are unambiguous, statements such as "choose one of the links below" or "all of the above" would conform to this Success Criterion.

WCAG was designed to apply only to controls that were displayed on a web page. The intent was to avoid describing controls solely via references to visual or auditory cues. When applying this to instructions for operating physical hardware controls (e.g. a web kiosk with dedicated content), tactile cues on the hardware might be described (e.g. the arrow shaped key, the round key on the right side). This success criterion is not intended to prevent the use of tactile cues in instructions.

BENEFITS

- People who are blind and people who have low vision may not be able to understand information if it is conveyed by shape and/or location. Providing additional information other than shape and/or location will allow them to understand the information conveyed by shape and/or alone.

EXAMPLES

- **Example 1: A schedule of competitive events uses color and shape to distinguish the time of each event**
  A table presents a list of times across the top row and a list of events in the first vertical column. The cell corresponding to the time of a particular event has a specific background color and diamond shaped glyph so it can be identified by color and shape.
- **Example 2: An on-line multi-page survey**
  An on-line multi-page survey uses a link implemented as a green arrow icon placed in the lower right hand corner of the content to move from one survey page to the next. The arrow is clearly labeled with "Next" and the instructions state, "To move to the next section of the survey, select the green arrow icon labeled 'Next' in the lower right corner below the last survey question." This example uses both positioning, color and labeling to help identify the icon.

TECHNIQUES

Sufficient Techniques

- G96: Providing textual identification of items that otherwise rely only on sensory information to be understood

Advisory Techniques

- Using an image with a text alternative for graphical symbols instead of a Unicode font glyph with the desired graphical appearance but different meaning

Failures

- F14: Failure of Success Criterion 1.3.3 due to identifying content only by its shape or location
- F26: Failure of Success Criterion 1.3.3 due to using a graphical symbol alone to convey information

1.4 DISTINGUISHABLE

Make it easier for users to see and hear content including separating foreground from background.

1.4.1 USE OF COLOR

Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.

Note: This success criterion addresses color perception specifically. Other forms of perception are covered in Guideline 1.3 including programmatic access to color and other visual presentation coding.

UNDERSTANDING

The intent of this Success Criterion is to ensure that all users can access information that is conveyed by color differences, that is, by the use of color where each color has a meaning assigned to it. If the information is conveyed through color differences in an image (or other non-text format), the color may not be seen by users with color deficiencies. In this case, providing the information conveyed with color through another visual means ensures users who cannot see color can still perceive the information.

Color is an important asset in design of Web content, enhancing its aesthetic appeal, its usability, and its accessibility. However, some users have difficulty perceiving color. People with partial sight often experience limited color vision, and many older users do not see color well. In addition, people using text-only, limited-color or monochrome displays and browsers will be unable to access information that is presented only in color.

Examples of information conveyed by color differences: “required fields are red", “error is shown in red", and “Mary's sales are in red, Tom's are in blue". Examples of indications of an action include: using color to indicate that a link will open in a new window or that a database entry has been updated successfully. An example of prompting a response would be: using highlighting on form fields to indicate that a required field had been left blank.

This should not in any way discourage the use of color on a page, or even color coding if it is redundant with other visual indication.

BENEFITS

- Users with partial sight often experience limited color vision.
- Some older users may not be able to see color well.
- Users who have color-blindness benefit when information conveyed by color is available in other visual ways.
- People using text-only, limited color, or monochrome displays may be unable to access color-dependent information.
- Users who have problems distinguishing between colors can look or listen for text cues.
- People using Braille displays or other tactile interfaces can detect text cues by touch.

EXAMPLES

- **A form that uses color and text to indicate required fields**
  A form contains both required and optional fields. Instructions at the top of the form explain that required fields are labeled with red text and also with an icon whose text alternative says, "Required." Both the red text

and the icon are programmatically associated with the appropriate form fields so that assistive technology users can determine the required fields.

- **An examination.**
  Students view an SVG image of a chemical compound and identify the chemical elements present based on the colors and numbers used in the diagram. The text alternatives associated with each element name the color of the element and indicate the element's position in the diagram. Students who cannot perceive color have the same information about the compound as their classmates. (This technique also satisfies Guideline 1.1 Level A.)
- **Disabled Form elements.**
  Form elements which are disabled via markup or scripting, are greyed out and made inactive by the user agent. When in the disabled state these elements do not receive focus. Assistive technologies can programmatically determine the state of disabled elements and will provide this information to the user as the elements are encountered on the page. The change in color and loss of focus provides redundant, visual information about the state of the control.

RELATED RESOURCES

- Vischeck
- AWARE Color Laboratory
- Wikipedia: Color Blindness
- Causes of Color: How do people inherit colorblindness? How often?: Genetics
- How to make figures and presentations that are friendly to Colorblind people

TECHNIQUES

Sufficient Techniques

Situation A: If the color of particular words, backgrounds, or other content is used to indicate information:

- G14: Ensuring that information conveyed by color differences is also available in text
- G205: Including a text cue for colored form control labels
- G182: Ensuring that additional visual cues are available when text color differences are used to convey information
- G183: Using a contrast ratio of 3:1 with surrounding text and providing additional visual cues on focus for links or controls where color alone is used to identify them

Situation B: If color is used within an image to convey information:

- G111: Using color and pattern
- G14: Ensuring that information conveyed by color differences is also available in text

Advisory Techniques

- Conveying information redundantly using color
- C15: Using CSS to change the presentation of a user interface component when it receives focus

Failures

- F13: Failure of Success Criterion 1.1.1 and 1.4.1 due to having a text alternative that does not include information that is conveyed by color differences in the image
- F73: Failure of Success Criterion 1.4.1 due to creating links that are not visually evident without color vision
- F81: Failure of Success Criterion 1.4.1 due to identifying required or error fields using color differences only

1.4.2 AUDIO CONTROL

If any audio on a Web page plays automatically for more than 3 seconds, either a mechanism is available to pause or stop the audio, or a mechanism is available to control audio volume independently from the overall system volume level.

Note: Since any content that does not meet this success criterion can interfere with a user's ability to use the whole page, all content on the Web page (whether or not it is used to meet other success criteria) must meet this success criterion. See Conformance Requirement 5: Non-Interference.

UNDERSTANDING

Individuals who use screen reading software can find it hard to hear the speech output if there is other audio playing at the same time. This difficulty is exacerbated when the screen reader's speech output is software based (as most are today) and is controlled via the same volume control as the sound. Therefore, it is important that the user be able to turn off the background sound. Note: Having control of the volume includes being able to reduce its volume to zero.

Playing audio automatically when landing on a page may affect a screen reader user's ability to find the mechanism to stop it because they navigate by listening and automatically started sounds might interfere with that navigation. Therefore, we discourage the practice of automatically starting sounds (especially if they last more than 3 seconds), and encourage that the sound be started by an action initiated by the user after they reach the page, rather than requiring that the sound be stopped by an action of the user after they land on the page.

See also 1.4.2: Low or No Background Audio.

BENEFITS

- Individuals who use screen reading technologies can hear the screen reader without other sounds playing. This is especially important for those who are hard of hearing and for those whose screen readers use the system volume (so they cannot turn sound down and screen reader up).
- This Success Criterion also benefits people who have difficulty focusing on visual content (including text) when audio is playing.

EXAMPLES

- An audio file begins playing automatically when a page is opened. However, the audio can be stopped by the user by selecting a "silent" link at the top of the page.
- A Flash splash page with sound that plays and then stops in less than 3 seconds.

- A Flash splash page with sound that plays automatically includes a control at the top that allows users to turn the sound off.

TECHNIQUES

Sufficient Techniques

- G60: Playing a sound that turns off automatically within three seconds
- G170: Providing a control near the beginning of the Web page that turns off sounds that play automatically
- G171: Playing sounds only on user request
- SL24: Using AutoPlay to Keep Silverlight Media from Playing Automatically
- FLASH18: Providing a control to turn off sounds that play automatically in Flash
- FLASH34: Turning off sounds that play automatically when an assistive technology is detected
- SL3: Controlling Silverlight MediaElement Audio Volume

Advisory Techniques

- Providing a site-wide preference to turn off audio in addition to providing a control near the top of the Web page that turns off sounds that play automatically

Failures

- F23: Failure of 1.4.2 due to playing a sound longer than 3 seconds where there is no mechanism to turn it off
- F93: Failure of Success Criterion 1.4.2 for absence of a way to pause or stop an HTML5 media element that autoplays

1.4.3 CONTRAST (MINIMUM)

The visual presentation of text and images of text has a contrast ratio of at least 4.5:1, except for the following:

- **Large Text:** Large-scale text and images of large-scale text have a contrast ratio of at least 3:1;
- **Incidental:** Text or images of text that are part of an inactive user interface component, that are pure decoration, that are not visible to anyone, or that are part of a picture that contains significant other visual content, have no contrast requirement.
- **Logotypes:** Text that is part of a logo or brand name has no minimum contrast requirement.

UNDERSTANDING

The intent of this Success Criterion is to provide enough contrast between text and its background so that it can be read by people with moderately low vision (who do not use contrast-enhancing assistive technology). For people without color deficiencies, hue and saturation have minimal or no effect on legibility as assessed by reading performance (Knoblauch et al., 1991). Color deficiencies can affect luminance contrast somewhat. Therefore, in the recommendation, the contrast is calculated in such a way that color is not a key factor so that people who have a color vision deficit will also have adequate contrast between the text and the background.

Text that is decorative and conveys no information is excluded. For example, if random words are used to create a background and the words could be rearranged or substituted without changing meaning, then it would be decorative and would not need to meet this criterion.

Text that is larger and has wider character strokes is easier to read at lower contrast. The contrast requirement for larger text is therefore lower. This allows authors to use a wider range of color choices for large text, which is helpful for design of pages, particularly titles. 18 point text or 14 point bold text is judged to be large enough to require a lower contrast ratio. (See The American Printing House for the Blind Guidelines for Large Printing and The Library of Congress Guidelines for Large Print under Resources). "18 point" and "bold" can both have different meanings in different fonts but, except for very thin or unusual fonts, they should be sufficient. Since there are so many different fonts, the general measures are used and a note regarding fancy or thin fonts is included.

When evaluating this success criterion, the font size in points should be obtained from the user agent or calculated on font metrics in the way that user agents do. Point sizes are based on the CSS pt size as defined in CSS3 Values. The ratio between sizes in points and CSS pixels is 1pt = 1.333px, therefore 14pt and 18pt are equivalent to approximately 18.5px and 24px.

Because different image editing applications default to different pixel densities (e.g. 72 PPI or 96 PPI), specifying point sizes for fonts from within an image editing application can be unreliable when it comes to presenting text at a specific size. When creating images of large-scale text, authors should ensure that the text in the resulting image is roughly equivalent to 1.2 and 1.5 em or to 120% or 150% of the default size for body text. For example, for a 72 PPI image, an author would need to use approximately 19 pt and 24 pt font sizes in order to successfully present images of large-scale text to a user.

The previously-mentioned contrast requirements for text also apply to images of text (text that has been rendered into pixels and then stored in an image format) as stated in Success Criterion 1.4.3.

This requirement applies to situations in which images of text were intended to be understood as text. Incidental text, such as in photographs that happen to include a street sign, are not included. Nor is text that for some reason is designed to be invisible to all viewers. Stylized text, such as in corporate logos, should be treated in terms of its function on the page, which may or may not warrant including the content in the text alternative. Corporate visual guidelines beyond logo and logotype are not included in the exception.

In this provision there is an exception that reads "that are part of a picture that contains significant other visual content,". This exception is intended to separate pictures that have text in them from images of text that are done to replace text in order to get a particular look.

Images of text do not scale as well as text because they tend to pixelate. It is also harder to change foreground and background contrast and color combinations for images of text, which is necessary for some users. Therefore, we suggest using text wherever possible, and when not, consider supplying an image of higher resolution.

The minimum contrast success criterion (1.4.3) applies to text in the page, including placeholder text and text that is shown when a pointer is hovering over an object or when an object has keyboard focus. If any of these are used in a page, the text needs to provide sufficient contrast.

Although this Success Criterion only applies to text, similar issues occur for content presented in charts, graphs, diagrams, and other non-text-based information. Content presented in this manner should also have good contrast to ensure that more users can access the information.

See also 1.4.1: Contrast (Enhanced).

Rationale for the Ratios Chosen

Text that is larger and has wider character strokes is easier to read at lower contrast. The contrast requirement for larger text is therefore lower. This allows authors to use a wider range of color choices for large text, which is helpful for design of pages, particularly titles. 18 point text or 14 point bold text is judged to be large enough to require a lower contrast ratio. (See The American Printing House for the Blind Guidelines for Large Printing and The Library of Congress Guidelines for Large Print under Resources). "18 point" and "bold" can both have different meanings in different fonts but, except for very thin or unusual fonts, they should be sufficient. Since there are so many different fonts, the general measures are used and a note regarding fancy or thin fonts is included.

When evaluating this success criterion, the font size in points should be obtained from the user agent or calculated on font metrics in the way that user agents do. Point sizes are based on the CSS pt size as defined in CSS3 Values. The ratio between sizes in points and CSS pixels is 1pt = 1.333px, therefore 14pt and 18pt are equivalent to approximately 18.5px and 24px.

Because different image editing applications default to different pixel densities (e.g. 72 PPI or 96 PPI), specifying point sizes for fonts from within an image editing application can be unreliable when it comes to presenting text at a specific size. When creating images of large-scale text, authors should ensure that the text in the resulting image is roughly equivalent to 1.2 and 1.5 em or to 120% or 150% of the default size for body text as rendered by the browser.

Hues are perceived differently by users with color vision deficiencies (both congenital and acquired) resulting in different colors and relative luminance contrasts than for normally sighted users. Because of this, effective contrast and readability are different for this population. However, color deficiencies are so diverse that prescribing effective general use color pairs (for contrast) based on quantitative data is not feasible. Requiring good luminance contrast accommodates this by requiring contrast that is independent of color perception. Fortunately, most of the luminance contribution is from the mid and long wave receptors which largely overlap in their spectral responses. The result is that effective luminance contrast can generally be computed without regard to specific color deficiency, except for the use of predominantly long wavelength colors against darker colors (generally appearing black) for those who have protanopia. (We provide an advisory technique on avoiding red on black for that reason). For more information see [[ARDITI-KNOBLAUCH-1994]] [[ARDITI-KNOBLAUCH-1996]] [[ARDITI]].

Some people with cognitive disabilities require color combinations or hues that have low contrast, and therefore we allow and encourage authors to provide mechanisms to adjust the foreground and background colors of the content. Some of the combinations that could be chosen may have contrast levels that will be lower than those found in the Success Criteria. This is not a violation of this Success Criteria provided there is a mechanism that will return to the default values set out in the Success Criteria.

A contrast ratio of 3:1 is the minimum level recommended by [[ISO-9241-3]] and [[ANSI-HFES-100-1988]] for standard text and vision. The 4.5:1 ratio is used in this provision to account for the loss in contrast that results from moderately low visual acuity, congenital or acquired color deficiencies, or the loss of contrast sensitivity that typically accompanies aging.

The rationale is based on a) adoption of the 3:1 contrast ratio for minimum acceptable contrast for normal observers, in the ANSI standard, and b) the empirical finding that in the population, visual acuity of 20/40 is associated with a contrast sensitivity loss of roughly 1.5 [[ARDITI-FAYE]]. A user with 20/40 would thus require a contrast ratio of 3 * 1.5 = 4.5 to 1. Following analogous empirical findings and the same logic, the user with 20/80 visual acuity would require contrast of about 7:1.

The contrast ratio of 4.5:1 was chosen for level AA because it compensated for the loss in contrast sensitivity usually experienced by users with vision loss equivalent to approximately 20/40 vision. (20/40 calculates to approximately 4.5:1.) 20/40 is commonly reported as typical visual acuity of elders at roughly age 80. [[GITTINGS-FOZARD]]

The contrast ratio of 7:1 was chosen for level AAA because it compensated for the loss in contrast sensitivity usually experienced by users with vision loss equivalent to approximately 20/80 vision. People with more than this degree of vision loss usually use assistive technologies to access their content (and the assistive technologies usually have contrast enhancing, as well as magnification capability built into them). The 7:1 level therefore generally provides compensation for the loss in contrast sensitivity experienced by users with low vision who do not use assistive technology and provides contrast enhancement for color deficiency as well.

Calculations in [[ISO-9241-3]] and [[ANSI-HFES-100-1988]] are for body text. A relaxed contrast ratio is provided for text that is much larger.

Notes on Formula

Conversion from nonlinear to linear RGB values is based on IEC/4WD 61966-2-1 [[IEC-4WD]] and on "A Standard Default Color Space for the Internet - sRGB" [[sRGB]].

The formula (L1/L2) for contrast is based on [[ISO-9241-3]] and [[ANSI-HFES-100-1988]] standards.

The ANSI/HFS 100-1988 standard calls for the contribution from ambient light to be included in the calculation of L1 and L2. The .05 value used is based on Typical Viewing Flare from [[IEC-4WD]] and the [[sRGB]] paper by M. Stokes et al.

This Success Criterion and its definitions use the terms "contrast ratio" and "relative luminance" rather than "luminance" to reflect the fact that Web content does not emit light itself. The contrast ratio gives a measure of the relative luminance that would result when displayed. (Because it is a ratio, it is dimensionless.)

Refer to related resources for a list of tools that utilize the contrast ratio to analyze the contrast of Web content.

See also 2.4.3: Focus Visible for techniques for indicating keyboard focus.

It is sometimes helpful for authors to not specify colors for certain sections of a page in order to help users who need to view content with specific color combinations to view the content in their preferred color scheme. Refer to 1.4.3: Images of Text for more information.

BENEFITS

- People with low vision often have difficulty reading text that does not contrast with its background. This can be exacerbated if the person has a color vision deficiency that lowers the contrast even further. Providing a minimum luminance contrast ratio between the text and its background can make the text more readable even if the person does not see the full range of colors. It also works for the rare individuals who see no color.

RELATED RESOURCES

- Contrast Analyser – Application
- Contrast Ratio Analyser - online service
- Colour Contrast Analyser - Firefox Extension
- Colour Contrast Check
- Contrast Ratio Calculator
- Atypical colour response
- Colors On the Web Color Contrast Analyzer
- Tool to convert images based on color loss so that contrast is restored as luminance contrast when there was only color contrast (that was lost due to color deficiency)
- List of color contrast tools

TECHNIQUES

Sufficient Techniques

Situation A: Text is less than 18 point if not bold and less than 14 point if bold

- G18: Ensuring that a contrast ratio of at least 4.5:1 exists between text (and images of text) and background behind the text
- G148: Not specifying background color, not specifying text color, and not using technology features that change those defaults
- G174: Providing a control with a sufficient contrast ratio that allows users to switch to a presentation that uses sufficient contrast
- SL13: Providing A Style Switcher To Switch To High Contrast

Situation B: Text is at least 18 point if not bold and less than 14 point if bold

- G145: Ensuring that a contrast ratio of at least 3:1 exists between text (and images of text) and background behind the text
- G148: Not specifying background color, not specifying text color, and not using technology features that change those defaults
- G174: Providing a control with a sufficient contrast ratio that allows users to switch to a presentation that uses sufficient contrast
- SL13: Providing A Style Switcher To Switch To High Contrast

Advisory Techniques

- G156: Using a technology that has commonly-available user agents that can change the foreground and background of blocks of text
- Using a higher contrast value for text that is over a patterned background (future link)
- Using Unicode text and style sheets instead of images of text (future link)
- Using a higher contrast values for lines in diagrams (future link)
- Using greater contrast level for red-black text/background combinations (future link)
- Using colors that are composed predominantly of mid spectral components for the light and spectral extremes (blue and red wavelengths) for the dark (future link)
- Using a light pastel background rather than a white background behind black text to create sufficient but not extreme contrast (future link)
- Making icons using simple line drawings that meet the contrast provisions for text (future link)
- Providing sufficient contrast ratio in graphs and charts (future link)
- Using a 3:1 contrast ratio or higher as the default presentation (future link)

- Providing sufficient color contrast for empty text fields (future link)

Failures

- F24: Failure of Success Criterion 1.4.3, 1.4.6 and 1.4.8 due to specifying foreground colors without specifying background colors or vice versa
- F83: Failure of Success Criterion 1.4.3 and 1.4.6 due to using background images that do not provide sufficient contrast with foreground text (or images of text)

1.4.4 RESIZE TEXT

Except for captions and images of text, text can be resized without assistive technology up to 200 percent without loss of content or functionality.

UNDERSTANDING

The intent of this Success Criterion is to ensure that visually rendered text, including text-based controls (text characters that have been displayed so that they can be seen [vs. text characters that are still in data form such as ASCII]) can be scaled successfully so that it can be read directly by people with mild visual disabilities, without requiring the use of assistive technology such as a screen magnifier. Users may benefit from scaling all content on the Web page, but text is most critical.

The scaling of content is primarily a user agent responsibility. User agents that satisfy UAAG 1.0 Checkpoint 4.1 allow users to configure text scale. The author's responsibility is to create Web content that does not prevent the user agent from scaling the content effectively. Authors may satisfy this Success Criterion by verifying that content does not interfere with user agent support for resizing text, including text-based controls, or by providing direct support for resizing text or changing the layout. An example of direct support might be via server-side script that can be used to assign different style sheets.

The author cannot rely on the user agent to satisfy this Success Criterion for HTML content if users do not have access to a user agent with zoom support. For example, if they work in an environment that requires them to use IE 6.

If the author is using a technology whose user agents do not provide zoom support, the author is responsible to provide this type of functionality directly or to provide content that works with the type of functionality provided by the user agent. If the user agent doesn't provide zoom functionality but does let the the user change the text size, the author is responsible for ensuring that the content remains usable when the text is resized.

Some user interface components that function as a label and require activation by the user to access content are not wide enough to accommodate the label's content. For example, in Web mail applications the subject column may not be wide enough to accommodate every possible subject header, but activating the subject header takes the user to the full message with the full subject header. In Web-based spreadsheets, cell content that is too long to be displayed in a column can be truncated, and the full content of the cell is available to the user when the cell receives focus. The content of a user interface component may also become too wide in user interfaces where the user can resize the column width. In this type of user interface component, line wrapping is not required; truncation is acceptable if the component's full content is available on focus or after user activation and an indication that this information can be accessed, is provided to the user in some way besides the fact that it is truncated.

Content satisfies the Success Criterion if it can be scaled up to 200%, that is, up to twice the width and height. Authors may support scaling beyond that limit, however, as scaling becomes more extreme, adaptive layouts may introduce usability problems. For example, words may be too wide to fit into the horizontal space available to them, causing them to be truncated; layout constraints may cause text to overlap with other content when it is scaled larger; or only one word of a sentence may fit on each line, causing the sentence to be displayed as a vertical column of text that is difficult to read.

The working group feels that 200% is a reasonable accommodation that can support a wide range of designs and layouts, and complements older screen magnifiers that provide a minimum magnification of 200%. Above 200%, zoom (which resizes text, images, and layout regions and creates a larger canvas that may require both horizontal and vertical scrolling) may be more effective than text resizing. Assistive technology dedicated to zoom support would usually be used in such a situation and may provide better accessibility than attempts by the author to support the user directly.

Images of text do not scale as well as text because they tend to pixelate, and therefore we suggest using text wherever possible. It is also harder to change foreground and background contrast and color combinations for images of text, which are necessary for some users.

See also 1.4.3: Visual Presentation.

BENEFITS

- This Success Criterion helps people with low vision by letting them increase text size in content so that they can read it.

EXAMPLES

- A user with vision impairments increases the text size on a Web page in a browser from 1 em to 1.2 ems. While the user could not read the text at the smaller size, she can read the larger text. All the information on the page is still displayed when the larger font is used for the text.
- A Web page contains a control for changing the scale of the page. Selecting different settings changes the layout of the page to use the best design for that scale.
- A user uses a zoom function in his user agent to change the scale of the content. All the content scales uniformly, and the user agent provides scroll bars, if necessary.

RELATED RESOURCES

- CSS 2 Box Model
- CSS 2 Visual formatting Model
- CSS 2 Visual formatting Model Details
- About fluid and fixed width layouts
- Accessible CSS

TECHNIQUES

Sufficient Techniques

- G142: Using a technology that has commonly-available user agents that support zoom
- SL22: Supporting Browser Zoom in Silverlight
- SL23: Using A Style Switcher to Increase Font Size of Silverlight Text Elements
- Ensuring that text containers resize when the text resizes AND using measurements that are relative to other measurements in the content (future link) *using one of the following techniques:*
    - C28: Specifying the size of text containers using em units
    - Techniques for relative measurements (future link) *using one of the following techniques:*
        - C12: Using percent for font sizes
        - C13: Using named font sizes
        - C14: Using em units for font sizes
    - Techniques for text container resizing (future link) *using one of the following techniques:*
        - SCR34: Calculating size and position in a way that scales with text size
        - G146: Using liquid layout
- G178: Providing controls on the Web page that allow users to incrementally change the size of all text on the page up to 200 percent
- G179: Ensuring that there is no loss of content or functionality when the text resizes and text containers do not change their width

Advisory Techniques

- Providing large fonts by default (future link)
- Using page-percent for container sizes (future link)
- Avoiding scaling font sizes smaller than the user-agent default (future link)
    - NOTE: The author won't actually know the font size, but should avoid percentage scaling that results in less than 100%
- Avoiding justified text (future link)
- Providing sufficient inter-line and inter-column spacing (future link)
- Providing different sizes for non-text content when it cannot have an equivalent accessible alternative (future link)
- Avoiding the use of text in raster images (future link)
- Using server-side scripts to resize images of text (future link)
- C17: Scaling form elements which contain text
- Ensuring that text in raster images is at least 18pt (future link)
- Scaling text down to 50% (future link)
- C20: Using relative measurements to set column widths so that lines can average 80 characters or less when the browser is resized
- C22: Using CSS to control visual presentation of text
- Providing a mechanism to allow captions to be enlarged (future link)

Failures

- F69: Failure of Success Criterion 1.4.4 when resizing visually rendered text up to 200 percent causes the text, image or controls to be clipped, truncated or obscured
- F80: Failure of Success Criterion 1.4.4 when text-based form controls do not resize when visually rendered text is resized up to 200%

1.4.5 IMAGES OF TEXT

If the technologies being used can achieve the visual presentation, text is used to convey information rather than images of text except for the following:

- **Customizable:** The image of text can be visually customized to the user's requirements;
- **Essential:** A particular presentation of text is essential to the information being conveyed.

Note*:* Logotypes (text that is part of a logo or brand name) are considered essential.

UNDERSTANDING

The intent of this Success Criterion is to encourage authors, who are using technologies which are capable of achieving their desired default visual presentation, to enable people who require a particular visual presentation of text to be able to adjust the text presentation as needed. This includes people who require the text in a particular font size, foreground and background color, font family, line spacing or alignment.

If an author can use text to achieve the same visual effect, he or she should present the information as text rather than using an image. If for any reason, the author cannot format the text to get the same effect, the effect won't be reliably presented on the commonly available user agents, or using a technology to meet this criterion would interfere with meeting other criteria such as 1.4.4, then an image of text can be used. This includes instances where a particular presentation of text is essential to the information being conveyed, such as type samples, logotypes, branding, etc. Images of text may also be used in order to use a particular font that is either not widely deployed or which the author doesn't have the right to redistribute, or to ensure that the text would be anti-aliased on all user agents.

Images of text can also be used where it is possible for users to customize the image of text to match their requirements.

The definition of image of text contains the note: Note: This does not include text that is part of a picture that contains significant other visual content. Examples of such pictures include graphs, screenshots, and diagrams which visually convey important information through more than just text.

Techniques for satisfying this Success Criterion are the same as those for Success Criterion 1.4.9, except that they only need to apply if the visual presentation can be achieved with the technologies that the author is using. For Success Criterion 1.4.9, the sufficient techniques would be applied only when the user can customize the output.

See also 1.4.4: Images of Text (No Exception).

BENEFITS

- F69: Failure of Success Criterion 1.4.4 when resizing visually rendered text up to 200 percent causes the text, image or controls to be clipped, truncated or obscured
- F80: Failure of Success Criterion 1.4.4 when text-based form controls do not resize when visually rendered text is resized up to 200%

EXAMPLES

- **Styled Headings**

Rather than using bitmap images to present headings in a specific font and size, an author uses CSS to achieve the same result.

- **Dynamically Generated Images**
  A Web page uses server-side scripting to present text as an an image. The page includes controls that allow the user to adjust the font size and foreground and background colors of the generated image.
- **A quote**
  A Web page contains a quote. The quote itself is presented as italicized text, indented from the left margin. The name of the person to whom the quote is attributed is below the quote with 1.5x the line space and further indented from the left margin. CSS is used to position the text; set the spacing between lines; as well as display the text's font family, size, color and decoration.
- **Navigation items**
  A Web page contains a menu of navigation links that have both an icon and text to describe their target. CSS is used to display the text's font family, size and foreground and background colors; as well as the spacing between the navigation links.
- **A logo containing text**
  A Web site contains the organization's logo in the top left corner of each Web page. The logo contains logotype (text as part, or all, of the logo). The visual presentation of the text is essential to the identity of the logo and is included as a gif image which does not allow the text characteristics to be changed. The image has a text alternative.
- **Representation of a font family**
  A Web page contains information about a particular font family. Substituting the font family with another font would defeat the purpose of the representation. The representation is included as a jpeg image which does not allow the text characteristics to be changed. The image has a text alternative.
- **A representation of a letter**
  A Web page contains a representation of an original letter. The depiction of the letter in its original format is essential to information being conveyed about the time period in which it was written. The letter is included as a gif image which does not allow the text characteristics to be changed. The image has a text alternative.
- **Symbolic text characters**
  A form allows users to enter blocks of text. The form provides a number of buttons, including functions to style the text and check spelling. Some of the buttons use text characters that do not form a sequence that expresses something in human language. For example "B" to increase font weight, "I" to italicize the text and "ABC" to check the spelling. The symbolic text characters are included as gif images which do not allow the text characteristics to be changed. The buttons have text alternatives.
- **Customizable font settings in images of text**
  A Web site allows users to specify font settings and all images of text on the site are then provided based on those settings.

RELATED RESOURCES

- CSS Web fonts
- Weblog comments: WebKit now supports CSS @font-face rules
- Creating Cross Browser Compatible CSS Text Shadows
- CSS and text

TECHNIQUES

Sufficient Techniques

- C22: Using CSS to control visual presentation of text

- SL31: Using Silverlight Font Properties to Control Text Presentation
- C30: Using CSS to replace text with images of text and providing user interface controls to switch
- G140: Separating information and structure from presentation to enable different presentations
- PDF7: Performing OCR on a scanned PDF document to provide actual text

Advisory Techniques

General technique for non-text content

- Identifying informative non-text content

CSS Techniques

- C12: Using percent for font sizes
- C13: Using named font sizes
- C14: Using em units for font sizes
- C8: Using CSS letter-spacing to control spacing within a word
- C6: Positioning content based on structural markup
- Avoid applying text styling to text characters within a word

2. OPERABLE

User interface components and navigation must be operable.

2.1 – KEYBOARD ACCESSIBLE

Make all functionality available from a keyboard.

2.1.1 KEYBOARD

All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.

Note 1: This exception relates to the underlying function, not the input technique. For example, if using handwriting to enter text, the input technique (handwriting) requires path-dependent input but the underlying function (text input) does not.

Note 2: This does not forbid and should not discourage providing mouse input or other input methods in addition to keyboard operation.

UNDERSTANDING

The intent of this Success Criterion is to ensure that, wherever possible, content can be operated through a keyboard or keyboard interface (so an alternate keyboard can be used). When content can be operated through a keyboard or alternate keyboard, it is operable by people with no vision (who cannot use devices such as mice that require eye-hand coordination) as well as by people who must use alternate keyboards or input devices that act as keyboard emulators. Keyboard emulators include speech input software, sip-and-puff software, on-screen keyboards, scanning software and a variety of assistive technologies and alternate keyboards. Individuals with low vision also may have trouble tracking a pointer and find the use of software much easier (or only possible) if they can control it from the keyboard.

Examples of "specific timings for individual keystrokes" include situations where a user would be required to repeat or execute multiple keystrokes within a short period of time or where a key must be held down for an extended period before the keystroke is registered.

The phrase "except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints" is included to separate those things that cannot reasonably be controlled from a keyboard.

Most actions carried out by a pointing device can also be done from the keyboard (for example, clicking, selecting, moving, sizing). However, there is a small class of input that is done with a pointing device that cannot be done from the keyboard in any known fashion without requiring an inordinate number of keystrokes. Free hand drawing, watercolor painting, and flying a helicopter through an obstacle course are all examples of functions that require path dependent input. Drawing straight lines, regular geometric shapes, re-sizing windows and dragging objects to a location (when the path to that location is not relevant) do not require path dependent input.

The use of MouseKeys would not satisfy this Success Criterion because it is not a keyboard equivalent to the application; it is a mouse equivalent (i.e., it looks like a mouse to the application).

It is assumed that the design of user input features takes into account that operating system keyboard accessibility features may be in use. For example, modifier key locking may be turned on. Content continues to function in such an environment, not sending events that would collide with the modifier key lock to produce unexpected results.

BENEFITS

- People who are blind (who cannot use devices such as mice that require eye-hand coordination)
- People with low vision (who may have trouble finding or tracking a pointer indicator on screen)
- Some people with hand tremors find using a mouse very difficult and therefore usually use a keyboard

EXAMPLES

- **Example 1: A drawing Program.**
  A drawing program allows users to create, size, position and rotate objects from the keyboard.
- **Example 2: A drag and Drop Feature.**
  An application that uses drag and drop also supports "cut" and "paste" or form controls to move objects.
- **Example 3: Moving between and connecting discrete points.**
  A connect-the-dots program allows the user to move between dots on a screen and use the spacebar to connect the current dot to the previous one.
- **Example 4: Exception - Painting Program.**
  A watercolor painting program passes as an exception because the brush strokes vary depending on the speed and duration of the movements.
- **Example 5: Exception - Model helicopter flight training simulator.**
  A model helicopter flight training simulator passes as an exception because the nature of the simulator is to teach real-time behavior of a model helicopter.
- **Example 6: A PDA with an optional keyboard**
  A PDA device that is usually operated via a stylus has an optional keyboard that can be attached. The keyboard allows full Web browsing in standard fashion. The Web content is operable because it was designed to work with keyboard-only access.

TECHNIQUES

Sufficient Techniques

- G202: Ensuring keyboard control for all functionality
- Ensuring keyboard control *using one of the following techniques:*
  - H91: Using HTML form controls and links
  - PDF3: Ensuring correct tab and reading order in PDF documents
  - PDF11: Providing links and link text using the Link annotation and the /Link structure element in PDF documents
  - PDF23: Providing interactive form controls in PDF documents
  - SL15: Providing Keyboard Shortcuts that Work Across the Entire Silverlight Application
- G90: Providing keyboard-triggered event handlers *using one of the following techniques:*
  - SCR20: Using both keyboard and other device-specific functions
  - SCR35: Making actions keyboard accessible by using the onclick event of anchors and buttons
  - SCR2: Using redundant keyboard and mouse event handlers
  - SL9: Handling Key Events to Enable Keyboard Functionality in Silverlight
  - SL14: Providing Custom Control Key Handling for Keyboard Functionality in Silverlight
- FLASH17: Providing keyboard access to a Flash object and avoiding a keyboard trap *using one of the following techniques:*

  - FLASH22: Adding keyboard-accessible actions to static elements
  - FLASH16: Making actions keyboard accessible by using the click event on standard components
  - FLASH14: Using redundant keyboard and mouse event handlers in Flash

Advisory Techniques

- Using XHTML role, state, and value attributes if repurposing static elements as interactive user interface components (future link) AND SCR29: Adding keyboard-accessible actions to static HTML elements (Scripting)
- Providing keyboard shortcuts to important links and form controls (future link)
- Using unique letter combinations to begin each item of a list (future link)
- Choosing the most abstract event handler (Scripting) (future link)
- Using the onactivate event (Scripting) (future link)
- Avoiding use of common user-agent keyboard commands for other purposes (future link)

Failures

- F42: Failure of Success Criteria 1.3.1, 2.1.1, 2.1.3, or 4.1.2 when emulating links
- F54: Failure of Success Criterion 2.1.1 due to using only pointing-device-specific event handlers (including gesture) for a function
- F55: Failure of Success Criteria 2.1.1, 2.4.7, and 3.2.1 due to using script to remove focus when focus is received

2.1.2 – NO KEYBOARD TRAP

If keyboard focus can be moved to a component of the page using a keyboard interface, then focus can be moved away from that component using only a keyboard interface, and, if it requires more than unmodified arrow or tab keys or other standard exit methods, the user is advised of the method for moving focus away.

Note: Since any content that does not meet this success criterion can interfere with a user's ability to use the whole page, all content on the Web page (whether it is used to meet other success criteria or not) must meet this success criterion. See Conformance Requirement 5: Non-Interference.

UNDERSTANDING

The intent of this Success Criterion is to ensure that that content does not "trap" keyboard focus within subsections of content on a Web page. This is a common problem when multiple formats are combined within a page and rendered using plug-ins or embedded applications.

There may be times when the functionality of the Web page restricts the focus to a subsection of the content, as long as the user knows how to leave that state and "untrap" the focus.

BENEFITS

- People who rely on a keyboard or keyboard interface to use the Web including people who are blind and people with physical disabilities.

EXAMPLES

- **A calendar widget**
  A calendar widget allows users to add, remove or update items in their calendar using the keyboard. The controls in the widget are part of the tab order within the Web page, allowing users to tab through the controls in the widget as well as to any links or controls that follow.
- **A puzzle applet**
  Once a user tabs into an applet, further tabs and other keystrokes are handled by the applet. Instructions describing the keystroke used to exit the applet are provided prior to the applet as well as within the applet itself.
- **A modal dialog box**
  A Web application brings up a dialog box. At the bottom of the dialog are two buttons, Cancel and OK. When the dialog has been opened, focus is trapped within the dialog; tabbing from the last control in the dialog takes focus to the first control in the dialog. The dialog is dismissed by activating the Cancel button or the OK button.

TECHNIQUES

Sufficient Techniques

- G21: Ensuring that users are not trapped in content
- FLASH17: Providing keyboard access to a Flash object and avoiding a keyboard trap

Failures

- Failure: F10: Failure of Success Criterion 2.1.2 and Conformance Requirement 5 due to combining multiple content formats in a way that traps users inside one format type

2.2 – ENOUGH TIME

Provide users enough time to read and use content.

2.2.1 – TIMING ADJUSTABLE

For each time limit that is set by the content, at least one of the following is true:

Turn off: The user is allowed to turn off the time limit before encountering it; or

Adjust: The user is allowed to adjust the time limit before encountering it over a wide range that is at least ten times the length of the default setting; or

Extend: The user is warned before time expires and given at least 20 seconds to extend the time limit with a simple action (for example, "press the space bar"), and the user is allowed to extend the time limit at least ten times; or

Real-time Exception: The time limit is a required part of a real-time event (for example, an auction), and no alternative to the time limit is possible; or

Essential Exception: The time limit is essential and extending it would invalidate the activity; or

20 Hour Exception: The time limit is longer than 20 hours.

Note: This success criterion helps ensure that users can complete tasks without unexpected changes in content or context that are a result of a time limit. This success criterion should be considered in conjunction with Success Criterion 3.2.1, which puts limits on changes of content or context as a result of user action.

UNDERSTANDING

The intent of this Success Criterion is to ensure that users with disabilities are given adequate time to interact with Web content whenever possible. People with disabilities such as blindness, low vision, dexterity impairments, and cognitive limitations may require more time to read content or to perform functions such as filling out on-line forms. If Web functions are time-dependent, it will be difficult for some users to perform the required action before a time limit occurs. This may render the service inaccessible to them. Designing functions that are not time-dependent will help people with disabilities succeed at completing these functions. Providing options to disable time limits, customize the length of time limits, or request more time before a time limit occurs helps those users who require more time than expected to successfully complete tasks. These options are listed in the order that will be most helpful for the user. Disabling time limits is better than customizing the length of time limits, which is better than requesting more time before a time limit occurs.

Any process that happens without user initiation after a set time or on a periodic basis is a time limit. This includes partial or full updates of content (for example, page refresh), changes to content, or the expiration of a window of opportunity for a user to react to a request for input.

It also includes content that is advancing or updating at a rate beyond the user's ability to read and/or understand it. In other words, animated, moving or scrolling content introduces a time limit on a users ability to read content.

In some cases, however, it is not possible to change the time limit (for example, for an auction or other real-time event) and exceptions are therefore provided for those cases.

Notes regarding server time limits

Timed server redirects can be found below under Common Failures.

Non-timed server redirects (e.g., 3xx response codes) are not applicable because there is no time limit: they work instantly.

This Success Criterion applies only to time limits that are set by the content itself. For example, if a time limit is included in order to address security concerns, it would be considered to have been set by the content because it is designed to be part of the presentation and interaction experience for that content. Time limits set externally to content, such as by the user agent or by factors intrinsic to the Internet are not under the author's control and not subject to WCAG conformance requirements. Time limits set by Web servers should be under the author's/organization's control and are covered. (Success Criteria 2.2.3, 2.2.4 and 2.2.5 may also apply.)

Ten times the default was chosen based on clinical experience and other guidelines. For example, if 15 seconds is allowed for a user to respond and hit a switch, 150 seconds would be sufficient to allow almost all users to hit a switch even if they had trouble.

20 seconds was also based on clinical experience and other guidelines. 20 seconds to hit 'any switch' is sufficient for almost all users including those with spasticity. Some would fail, but some would fail all lengths of time. A reasonable period for requesting more time is required since an arbitrarily long time can provide security risks to all users, including those with disabilities, for some applications. For example, with kiosks or terminals that are used for financial transactions, it is quite common for people to walk away without signing off. This leaves them vulnerable to those walking up behind them. Providing a long period of inactivity before asking, and then providing a long period for the person to indicate that they are present can leave terminals open for abuse. If there is no activity the system should ask if the user is there. It should then ask for an indication that a person is there ('hit any key') and then wait long enough for almost anyone to respond. For "hit any key," 20 seconds would meet this. If the person indicates that they are still present, the device should return the user to the exact condition that existed before it asked the question.

20 hours was chosen as an upper limit because it is longer than a full waking day.

In cases where timing is not an intrinsic requirement but giving users control over timed events would invalidate the outcome, a third party can control the time limits for the user (for example, granting double time on a test).

See also 2.2.1: No Timing.

BENEFITS

- People with physical disabilities often need more time to react, to type and to complete activities. People with low vision need more time to locate things on screen and to read. People who are blind and using screen readers may need more time to understand screen layouts, to find information and to operate controls. People who have cognitive or language limitations need more time to read and to understand. People who are deaf and communicate in sign language may need more time to read information printed in text (which may be a second language for some).

- In circumstances where a sign-language interpreter may be relating audio content to a user who is deaf, control over time limits is also important.
- People with reading disabilities, cognitive limitations, and learning disabilities who may need more time to read or comprehend information can have additional time to read the information by pausing the content.

EXAMPLES

- A Web site uses a client side time limit to help protect users who may step away from their computer. After a period of inactivity the Web page asks if the user needs more time. If it doesn't get a response – it times out.
- A Web page has a field that automatically updates with the latest headlines in a rotating fashion. There is an interactive control that allows the user to extend the length of time between each update to as much as ten times the default. The control can be operated with either a mouse or a keyboard.
- A Web page includes an animation which includes text that appears and disappears throughout. In some cases, the text is scrolling across the screen and in others, it is only displayed for a short time before it fades into the background. The page includes a pause button so that users who have trouble reading the text before it disappears can read it.
- In an auction, there is a time limit on the amount of time a user has to submit a bid. Since the time limit applies to all users who want to bid on a particular item, it would be unfair to extend the time limit for any one particular user. Therefore, a time limit is required for this type of activity and no extension, adjustment, or deactivation of the time limit is required by this Success Criteria.
- An on-line ticket-purchasing site gives the user two minutes to confirm a purchase before the seats are returned to the general pool. Because tickets on such sites can sell out quickly, holding a ticket longer than that may invalidate the nature of the site, so this is a case in which the timing is essential and cannot be extended without invalidating the activity. However, the site does move as much of the process out of the time-critical period as possible, for instance allowing users to provide necessary information like name, payment method, etc., before entering the time-critical stage.
- A ticket-purchasing site allows the user two minutes to confirm purchase of selected seats, but warns the user when their time is almost out and allows the user to extend this time limit some number of times with a simple action such as clicking a "Extend time limit" button.

TECHNIQUES

Sufficient Techniques

Situation A – If there are session time limits:

- G133: Providing a checkbox on the first page of a multipart form that allows users to ask for longer session time limit or no session time limit
- G198: Providing a way for the user to turn the time limit off

Situation B – If a time limit is controlled by a script on the page:

- G198: Providing a way for the user to turn the time limit off
- G180: Providing the user with a means to set the time limit to 10 times the default time limit
- SCR16: Providing a script that warns the user a time limit is about to expire **AND** SCR1: Allowing the user to extend the default time limit
- FLASH19: Providing a script that warns the user a time limit is about to expire and provides a way to extend it
- FLASH24: Allowing the user to extend the default time limit
- SL21: Replacing A Silverlight Timed Animation With a Nonanimated Element

Situation C – If there are time limits on reading:

- G4: Allowing the content to be paused and restarted from where it was paused
- G198: Providing a way for the user to turn the time limit off
- SCR33: Using script to scroll content, and providing a mechanism to pause it

SCR36: Providing a mechanism to allow users to display moving, scrolling, or auto-updating text in a static window or area

Advisory Techniques

- It is advisable to use a script to poll the server and notify a user if a time limit is present
- It is advisable to use sounds to focus a user's attention

Failures

- F40: Failure of Success Criterion 2.2.1 and 2.2.4 due to using meta redirect with a time limit
- F41: Failure of Success Criterion 2.2.1, 2.2.4, and 3.2.5 due to using meta refresh to reload the page
- F58: Failure of Success Criterion 2.2.1 due to using server-side techniques to automatically redirect pages after a time-out

2.2.2 – PAUSE, STOP, HIDE

For moving, blinking, scrolling, or auto-updating information, all of the following are true:

Moving, blinking, scrolling: For any moving, blinking or scrolling information that (1) starts automatically, (2) lasts more than five seconds, and (3) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it unless the movement, blinking, or scrolling is part of an activity where it is essential; and

Auto-updating: For any auto-updating information that (1) starts automatically and (2) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it or to control the frequency of the update unless the auto-updating is part of an activity where it is essential.

Note 1: For requirements related to flickering or flashing content, refer to Guideline 2.3.

Note 2: Since any content that does not meet this success criterion can interfere with a user's ability to use the whole page, all content on the Web page (whether it is used to meet other success criteria or not) must meet this success criterion. See Conformance Requirement 5: Non-Interference.

Note 3: Content that is updated periodically by software or that is streamed to the user agent is not required to preserve or present information that is generated or received between the initiation of the pause and resuming presentation, as this may not be technically possible, and in many situations could be misleading to do so.

Note 4: An animation that occurs as part of a preload phase or similar situation can be considered essential if interaction cannot occur during that phase for all users and if not indicating progress could confuse users or cause them to think that content was frozen or broken.

UNDERSTANDING

The intent of this Success Criterion is to avoid distracting users during their interaction with a Web page.

"Moving, blinking and scrolling" refers to content in which the visible content conveys a sense of motion. Common examples include motion pictures, synchronized media presentations, animations, real-time games, and scrolling stock tickers. "Auto-updating" refers to content that updates or disappears based on a preset time interval. Common time-based content includes audio, automatically updated weather information, news, stock price updates, and auto-advancing presentations and messages. The requirements for moving, blinking and scrolling content and for auto-updating content are the same except that:

- authors have the option of providing the user with a means to control the frequency of updates when content is auto-updating and
- there is no five second exception for auto-updating since it makes little sense to auto-update for just three seconds and then stop

Content that moves or auto-updates can be a barrier to anyone who has trouble reading stationary text quickly as well as anyone who has trouble tracking moving objects. It can also cause problems for screen readers.

Moving content can also be a severe distraction for some people. Certain groups, particularly those with attention deficit disorders, find blinking content distracting, making it difficult for them to concentrate on other parts of the Web page. Five seconds was chosen because it is long enough to get a user's attention, but not so long that a user cannot wait out the distraction if necessary to use the page.

Content that is paused can either resume in real-time or continue playing from the point in the presentation where the user left off.

1. Pausing and resuming where the user left off is best for users who want to pause to read content and works best when the content is not associated with a real-time event or status.
   See 2.2.1: Timing Adjustable for additional requirements related to time-limits for reading.
2. Pausing and jumping to current display (when pause is released) is better for information that is real-time or "status" in nature. For example, weather radar, a stock ticker, a traffic camera, or an auction timer, would present misleading information if a pause caused it to display old information when the content was restarted.
   Hiding content would have the same result as pausing and jumping to current display (when pause is released).

For a mechanism to be considered "a mechanism for the user to pause," it must provide the user with a means to pause that does not tie up the user or the focus so that the page cannot be used. The word "pause" here is meant in the sense of a "pause button" although other mechanisms than a button can be used. Having an animation stop only so long as a user has focus on it (where it restarts as soon as the user moves the focus away) would not be considered a "mechanism for the user to pause" because it makes the page unusable in the process and would not meet this SC.

It is important to note that the terms "blinking" and "flashing" can sometimes refer to the same content.

- "Blinking" refers to content that causes a distraction problem. Blinking can be allowed for a short time as long as it stops (or can be stopped)

- "Flashing" refers to content that can trigger a seizure (if it is more than 3 per second and large and bright enough). This cannot be allowed even for a second or it could cause a seizure. And turning the flash off is also not an option since the seizure could occur faster than most users could turn it off.
- Blinking usually does not occur at speeds of 3 per second or more, but it can. If blinking occurs faster than 3 per second, it would also be considered a flash.

BENEFITS

- Providing content that stops blinking after five seconds or providing a mechanism for users to stop blinking content allows people with certain disabilities to interact with the Web page.
- One use of content that blinks is to draw the visitor's attention to that content. Although this is an effective technique for all users with vision, it can be a problem for some users if it persists. For certain groups, including people with low literacy, reading and intellectual disabilities, and people with attention deficit disorders, content that blinks may make it difficult or even impossible to interact with the rest of the Web page.

EXAMPLES

- **An essential animation can be paused without affecting the activity**
  A Web site helps users understand 'how things work' through animations that demonstrate processes. Animations have "pause" and "restart" buttons.
- **A stock ticker**
  A stock ticker has "pause" and "restart" buttons. Pausing the ticker causes it to pause on the currently displayed stock. Restarting causes the ticker to resume from the stopped point but with a notice that the display is delayed. Since the intent of the stock ticker is usually to provide realtime information, there might also be a button that would advance the ticker to the most recently traded stock.
- **A game is designed so that users take turns rather than competing in real-time**
  One party can pause the game without invalidating the competitive aspect of it.
- **A Web advertisement**
  An advertisement blinks to get viewers attention but stops after 5 seconds
- **A form prompt**
  A form blinks an arrow near the submit button if a user finishes filling out the form but does not activate the submit button. The blinking stops after 5 seconds.
- **An animation**
  An animation runs in the upper portion of the page but has a "freeze animation" button near the bottom of the animation.
- **A "loading" animation**
  A preloader animation is shown on a page which requires a certain percentage of a large file to be downloaded before playback can begin. The animation is the only content on the page and instructs the user to please wait while the video loads. Because the moving content is not presented in parallel with other content, no mechanism to pause, stop or hide it needs to be provided, even though the animation may run for more than 5 seconds for users with slower connections.
- **A full-page advertisement**
  A site requires that all users view a 15 second advertisement before they can access free content available from their site. Because viewing the advertisement is a requirement for all users and because it is not presented in parallel with other content, no mechanism to pause, stop or hide it needs to be provided.

TECHNIQUES

Sufficient Techniques

- G4: Allowing the content to be paused and restarted from where it was paused
- SL11: Pausing or Stopping A Decorative Silverlight Animation
- SL12: Pausing, Stopping, or Playing Media in Silverlight MediaElements
- SCR33: Using script to scroll content, and providing a mechanism to pause it
- FLASH35: Using script to scroll Flash content, and providing a mechanism to pause it
- G11: Creating content that blinks for less than 5 seconds
- G187: Using a technology to include blinking content that can be turned off via the user agent
- G152: Setting animated gif images to stop blinking after n cycles (within 5 seconds)
- SCR22: Using scripts to control blinking and stop it in five seconds or less
- FLASH36: Using scripts to control blinking and stop it in five seconds or less
- G186: Using a control in the Web page that stops moving, blinking, or auto-updating content
- G191: Providing a link, button, or other mechanism that reloads the page without any blinking content
- SL24: Using AutoPlay to Keep Silverlight Media from Playing Automatically

Advisory Techniques

- It is advisable to provide a mechanism to stop all content that blinks within a web page
- It is advisable to provide the user with a means to stop moving content, even if it stops automatically within 5 seconds

Failures

- Failure F4: Failure of Success Criterion 2.2.2 due to using text-decoration:blink without a mechanism to stop it in less than five seconds
- Failure F7: Failure of Success Criterion 2.2.2 due to an object or applet, such as Java or Flash, that has blinking content without a mechanism to pause the content that blinks for more than five seconds
- Failure F16: Failure of Success Criterion 2.2.2 due to including scrolling content where movement is not essential to the activity without also including a mechanism to pause and restart the content
- Failure F47: Failure of Success Criterion 2.2.2 due to using the blink element
- Failure F50: Failure of Success Criterion 2.2.2 due to a script that causes a blink effect without a mechanism to stop the blinking at 5 seconds or less

2.3 – SEIZURES AND PHYSICAL REACTIONS

Do not design content in a way that is known to cause seizures or physical reactions.

2.3.1 – THREE FLASHES OR BELOW THRESHOLD

Web pages do not contain anything that flashes more than three times in any one second period, or the flash is below the general flash and red flash thresholds.

Note: Since any content that does not meet this success criterion can interfere with a user's ability to use the whole page, all content on the Web page (whether it is used to meet other success criteria or not) must meet this success criterion. See Conformance Requirement 5: Non-Interference.

UNDERSTANDING

The intent of this Success Criterion is to allow users to access the full content of a site without inducing seizures due to photosensitivity.

Individuals who have photosensitive seizure disorders can have a seizure triggered by content that flashes at certain frequencies for more than a few flashes. People are even more sensitive to red flashing than to other colors, so a special test is provided for saturated red flashing. These guidelines are based on guidelines for the broadcasting industry as adapted for computer screens, where content is viewed from a closer distance (using a larger angle of vision).

Flashing can be caused by the display, the computer rendering the image or by the content being rendered. The author has no control of the first two. They can be addressed by the design and speed of the display and computer. The intent of this criterion is to ensure that flicker that violates the flash thresholds is not caused by the content itself. For example, the content could contain a video clip or animated image of a series of strobe flashes, or close-ups of rapid-fire explosions.

This Success Criterion replaces a much more restrictive criterion in WCAG 1.0 that did not allow any flashing (even of a single pixel) within a broad frequency range (3 to 50 Hz). This Success Criterion is based on existing specifications in use in the UK and by others for television broadcast and has been adapted for computer display viewing. The 1024 x 768 screen is used as the reference screen resolution for the evaluation. The 341 x 256 pixel block represents a 10 degree viewport at a typical viewing distance. (The 10 degree field is taken from the original specifications and represents the central vision portion of the eye, where people are most susceptible to photo stimuli.)

The combined area of flashes occurring concurrently and contiguously means the total area that is actually flashing at the same time. It is calculated by adding up the contiguous area that is flashing simultaneously within any 10 degree angle of view.

The terms "blinking" and "flashing" can sometimes refer to the same content.

- "Blinking" refers to content that causes a distraction problem. Blinking can be allowed for a short time as long as it stops (or can be stopped)
- "Flashing" refers to content that can trigger a seizure (if it is more than 3 per second and large and bright enough). This cannot be allowed even for a second or it could cause a seizure. And turning the flash off is also not an option since the seizure could occur faster than most users could turn it off.

- Blinking usually does not occur at speeds of 3 per second or more, but it can. If blinking occurs faster than 3 per second, it would also be considered a flash.

BENEFITS

- Individuals who have seizures when viewing flashing material will be able to view all of the material on a site without having a seizure and without having to miss the full experience of the content by being limited to text alternatives. This includes people with photosensitive epilepsy as well as other photosensitive seizure disorders.

EXAMPLES

- A Web site has video of muzzle flash of machine gun fire, but limits the size of the flashing image to a small portion of the screen below the flash threshold size.
- A movie with a scene involving very bright lightning flashes is edited so that the lightning only flashes three times in any one second period.

RELATED RESOURCES

- Harding FPA Web Site
- Trace Center Photosensitive Epilepsy Analysis Tool (PEAT)
- Epilepsy Action
- Epilepsy Foundation

TECHNIQUES

Sufficient Techniques

- G19: Ensure no component of the content flashes more than three times in any 1-second period
- G176: Keeping the flashing area small enough
- G15: Use a tool to ensure content does not violate the general flash threshold or red flash threshold

Advisory Techniques

- Reduce contrast for any flashing content
- Avoid fully saturated reds for any flashing content
- Reduce the number of flashes even if they do not violate thresholds
- Provide a mechanism to suppress any flashing content before it begins
- Slow down live material to avoid rapid flashes (as in flashbulbs)
- Freeze the image momentarily if 3 flashes within one second are detected
- Drop the contrast ratio if 3 flashes within one second are detected
- Allow users to set a custom flash rate limit

2.4 – NAVIGABLE

Provide ways to help users navigate, find content, and determine where they are. It is advisable to limit the number of links per page, provide mechanisms to navigate to different sections of the content of a web page, make links visually distinctive, and highlight search terms.

2.4.1 – BYPASS BLOCKS

A mechanism is available to bypass blocks of content that are repeated on multiple Web pages.

UNDERSTANDING

The intent of this Success Criterion is to allow people who navigate sequentially through content more direct access to the primary content of the Web page. Web pages and applications often have content that appears on other pages or screens. Examples of repeated blocks of content include but are not limited to navigation links, heading graphics, and advertising frames. Small repeated sections such as individual words, phrases or single links are not considered blocks for the purposes of this provision.

This is in contrast to a sighted user's ability to ignore the repeated material either by focusing on the center of the screen (where main content usually appears) or a mouse user's ability to select a link with a single mouse click rather than encountering every link or form control that comes before the item they want.

It is not the intent of this Success Criterion to require authors to provide methods that are redundant to functionality provided by the user agent. Most web browsers provide keyboard shortcuts to move the user focus to the top of the page, so if a set of navigation links is provided at the bottom of a web page providing a "skip" link may be unnecessary.

Although this Success Criterion deals with blocks of content that are repeated on multiple pages, we also strongly promote structural markup on individual pages as per Success Criteria 1.3.1.

Although the success criterion does not specifically use the term “within a set of web pages”, the concept of the pages belonging to a set is implied. An author would not be expected to avoid any possible duplication of content in any two pages that are not in some way related to each other; that are not "Web pages that share a common purpose and that are created by the same author, group or organization” (the definition of set of web pages).

Even for web pages that are not in a set, if a web page has blocks of text that are repeated within the page it may be helpful (but not required) to provide a means to skip over them.

BENEFITS

- When this Success Criterion is not satisfied, it may be difficult for people with some disabilities to reach the main content of a Web page quickly and easily.
- Screen reader users who visit several pages on the same site can avoid having to hear all heading graphics and dozens of navigation links on every page before the main content is spoken.
- People who use only the keyboard or a keyboard interface can reach content with fewer keystrokes. Otherwise, they might have to make dozens of keystrokes before reaching a link in the main content area. This can take a long time and may cause severe physical pain for some users.
- People who use screen magnifiers do not have to search through the same headings or other blocks of information to find where the content begins each time they enter a new page.

- People with cognitive limitations as well as people who use screen readers may benefit when links are grouped into lists

EXAMPLES

- A news organization's home page contains a main story in the middle of the page, surrounded by many blocks and sidebars for advertising, searching, and other services. There is a link at the top of the page that jumps to the main story. Without using this link, a keyboard user needs to tab through approximately 40 links to reach the main story; the screen reader user has to listen to 200 words; and the screen magnifier user must search around for the location of the main body.

RELATED RESOURCES

- WebAIM: Semantic Structure
- Heading Tags

TECHNIQUES

Sufficient Techniques

- Create links to skip blocks of repeated material *using one of the following techniques:*
  - G1: Adding a link at the top of each page that goes directly to the main content area
  - G123: Adding a link at the beginning of a block of repeated content to go to the end of the block
  - G124: Adding links at the top of the page to each area of the content
  - SL25: Using Controls and Programmatic Focus to Bypass Blocks of Content in Silverlight
- Group blocks of repeated material in a way that can be skipped *using one of the following techniques:*
  - ARIA11: Using ARIA landmarks to identify regions of a page
  - H69: Providing heading elements at the beginning of each section of content
  - PDF9: Providing headings by marking content with heading tags in PDF documents
  - H70: Using frame elements to group blocks of repeated material **AND** H64: Using the title attribute of the frame and iframe elements
  - SCR28: Using an expandable and collapsible menu to bypass block of content
  - SL29: Using Silverlight "List" Controls to Define Blocks that can be Bypassed

Advisory Techniques

- Provide keyboard access to important links and form controls
- Provide skip links to enhance page navigation
- Provide access keys
- Use accessibility supported technologies which allow structured navigation by user agents and assistive technologies
- C6: Position content based on structural markup

2.4.2 – PAGE TITLED

Web pages have titles that describe topic or purpose.

UNDERSTANDING

The intent of this Success Criterion is to help users find content and orient themselves within it by ensuring that each Web page has a descriptive title. Titles identify the current location without requiring users to read or interpret page content. When titles appear in site maps or lists of search results, users can more quickly identify the content they need. User agents make the title of the page easily available to the user for identifying the page. For instance, a user agent may display the page title in the window title bar or as the name of the tab containing the page.

In cases where the page is a document or a web application, the name of the document or web application would be sufficient to describe the purpose of the page. Note that it is not required to use the name of the document or web application; other things may also describe the purpose or the topic of the page.

Success Criteria 2.4.4 and 2.4.9 deal with the purpose of links, many of which are links to web pages. Here also, the name of a document or web application being linked to would be sufficient to describe the purpose of the link. Having the link and the title agree, or be very similar, is good practice and provides continuity between the link 'clicked on' and the web page that the user lands on.

BENEFITS

- This criterion benefits all users in allowing users to quickly and easily identify whether the information contained in the Web page is relevant to their needs.
- People with visual disabilities will benefit from being able to differentiate content when multiple Web pages are open.
- People with cognitive disabilities, limited short-term memory and reading disabilities also benefit from the ability to identify content by its title.
- This criterion also benefits people with severe mobility impairments whose mode of operation relies on audio when navigating between Web pages.

EXAMPLES

- **An HTML Web page**
  The descriptive title of an HTML Web page is marked up with the <title> element so that it will be displayed in the title bar of the user agent.
- **A document collection.**
  The title of Understanding WCAG 2.0 is "Understanding WCAG 2.0."
  - The introduction page has the title "Introduction to Understanding WCAG 2.0."
  - Major sections of the document are pages titled "Understanding Guideline X" and "Understanding Success Criterion X."
  - Appendix A has the title "Glossary."
  - Appendix B has the title "Acknowledgements."
  - Appendix C has the title "References."
- **A Web application.**

A banking application lets a user inspect his bank accounts, view past statements, and perform transactions. The Web application dynamically generates titles for each Web page, e.g., "Bank XYZ, accounts for John Smith" "Bank XYZ, December 2005 statement for Account 1234-5678".

TECHNIQUES

Sufficient Techniques

- G88: Providing descriptive titles for Web pages *using one of the following techniques:*
    - H25: Providing a title using the title element
    - PDF18: Specifying the document title using the Title entry in the document information dictionary of a PDF document

Advisory Techniques

- G127: Identify a Web page's relationship to a larger collection of Web pages
- Identify the subject of the Web page
- Provide a meaningful name for identifying frames
- Use unique titles for Web pages
- Provide a descriptive top-level page heading

Failures

- F25: Failure of SC 2.4.2 due to the title of a Web page not identifying the contents

2.4.3 – FOCUS ORDER

If a Web page can be navigated sequentially and the navigation sequences affect meaning or operation, focusable components receive focus in an order that preserves meaning and operability.

UNDERSTANDING

The intent of this Success Criterion is to ensure that when users navigate sequentially through content, they encounter information in an order that is consistent with the meaning of the content and can be operated from the keyboard. This reduces confusion by letting users form a consistent mental model of the content. There may be different orders that reflect logical relationships in the content. For example, moving through components in a table one row at a time or one column at a time both reflect the logical relationships in the content. Either order may satisfy this Success Criterion.

The way that sequential navigation order is determined in Web content is defined by the technology of the content. For example, simple HTML defines sequential navigation via the notion of tabbing order. Dynamic HTML may modify the navigation sequence using scripting along with the addition of a tabindex attribute to allow focus to additional elements. If no scripting or tabindex attributes are used, the navigation order is the order that components appear in the content stream. (See HTML 4.01 Specification, section 17.11, "Giving focus to an element").

An example of keyboard navigation that is not the sequential navigation addressed by this Success Criterion is using arrow key navigation to traverse a tree component. The user can use the up and down arrow keys to move from tree node to tree node. Pressing the right arrow key may expand a node, then using the down arrow key, will move into the newly expanded nodes. This navigation sequence follows the expected sequence for a tree control - as additional items get expanded or collapsed, they are added or removed from the navigation sequence.

The focus order may not be identical to the programmatically determined reading order (see Success Criterion 1.3.2) as long as the user can still understand and operate the Web page. Since there may be several possible logical reading orders for the content, the focus order may match any of them. However, when the order of a particular presentation differs from the programmatically determined reading order, users of one of these presentations may find it difficult to understand or operate the Web page. Authors should carefully consider all these users as they design their Web pages.

For example, a screen reader user interacts with the programmatically determined reading order, while a sighted keyboard user interacts with the visual presentation of the Web page. Care should be taken so that the focus order makes sense to both of these sets of users and does not appear to either of them to jump around randomly.

For clarity:

1. Focusable components need to receive focus in an order that preserves meaning and operability only when navigation sequences affect meaning and operability.
2. In those cases where it is required, there may be more than one order that will preserve meaning and operability.
3. If there is more than one order that preserves meaning and operability, only one of them needs to be provided.

BENEFITS

These techniques benefit keyboard users who navigate documents sequentially and expect the focus order to be consistent with the sequential reading order.

- People with mobility impairments who must rely on keyboard access for operating a page benefit from a logical, usable focus order.
- People with disabilities that make reading difficult can become disoriented when tabbing takes focus someplace unexpected. They benefit from a logical focus order.
- People with visual impairments can become disoriented when tabbing takes focus someplace unexpected or when they cannot easily find the content surrounding an interactive element.
- Only a small portion of the page may be visible to an individual using a screen magnifier at a high level of magnification. Such a user may interpret a field in the wrong context if the focus order is not logical.

EXAMPLES

1. On a web page that contains a tree of interactive controls, the user can use the up and down arrow keys to move from tree node to tree node. Pressing the right arrow key expands a node, then using the down arrow key moves into the newly expanded nodes.
2. A Web page implements modeless dialogs via scripting. When the trigger button is activated, a dialog opens. The interactive elements in the dialog are inserted in the focus order immediately after the button. When the dialog is open, the focus order goes from the button to the elements of the dialog, then to the

interactive element following the button. When the dialog is closed, the focus order goes from the button to the following element.

3. A Web page implements modal dialogs via scripting. When the trigger button is activated, a dialog opens and focus is set to the first interactive element in the dialog. As long as the dialog is open, focus is limited to the elements of the dialog. When the dialog is dismissed, focus returns to the button or the element following the button.
4. An HTML Web page is created with the left hand navigation occurring in the HTML after the main body content, and styled with CSS to appear on the left hand side of the page. This is done to allow focus to move to the main body content first without requiring tabIndex attributes or JavaScript.
5. While this example passes the Success Criterion, it is not necessarily true that all CSS positioning would. More complex positioning examples may or may not preserve meaning and operability
6. The following example fails to meet the Success Criterion:
7. A company's Web site includes a form that collects marketing data and allows users to subscribe to several newsletters published by the company. The section of the form for collecting marketing data includes fields such as name, street address, city, state or province, and postal code. Another section of the form includes several checkboxes so that users can indicate newsletters they want to receive. However, the tab order for the form skips between fields in different sections of the form, so that focus moves from the name field to a checkbox, then to the street address, then to another checkbox.

TECHNIQUES

Sufficient Techniques

- G59: Placing the interactive elements in an order that follows sequences and relationships within the content
- Giving focus to elements in an order that follows sequences and relationships within the content *using one of the following techniques:*
  - H4: Creating a logical tab order through links, form controls, and objects
  - FLASH15: Using the tabIndex property to specify a logical reading order and a logical tab order in Flash
  - C27: Making the DOM order match the visual order
  - PDF3: Ensuring correct tab and reading order in PDF documents
  - SL34: Using the Silverlight Default Tab Sequence and Altering Tab Sequences With Properties
- Changing a Web page dynamically *using one of the following techniques:*
  - SCR26: Inserting dynamic content into the Document Object Model immediately following its trigger element
  - SCR37: Creating Custom Dialogs in a Device Independent Way
- SCR27: Reordering page sections using the Document Object Model

Advisory Techniques

- It is advisable to provide a highly visible highlighting mechanism for links or controls when they receive keyboard focus
- It is advisable to create alternative presentation orders

Failures

- F44: Failure of Success Criterion 2.4.3 due to using tabindex to create a tab order that does not preserve meaning and operability
- F85: Failure of Success Criterion 2.4.3 due to using dialogs or menus that are not adjacent to their trigger control in the sequential navigation order

2.4.4 – LINK PURPOSE

The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general.

UNDERSTANDING

The intent of this Success Criterion is to help users understand the purpose of each link so they can decide whether they want to follow the link. Whenever possible, provide link text that identifies the purpose of the link without needing additional context. Assistive technology has the ability to provide users with a list of links that are on the Web page. Link text that is as meaningful as possible will aid users who want to choose from this list of links. Meaningful link text also helps those who wish to tab from link to link. Meaningful links help users choose which links to follow without requiring complicated strategies to understand the page.

The text of, or associated with, the link is intended to describe the purpose of the link. In cases where the link takes one to a document or a web application, the name of the document or web application would be sufficient to describe the purpose of the link (which is to take you to the document or web application). Note that it is not required to use the name of the document or web application; other things may also describe the purpose of the link.

Success Criterion 2.4.2 deals with the titles of pages. Here also, the name of a document or web application being presented on the page would be sufficient to describe the purpose of the page. Having the link and the title agree, or be very similar, is good practice and provides continuity between the link 'clicked on' and the web page that the user lands on.

In some situations, authors may want to provide part of the description of the link in logically related text that provides the context for the link. In this case the user should be able to identify the purpose of the link without moving focus from the link. In other words, they can arrive on a link and find out more about it without losing their place. This can be achieved by putting the description of the link in the same sentence, paragraph, list item, or table cell as the link, or in the table header cell for a link in a data table, because these are directly associated with the link itself. Alternatively, authors may choose to use an ARIA technique to associate additional text on the page with the link.

This context will be most usable if it precedes the link. (For instance, if you must use ambiguous link text, it is better to put it at the end of the sentence that describes its destination, rather than putting the ambiguous phrase at the beginning of the sentence.) If the description follows the link, there can be confusion and difficulty for screen reader users who are reading through the page in order (top to bottom).

It is a best practice for links with the same destination to have consistent descriptions (and this is a requirement per Success Criterion 3.2.4 for pages in a set). It is also a best practice for links with different purposes and destinations to have different descriptions.

A best practice for links to conforming alternate versions is to ensure that the link text to the conforming alternate version indicates in link text that the page it leads to represents the more accessible version. This information may also be provided in text - the goal is to ensure that the end user knows what the purpose of the link is.

The Success Criterion includes an exception for links for which the purpose of the link cannot be determined from the information on the Web page. In this situation, the person with the disability is not at a disadvantage; there is no additional context available to understand the link purpose. However, whatever amount of context is available on the Web page that can be used to interpret the purpose of the link must be made available in the link text or programmatically associated with the link to satisfy the Success Criterion.

There may be situations where the purpose of the link is is supposed to be unknown or obscured. For instance, a game may have links identified only as door #1, door #2, and door #3. This link text would be sufficient because the purpose of the links is to create suspense for all users.

See also 2.4.2: Link Purpose (Link Only).

BENEFITS

- This Success Criterion helps people with motion impairment by letting them skip links that they are not interested in, avoiding the keystrokes needed to visit the referenced content and then returning to the current content.
- People with cognitive limitations will not become disoriented by multiple means of navigation to and from content they are not interested in.
- People with visual disabilities will be able to determine the purpose of a link by exploring the link's context.

EXAMPLES

- **A link contains text that gives a description of the information at that URI**
  A page contains the sentence "There was much bloodshed during the Medieval period of history." Where "Medieval period of history" is a link.
- **A link is preceded by a text description of the information at that URI**
  A page contains the sentence "Learn more about the Government of Ireland's Commission on Electronic Voting at Go Vote!" where "Go Vote!" is a link.
- **Both an icon and text are included in the same link**
  An icon of a voting machine and the text "Government of Ireland's Commission of Electronic Voting" are combined to make a single link. The alt text for the icon is null, since the purpose of the link is already described by the text of the link next to the icon.
- **A list of book titles**
  A list of books is available in three formats: HTML, PDF, and mp3 (a recording of a person reading the book). To avoid hearing the title of each book three times (once for each format), the first link for each book is the title of the book, the second link says "PDF" and the third says, "mp3."
- **News article summaries**
  A Web page contains a collection of news articles. The main page lists the first few sentences of each article, followed by a "Read more" link. A screen reader command to read the current paragraph provides the context to interpret the purpose of the link.

RELATED RESOURCES

- Using Link Titles to Help Users Predict Where They Are Going
- WebAIM Techniques for Hypertext Links

TECHNIQUES

Sufficient Techniques

- G91: Providing link text that describes the purpose of a link
- H30: Providing link text that describes the purpose of a link for anchor elements

- H24: Providing text alternatives for the area elements of image maps
- FLASH27: Providing button labels that describe the purpose of a button
- Allowing the user to choose short or long link text *using one of the following techniques:*
    - G189: Providing a control near the beginning of the Web page that changes the link text
    - SCR30: Using scripts to change the link text
    - FLASH7: Using scripting to change control labels
- G53: Identifying the purpose of a link using link text combined with the text of the enclosing sentence
- Providing a supplemental description of the purpose of a link *using one of the following techniques:*
    - H33: Supplementing link text with the title attribute
    - C7: Using CSS to hide a portion of the link text
- Identifying the purpose of a link using link text combined with programmatically determined link context *using one of the following techniques:*
    - ARIA7: Using aria-labelledby for link purpose
    - ARIA8: Using aria-label for link purpose
    - H77: Identifying the purpose of a link using link text combined with its enclosing list item
    - H78: Identifying the purpose of a link using link text combined with its enclosing paragraph
    - H79: Identifying the purpose of a link in a data table using the link text combined with its enclosing table cell and associated table header cells
    - H81: Identifying the purpose of a link in a nested list using link text combined with the parent list item under which the list is nested
- G91: Providing link text that describes the purpose of a link **AND** Semantically indicating links *using one of the following techniques:*
    - PDF11: Providing links and link text using the Link annotation and the /Link structure element in PDF documents
    - PDF13: Providing replacement text using the /Alt entry for links in PDF documents
    - SL18: Providing Text Equivalent for Nontext Silverlight Controls With AutomationProperties.Name

Advisory Techniques

- H2: Combining adjacent image and text links for the same resource
- FLASH5: Combining adjacent image and text buttons for the same resource
- H80: Identifying the purpose of a link using link text combined with the preceding heading element

Failures

- F63: Failure of Success Criterion 2.4.4 due to providing link context only in content that is not related to the link
- F89: Failure of Success Criteria 2.4.4, 2.4.9 and 4.1.2 due to not providing an accessible name for an image which is the only content in a link

2.4.5 – MULTIPLE WAYS

More than one way is available to locate a Web page within a set of Web pages except where the Web Page is the result of, or a step in, a process.

UNDERSTANDING

The intent of this Success Criterion is to make it possible for users to locate content in a manner that best meets their needs. Users may find one technique easier or more comprehensible to use than another.

Even small sites should provide users some means of orientation. For a three or four page site, with all pages linked from the home page, it may be sufficient simply to provide links from and to the home page where the links on the home page can also serve as a site map.

BENEFITS

- Providing an opportunity to navigate sites in more than one manner can help people find information faster. Users with visual impairments may find it easier to navigate to the correct part of the site by using a search, rather than scrolling through a large navigation bar using a screen magnifier or screen reader. A person with cognitive disabilities may prefer a table of contents or site map that provides an overview of the site rather than reading and traversing through several Web pages. Some users may prefer to explore the site in a sequential manner, moving from Web page to Web page in order to best understand the concepts and layout.
- Individuals with cognitive limitations may find it easier to use search features than to use a hierarchical navigation scheme that be difficult to understand.

EXAMPLES

- **A search mechanism.**
  A large food processing company provides a site containing recipes created using its products. The site provides a search mechanism to search for recipes using a particular ingredient. In addition, it provides a list box that lists several categories of foods. A user may type "soup" in to the search engine or may select "soup" from the list box to go to a page with a list of recipes made from the company's soup products
- **Links between Web pages.**
  A local hair salon has created a Web site to promote its services. The site contains only five Web pages. There are links on each Web page to sequentially move forward or backward through the Web pages. In addition, each Web page contains a list of links to reach each of the other Web pages.
- **Where content is a result of a process or task - Funds transfer confirmation.**
  An on-line banking site allows fund transfer between accounts via the Web. There is no other way to locate the confirmation of fund transfer until the account owner completes the transfer.
- **Where content is a result of a process or task - Search engine results.**
  A search engine provides the search results based on user input. There is no other way to locate the search results except to perform the search process itself.

TECHNIQUES

Sufficient Techniques

- G125: Providing links to navigate to related Web pages
- G64: Providing a Table of Contents
- PDF2: Creating bookmarks in PDF documents
- G63: Providing a site map
- G161: Providing a search function to help users find content
- G126: Providing a list of links to all other Web pages
- G185: Linking to all of the pages on the site from the home page

Advisory Techniques

- H59: Use the link element and navigation tools
- Include information about presentation modes in tables of contents and concept maps

2.4.6 – HEADINGS AND LABELS

Headings and labels describe topic or purpose.

UNDERSTANDING

The intent of this Success Criterion is to help users understand what information is contained in Web pages and how that information is organized. When headings are clear and descriptive, users can find the information they seek more easily, and they can understand the relationships between different parts of the content more easily. Descriptive labels help users identify specific components within the content.

Labels and headings do not need to be lengthy. A word, or even a single character, may suffice if it provides an appropriate cue to finding and navigating content.

This success criterion does not require headings or labels. This success criterion requires that if headings or labels are provided, they be descriptive. Also note that, if headings or labels are provided, they must meet 1.3.1: Info and Relationships.

BENEFITS

- Descriptive headings are especially helpful for users who have disabilities that make reading slow and for people with limited short-term memory. These people benefit when section titles make it possible to predict what each section contains.
- People who have difficulty using their hands or who experience pain when doing so will benefit from techniques that reduce the number of keystrokes required to reach the content they need.
- This Success Criterion helps people who use screen readers by ensuring that labels and headings are meaningful when read out of context, for example, in a Table of Contents, or when jumping from heading to heading within a page.
- This Success Criterion may also help users with low vision who can see only a few words at a time.

EXAMPLES

- **A news site.**
  The home page of a news site lists the headlines for the top stories of the hour. Under each heading are the first 35 words of the story and a link to the full article. Each headline gives a clear idea of the article's subject.
- **A guide on how to write well**
  A guide on writing contains the following section titles: How To Write Well, Cut Out Useless Words, Identify Unnecessary Words, etc. The section headings are clear and concise and the structure of the information is reflected in the structure of the headings.
- **Consistent headings in different articles**
  A Web site contains papers from a conference. Submissions to the conference are required to have the following organization: Summary, Introduction, [other sections unique to this article], Conclusion, Author Biography, Glossary, and Bibliography. The title of each Web page clearly identifies the article it contains, creating a useful balance between the uniqueness of the articles and the consistency of the section headings.
- **A form asking the name of the user**
  A form asks the name of the user. It consists of two input fields to ask for the first and last name. The first field is labeled "First name", the second is labeled "Last name"."

RELATED RESOURCES

- How Users Read on the Web A study showing that most users scan Web pages rather than reading them word by word.
- Applying Writing Guidelines to Web Pages A report on the effects of making Web sites concise, easy to scan, and objective.

TECHNIQUES

Sufficient Techniques

- G130: Providing descriptive headings
- G131: Providing descriptive labels
- It is advisable to use unique section headings in a web page
- It is advisable to start section headings with unique information

2.4.7 – FOCUS VISIBLE

Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible.

UNDERSTANDING

The purpose of this success criterion is to help a person know which element has the keyboard focus.

The purpose of this success criterion is to help a person know which element among multiple elements has the keyboard focus. If there is only one keyboard actionable control on the screen, the success criterion would be met because the visual design presents only one keyboard actionable item.

Note that a keyboard focus indicator can take different forms. One common way is a caret within the text field to indicate that the text field has the keyboard focus. Another is a visual change to a button to indicate that that button has the keyboard focus.

BENEFITS

- This Success Criterion helps anyone who relies on the keyboard to operate the page, by letting them visually determine the component on which keyboard operations will interact at any point in time.
- People with attention limitations, short term memory limitations, or limitations in executive processes benefit by being able to discover where the focus is located.

EXAMPLES

- When text fields receive focus, a vertical bar is displayed in the field, indicating that the user can insert text, OR all of the text is highlighted, indicating that the user can type over the text.
- When a user interface control receives focus, a visible border is displayed around it.

RELATED RESOURCES

- Styling form controls with CSS, revisited

TECHNIQUES

Sufficient Techniques

- G149: Use user interface components that are highlighted by the user agent when they receive focus
- C15: Using CSS to change the presentation of a user interface component when it receives focus
- G165: Using the default focus indicator for the platform so that high visibility default focus indicators will carry over
- G195: Using an author-supplied, highly visible focus indicator
- SCR31: Using script to change the background color or border of the element with focus
- FLASH20: Reskinning Flash components to provide highly visible focus indication
- SL2: Changing The Visual Focus Indicator in Silverlight
- SL7: Designing a Focused Visual State for Custom Silverlight Controls

Advisory Techniques

- It is advisable to highlight a link or control when the mouse hovers over it
- It is advisable to provide a highly visible highlighting mechanism for links or controls when they receive keyboard focus

Failures

- F55: Failure of Success Criteria 2.1.1, 2.4.7, and 3.2.1 due to using script to remove focus when focus is received
- F78: Failure of Success Criterion 2.4.7 due to styling element outlines and borders in a way that removes or renders non-visible the visual focus indicator

3. UNDERSTANDABLE

Information and the operation of user interface must be understandable.

3.1 READABLE

Make text content readable and understandable.

It is advisable to use the following techniques to meet all SC within this principal.

- Setting expectations about content in the page from uncontrolled sources
- Providing sign language interpretation for all content
- Using the clearest and simplest language appropriate for the content
- Avoiding centrally aligned text
- Avoiding text that is fully justified (to both left and right margins) in a way that causes poor spacing between words or characters
- Using left-justified text for languages that are written left to right and right-justified text for languages that are written right-to-left
- Limiting text column width
- Avoiding chunks of italic text
- Avoiding overuse of different styles on individual pages and in sites
- Making links visually distinct
- Using images, illustrations, video, audio, or symbols to clarify meaning
- Providing practical examples to clarify content
- Using a light pastel background rather than a white background behind black text
- Avoiding the use of unique interface controls unnecessarily
- Using upper and lower case according to the spelling rules of the text language
- Avoiding unusual foreign words
- Providing sign language versions of information, ideas, and processes that must be understood in order to use the content
- Making any reference to a location in a Web page into a link to that location
- Making references to a heading or title include the full text of the title
- Providing easy-to-read versions of basic information about a set of Web pages, including information about how to contact the Webmaster
- Providing a sign language version of basic information about a set of Web pages, including information about how to contact the Webmaster

3.1.1 – LANGUAGE OF PAGE

The default human language of each Web page can be programmatically determined.

UNDERSTANDING

The intent of this Success Criterion is to ensure that content developers provide information in the Web page that user agents need to present text and other linguistic content correctly. Both assistive technologies and conventional user agents can render text more accurately when the language of the Web page is identified. Screen readers can load the correct pronunciation rules. Visual browsers can display characters and scripts correctly. Media players can show captions correctly. As a result, users with disabilities will be better able to understand the content.

The default human language of the Web page is the default text-processing language as discussed in Internationalization Best Practices: Specifying Language in XHTML & HTML Content. When a Web page uses several languages, the default text-processing language is the language which is used most. (If several languages are used equally, the first language used should be chosen as the default human language.)

For multilingual sites targeting Conformance Level A, the Working Group strongly encourages developers to follow Success Criterion 3.1.2 as well even though that is a Level AA Success Criterion.

BENEFITS

This Success Criterion helps:

- people who use screen readers or other technologies that convert text into synthetic speech;
- people who find it difficult to read written material with fluency and accuracy, such as recognizing characters and alphabets or decoding words;
- people with certain cognitive, language and learning disabilities who use text-to-speech software
- people who rely on captions for synchronized media.

EXAMPLES

- **Example 1. A Web page with content in two languages**
  A Web page produced in Germany and written in HTML includes content in both German and English, but most of the content is in German. The default human language is identified as German (de) by the lang attribute on the html element.

RELATED RESOURCES

- Internationalization Best Practices: Specifying Language in XHTML & HTML Content

TECHNIQUES

Sufficient Techniques

- H57: Using language attributes on the html element
- FLASH13: Using HTML language attributes to specify language in Flash content
- PDF16: Setting the default language using the /Lang entry in the document catalog of a PDF document
- PDF19: Specifying the language for a passage or phrase with the Lang entry in PDF documents

Advisory Techniques

- SVR5: Specify the default language in the HTTP header
- Use http or the Content-Language meta tag for metadata

3.1.2 – LANGUAGE OF PARTS

The human language of each passage or phrase in the content can be programmatically determined except for proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the immediately surrounding text.

UNDERSTANDING

The intent of this Success Criterion is to ensure that user agents can correctly present content written in multiple languages. This makes it possible for user agents and assistive technologies to present content according to the presentation and pronunciation rules for that language. This applies to graphical browsers as well as screen readers, braille displays, and other voice browsers.

Both assistive technologies and conventional user agents can render text more accurately if the language of each passage of text is identified. Screen readers can use the pronunciation rules of the language of the text. Visual browsers can display characters and scripts in appropriate ways. This is especially important when switching between languages that read from left to right and languages that read from right to left, or when text is rendered in a language that uses a different alphabet. Users with disabilities who know all the languages used in the Web page will be better able to understand the content when each passage is rendered appropriately.

When no other language has been specified for a phrase or passage of text, its human language is the default human language of the Web page (see Success Criterion 3.1.1). So the human language of all content in single language documents can be programmatically determined.

Individual words or phrases in one language can become part of another language. For example, "rendezvous" is a French word that has been adopted in English, appears in English dictionaries, and is properly pronounced by English screen readers. Hence a passage of English text may contain the word "rendezvous" without specifying that its human language is French and still satisfy this Success Criterion. Frequently, when the human language of text appears to be changing for a single word, that word has become part of the language of the surrounding text. Because this is so common in some languages, single words should be considered part of the language of the surrounding text unless it is clear that a change in language was intended. If there is doubt whether a change in language is intended, consider whether the word would be pronounced the same (except for accent or intonation) in the language of the immediately surrounding text.

Most professions require frequent use of technical terms which may originate from a foreign language. Such terms are usually not translated to all languages. The universal nature of technical terms also facilitate communication between professionals.

Some common examples of technical terms include: Homo sapiens, Alpha Centauri, hertz, and habeas corpus.

Identifying changes in language is important for a number of reasons:

- It allows braille translation software to follow changes in language, e.g., substitute control codes for accented characters, and insert control codes necessary to prevent erroneous creation of Grade 2 braille contractions.
- Speech synthesizers that support multiple languages will be able to speak the text in the appropriate accent with proper pronunciation. If changes are not marked, the synthesizer will try its best to speak the words in the default language it works in. Thus, the French word for car, "voiture" would be pronounced "voyture" by a speech synthesizer that uses English as its default language.
- Marking changes in language can benefit future developments in technology, for example users who are unable to translate between languages themselves will be able to use machines to translate unfamiliar languages.
- Marking changes in language can also assist user agents in providing definitions using a dictionary.

BENEFITS

This Success Criterion helps:

- people who use screen readers or other technologies that convert text into synthetic speech;
- people who find it difficult to read written material with fluency and accuracy, such as recognizing characters and alphabets, decoding words, and understanding words and phrases;
- people with certain cognitive, language and learning disabilities who use text-to-speech software;
- people who rely on captions to recognize language changes in the soundtrack of synchronized media content.

EXAMPLES

1. **A German phrase in an English sentence.**
   In the sentence, "He maintained that the DDR (German Democratic Republic) was just a 'Treppenwitz der Weltgeschichte'," the German phrase 'Treppenwitz der Weltgeschichte' is marked as German. Depending on the markup language, English may either be marked as the language for the entire document except where specified, or marked at the paragraph level. When a screen reader encounters the German phrase, it changes pronunciation rules from English to German to pronounce the word correctly.
2. **Alternative language links**
   An HTML Web page includes links to versions of the page in other languages (e.g., Deutsch, Français, Nederlands, Castellano, etc.). The text of each link is the name of the language, in that language. The language of each link is indicated via a lang attribute.
3. **"Podcast" used in a French sentence.**
   Because "podcast" is part of the vernacular of the immediately surrounding text in the following excerpt, "À l'occasion de l'exposition "Energie éternelle. 1500 ans d'art indien", le Palais des Beaux-Arts de Bruxelles a lancé son premier podcast. Vous pouvez télécharger ce podcast au format M4A et MP3," no indication of language change is required.

RELATED RESOURCES

- Language tags in HTML and XML - W3C Internationalization Working Group
- Internationalization Best Practices: Specifying Language in XHTML & HTML Content

TECHNIQUES

Sufficient Techniques

- H58: Using language attributes to identify changes in the human language
- FLASH13: Using HTML language attributes to specify language in Flash content
- PDF19: Specifying the language for a passage or phrase with the Lang entry in PDF documents
- SL4: Declaring Discrete Silverlight Objects to Specify Language Parts in the HTML DOM

Advisory Techniques

- SL27: Use Language/Culture Properties as Exposed by Silverlight Applications and Assistive Technologies
- Make text that is not in the default human language of the Web page visually distinct
- Give the names of any languages used in foreign passages or phrases
- Provide language markup on proper names to facilitate correct pronunciation by screen readers

3.2 PREDICTABLE

Make Web pages appear and operate in predictable ways. It is advisable to position labels to maximize predictability of relationships to comply with this guideline.

3.2.1 – ON FOCUS

When any component receives focus, it does not initiate a change of context.

UNDERSTANDING

The intent of this Success Criterion is to ensure that functionality is predictable as visitors navigate their way through a document. Any component that is able to trigger an event when it receives focus must not change the context. Examples of changing context when a component receives focus include, but are not limited to:

- forms submitted automatically when a component receives focus;
- new windows launched when a component receives focus;
- focus is changed to another component when that component receives focus;

Focus may be moved to a control either via the keyboard (e.g. tabbing to a control) or the mouse (e.g. clicking on a text field). Moving the mouse over a control does not move the focus unless scripting implements this behavior. Note that for some types of controls, clicking on a control may also activate the control (e.g. button), which may, in turn, initiate a change in context.

What is meant by "component" here is also sometimes called "user interface element" or "user interface component''.

BENEFITS

- This Success Criterion helps people with visual disabilities, cognitive limitations, and motor impairments by reducing the chance that a change of context will occur unexpectedly.

EXAMPLES

- **Example 1: A dropdown menu**
  A dropdown menu on a page allows users to choose between jump destinations. If the person uses the keyboard to move down to a choice and activates it (with a spacebar or enter key) it will jump to a new page. However, if the person moves down to a choice and either hits the escape or the tab key to move out of the pulldown menu – it does not jump to a new screen as the focus shifts out of the dropdown menu.
- **Example of a Failure: A help dialog**
  When a field receives focus, a help dialog window describing the field and providing options opens. As a keyboard user tabs through the Web page, the dialog opens, moving the keyboard focus away from the control every time the user attempts to tab past the field.

TECHNIQUES

Sufficient Techniques

- G107: Using "activate" rather than "focus" as a trigger for changes of context

NOTE: A change of content is not always a change of context. This success criterion is automatically met if changes in content are not also changes of context.

Advisory Techniques
Not causing persistent changes of state or value when a component receives focus, or providing an alternate means to reset any changes

- G200: Open new windows and tabs from a link only when necessary
- G201: Give users advanced warning when opening a new window

Failures

- F52: Failure of Success Criterion 3.2.1 and 3.2.5 due to opening a new window as soon as a new page is loaded
- F55: Failure of Success Criteria 2.1.1, 2.4.7, and 3.2.1 due to using script to remove focus when focus is received

3.2.2 – ON INPUT

Changing the setting of any user interface component does not automatically cause a change of context unless the user has been advised of the behavior before using the component.

UNDERSTANDING

The intent of this Success Criterion is to ensure that entering data or selecting a form control has predictable effects. Changing the setting of any user interface component is changing some aspect in the control that will persist when the user is no longer interacting with it. So checking a checkbox, entering text into a text field, or changing the selected option in a list control changes its setting, but activating a link or a button does not. Changes in context can confuse users who do not easily perceive the change or are easily distracted by changes. Changes of context are appropriate only when it is clear that such a change will happen in response to the user's action.

This Success Criterion covers changes in context due to changing the setting of a control. Clicking on links or tabs in a tab control is activating the control, not changing the setting of that control.

What is meant by "component" and "user interface component" here is also sometimes called "user interface element".

BENEFITS

- This Success Criterion helps users with disabilities by making interactive content more predictable. Unexpected changes of context can be so disorienting for users with visual disabilities or cognitive limitations that they are unable to use the content.
- Individuals who are unable to detect changes of context are less likely to become disoriented while navigating a site. For example:
    - Individuals who are blind or have low vision may have difficulty knowing when a visual context change has occurred, such as a new window popping up. In this case, warning users of context changes in advance minimizes confusion when the user discovers that the back button no longer behaves as expected.
- Some individuals with low vision, with reading and intellectual disabilities, and others who have difficulty interpreting visual cues may benefit from additional cues in order to detect changes of context.

EXAMPLES

- A form is provided for creating calendar entries in a Web based calendaring and scheduling application. Along with the standard fields for subject, time and location, there is a set of radio buttons to select the type of calendar entry to create. The calendar entry type can be meeting, appointment or reminder. If the user selects the radio for meeting, additional fields are displayed on the page for entering the meeting participants. Different fields appear if the reminder button is chosen. Because only parts of the entry change and the overall structure remains the same the basic context remains for the user.
- A form contains fields representing US phone numbers. All of the numbers have a three digit area code followed by a three digit prefix and finally a four digit number, and each part of the phone number is entered into a separate field. When the user completes the entry of one field the focus automatically moves to the next field of the phone number. This behavior of phone fields is described for the user at the beginning of the form.

TECHNIQUES

Sufficient Techniques

- G80: Providing a submit button to initiate a change of context *using one of the following techniques:*
  - H32: Providing submit buttons
  - H84: Using a button with a select element to perform an action
  - FLASH4: Providing submit buttons in Flash
  - PDF15: Providing submit buttons with the submit-form action in PDF forms
  - SL10: Implementing a Submit-Form Pattern in Silverlight
- G13: Describing what will happen before a change to a form control that causes a change of context to occur is made
- SCR19: Using an onchange event on a select element without causing a change of context

Note: A change of content is not always a change of context. This success criterion is automatically met if changes in content are not also changes of context.

Advisory Techniques

- G201: Giving users advanced warning when opening a new window

Failures

- F36: Failure of Success Criterion 3.2.2 due to automatically submitting a form and presenting new content without prior warning when the last field in the form is given a value
- F37: Failure of Success Criterion 3.2.2 due to launching a new window without prior warning when the selection of a radio button, check box or select list is changed

3.2.3 – CONSISTENT NAVIGATION

Navigational mechanisms that are repeated on multiple Web pages within a set of Web pages occur in the same relative order each time they are repeated, unless a change is initiated by the user.

UNDERSTANDING

The intent of this Success Criterion is to encourage the use of consistent presentation and layout for users who interact with repeated content within a set of Web pages and need to locate specific information or functionality more than once. Individuals with low vision who use screen magnification to display a small portion of the screen at a time often use visual cues and page boundaries to quickly locate repeated content. Presenting repeated content in the same order is also important for visual users who use spatial memory or visual cues within the design to locate repeated content.

It is important to note that the use of the phrase "same order" in this section is not meant to imply that subnavigation menus cannot be used or that blocks of secondary navigation or page structure cannot be used. Instead, this Success Criterion is intended to assist users who interact with repeated content across Web pages to be able to predict the location of the content they are looking for and find it more quickly when they encounter it again.

Users may initiate a change in the order by using adaptive user agents or by setting preferences so that the information is presented in a way that is most useful to them.

BENEFITS

- Ensuring that repeated components occur in the same order on each page of a site helps users become comfortable that they will able to predict where they can find things on each page. This helps users with cognitive limitations, users with low vision, users with intellectual disabilities, and also those who are blind.

EXAMPLES

- **A consistently located control**
  A search field is the last item on every Web page in a site. users can quickly locate the search function.
- **An expanding navigation menu**
  A navigation menu includes a list of seven items with links to the main sections of a site. When a user selects one of these items, a list of subnavigation items is inserted into the top-level navigation menu.
- **Consistently positioned skip navigation controls**
  A "skip navigation" (or "skip to main content") link is included as the first link on every page in a Web site. The link allows users to quickly bypass heading information and navigational content and begin interacting with the main content of a page.
- **Skip to navigation link**
  Navigational content is consistently located at the end of each page in a set of Web pages. A "skip to navigation" link is consistently located at the beginning of each page so that keyboard users can easily locate it when needed.

RELATED RESOURCES

- Detweiler, M.C. and Omanson, R.C. (1996), Ameritech Web Page User Interface Standards and Design Guidelines.
- Understanding disability issues when designing Web sites.

TECHNIQUES

Sufficient Techniques

- G61: Presenting repeated components in the same relative order each time they appear

Advisory Techniques

- PDF14: Provide running headers and footers in PDF documents
- PDF17: Specify consistent page numbering for PDF documents
- Use templates to ensure consistency across multiple Web pages
- Create layout, positioning, layering, and alignment

Failures

- F66: Failure of Success Criterion 3.2.3 due to presenting navigation links in a different relative order on different pages

3.2.4 – CONSISTENT IDENTIFICATION

Components that have the same functionality within a set of Web pages are identified consistently.

UNDERSTANDING

The intent of this Success Criterion is to ensure consistent identification of functional components that appear repeatedly within a set of Web pages. A strategy that people who use screen readers use when operating a Web site is to rely heavily on their familiarity with functions that may appear on different Web pages. If identical functions have different labels on different Web pages, the site will be considerably more difficult to use. It may also be confusing and increase the cognitive load for people with cognitive limitations. Therefore, consistent labeling will help.

This consistency extends to the text alternatives. If icons or other non-text items have the same functionality, then their text alternatives should be consistent as well.

If there are two components on a web page that both have the same functionality as a component on another page in a set of web pages, then all 3 must be consistent. Hence the two on the same page will be consistent.

While it is desirable and best practice always to be consistent within a single web page, 3.2.4 only addresses consistency within a set of web pages where something is repeated on more than one page in the set.

BENEFITS

- People who learn functionality on one page on a site can find the desired functions on other pages if they are present.
- When non-text content is used in a consistent way to identify components with the same functionality, people with difficulty reading text or detecting text alternatives can interact with the Web without depending on text alternatives.
- People who depend on text alternatives can have a more predictable experience. They can also search for the component if it has a consistent label on different pages.

EXAMPLES

- **Example 1: Document Icon**

A document icon is used to indicate document download throughout a site. The text alternative for the icon always begins with the word “Download," followed by a shortened form of the document title. Using different text alternatives to identify document names for different documents is a consistent use of text alternatives.

- **Example 2: Check Mark**
  A check mark icon functions as "approved", on one page but as "included" on another. Since they serve different functions, they have different text alternatives.
- **Example 3: Consistent references to other pages**
  A Web site publishes articles on-line. Each article spans multiple Web pages and each page contains a link to the first page, the next page and the previous page of the article. If the references to the next page read "page 2", "page 3", "page 4" etcetera, the labels are not the same but they are consistent. Therefore, these references are not failures of this Success Criterion.
- **Example 4: Icons with similar functions**
  An e-commerce application uses a printer icon that allows the user to print receipts and invoices. In one part of the application, the printer icon is labeled "Print receipt" and is used to print receipts, while in another part it is labeled "Print invoice" and is used to print invoices. The labeling is consistent ("Print x"), but the labels are different to reflect the different functions of the icons. Therefore, this example does not fail the Success Criterion.
- **Example 5: Save icon**
  A common "save" icon is used through out the site where page save function is provided on multiple Web pages.
- **Example 6: Icon and adjacent link to same destination**
  An icon with alt text and a link are next to each other and go to the same location. The best practice would be to group them into one link as per H2. However if they are visually positioned one above the other but separated in the source, this may not be possible. To meet the Success Criterion, the link text for these two links need only be consistent, not identical. But best practice is to have identical text so that when users encounter the second one, it is clear that it goes to the same place as the first.
- **Example 7: Example of a Failure**
  A submit "search" button on one Web page and a "find" button on another Web page both have a field to enter a term and list topics in the Web site related to the term submitted. In this case, the buttons have the same functionality but are not labeled consistently.

TECHNIQUES

Sufficient Techniques

- G197: Using labels, names, and text alternatives consistently for content that has the same functionality **AND** following the sufficient techniques for Success Criterion 1.1.1 and sufficient techniques for Success Criterion 4.1.2 for providing labels, names, and text alternatives

Note 1: Text alternatives that are “consistent” are not always “identical.” For instance, you may have an graphical arrow at the bottom of a Web page that links to the next Web page. The text alternative may say “Go to page 4.” Naturally, it would not be appropriate to repeat this exact text alternative on the next Web page. It would be more appropriate to say “Go to page 5”. Although these text alternatives would not be identical, they would be consistent, and therefore would satisfy this Success Criterion.

Note 2: A single non-text-content-item may be used to serve different functions. In such cases, different text alternatives are necessary and should be used. Examples can be commonly found with the use of icons such as check marks, cross marks, and traffic signs. Their functions can be different depending on the context of the Web page. A check mark icon may function as “approved”, “completed”, or “included”, to name a few, depending on the situation. Using “check mark” as text alternative across all Web pages does not help users understand the function of the icon. Different text alternatives can be used when the same non-text content serves multiple functions.

Advisory Techniques

- It is advisable to ensure the text alternative conveys the function of the component and what will happen when the user activates it
- It is advisable to use the same non-text content for a given function whenever possible

Failures

- Failure F31: Failure of Success Criterion 3.2.4 due to using two different labels for the same function on different Web pages within a set of Web pages

3.3 INPUT ASSISTANCE

Help users avoid and correct mistakes.

3.3.1 – ERROR IDENTIFICATION

If an input error is automatically detected, the item that is in error is identified and the error is described to the user in text.

UNDERSTANDING

The intent of this Success Criterion is to ensure that users are aware that an error has occurred and can determine what is wrong. The error message should be as specific as possible. In the case of an unsuccessful form submission, re-displaying the form and indicating the fields in error is insufficient for some users to perceive that an error has occurred. Screen reader users, for example, will not know there was an error until they encounter one of the indicators. They may abandon the form altogether before encountering the error indicator, thinking that the page simply is not functional. Per the definition in WCAG 2.0, an "input error" is information provided by the user that is not accepted. This includes:

- information that is required by the web page but omitted by the user, or
- information that is provided by the user but that falls outside the required data format or allowed values.

For example:

- the user fails to enter the proper abbreviation in to state, province, region, etc. field;
- the user enters a state abbreviation that is not a valid state;
- the user enters a non existent zip or postal code;
- the user enters a birth date 2 years in the future;
- the user enters alphabetic characters or parentheses into their phone number field that only accepts numbers;
- the user enters a bid that is below the previous bid or the minimum bid increment.

If a user enters a value that is too high or too low, and the coding on the page automatically changes that value to fall within the allowed range, the user's error would still need to be described to them as required by the success criterion. Such an error description telling the person of the changed value would meet both this success criterion (Error Identification) and Success Criterion 3.3.3 (Error Suggestion).

The identification and description of an error can be combined with programmatic information that user agents or assistive technologies can use to identify an error and provide error information to the user. For example, certain technologies can specify that the user's input must not fall outside a specific range, or that a form field is required. Currently, few technologies support this kind of programmatic information, but the Success Criterion does not require, nor prevent it.

It is perfectly acceptable to indicate the error in other ways such as image, color etc, in addition to the text description.

See also 3.3.1: Error Suggestion.

BENEFITS

- Providing information about input errors in text allows users who are blind or colorblind to perceive the fact that an error occurred.
- This Success Criterion may help people with cognitive, language, and learning disabilities who have difficulty understanding the meaning represented by icons and other visual cues.

EXAMPLES

- Identifying errors in a form submission
  An airline Web site offers a special promotion on discounted flights. The user is asked to complete a simple form that asks for personal information such as name, address, phone number, seating preference and e-mail address. If any of the fields of the form are either not completed or completed incorrectly, an alert is displayed notifying the user which field or fields were missing or incorrect.

  Note: This Success Criterion does not mean that color or text styles cannot be used to indicate errors. It simply requires that errors also be identified using text. In this example, two asterisks are used in addition to color.

- Providing multiple cues
  The user fails to fill in two fields on the form. In addition to describing the error and providing a unique character to make it easy to search for the fields, the fields are highlighted in yellow to make it easier to visually search for them as well.

TECHNIQUES

Sufficient Techniques

Situation A: If a form contains fields for which information from the user is mandatory.

- G83: Providing text descriptions to identify required fields that were not completed
- ARIA21: Using Aria-Invalid to Indicate An Error Field
- SCR18: Providing client-side validation and alert
- PDF5: Indicating required form controls in PDF forms
- SL35: Using the Validation and ValidationSummary APIs to Implement Client Side Forms Validation in Silverlight

Situation B: If information provided by the user is required to be in a specific data format or of certain values.

- ARIA18: Using aria-alertdialog to Identify Errors
- ARIA19: Using ARIA role=alert or Live Regions to Identify Errors
- ARIA21: Using Aria-Invalid to Indicate An Error Field
- G84: Providing a text description when the user provides information that is not in the list of allowed values
- G85: Providing a text description when user input falls outside the required format or values
- SCR18: Providing client-side validation and alert
- SCR32: Providing client-side validation and adding error text via the DOM
- FLASH12: Providing client-side validation and adding error text via the accessible description
- PDF22: Indicating when user input falls outside the required format or values in PDF forms
- SL35: Using the Validation and ValidationSummary APIs to Implement Client Side Forms Validation in Silverlight

Advisory Techniques

- G139: Create a mechanism that allows users to jump to errors
- Validate form submissions on the server
- Re-display a form with a summary of errors
- Provide error notification as the user enters information
- Include error notification information in the page title
- Highlight or visually emphasizing errors where they occur
- Supplement text with non-text content when reporting errors
- G199: Provide success feedback when data is submitted successfully
- Use sounds to focus user's attention

3.3.2 – LABELS OR INSTRUCTIONS

Labels or instructions are provided when content requires user input.

UNDERSTANDING

The intent of this success criterion is to have content authors place instructions or labels that identify the controls in a form so that users know what input data is expected. Instructions or labels may also specify data formats for fields especially if they are out of the customary formats or if there are specific rules for correct input. Content authors may also choose to make such instructions available to users only when the individual control has focus especially when instructions are long and verbose.

The intent of this Success Criterion is not to clutter the page with unnecessary information but to provide important cues and instructions that will benefit people with disabilities. Too much information or instruction can be just as much of a hindrance as too little. The goal is to make certain that enough information is provided for the user to accomplish the task without undue confusion or navigation.

When labels are provided for input objects, the input object's relationship to the label (or to redundant text serving as the label) must be programmatically determinable or available in text per 1.3.1: Info and Relationships.

BENEFITS

- When label elements are associated with input elements the label is spoken by screen readers when the field receives focus and users with impaired motor control are helped by a larger clickable area for the control, since clicking on the label or the control will activate the control.
- Field labels located in close proximity to the associated field assist users of screen magnifiers because the field and label are more likely to visible within the magnified area of the page.
- Providing examples of expected data formats help users with cognitive, language and learning disabilities to enter information correctly.
- Clearly identifying required fields prevents a keyboard only user from submitting an incomplete form and having to navigate the redisplayed form to find the uncompleted field and provide the missing information.

EXAMPLES

- A field which requires the user to enter the two character abbreviation for a US state has a link next to it which will popup an alphabetized list of state names and the correct abbreviation.

- A field for entering a date contains initial text which indicates the correct format for the date.
- A field for entering a given name is clearly labeled with "Given Name" and the field for family name is labeled "Family Name" to avoid confusion over which name is requested.
- A U.S. phone number separates the area code, exchange, and number into three fields. Parentheses surround the area code field, and a dash separates the exchange and number fields. While the punctuation provides visual clues to those familiar with the U.S. telephone number format, the punctuation is not sufficient to label the fields. The single "Phone number" label also cannot label all three fields. To address this, the three fields are grouped in a fieldset with the legend "Phone number". Visual labels for the fields (beyond the punctuation) cannot be provided in the design, so invisible labels are provided with the "title" attribute to each of the three fields. The value of this attribute for the three fields are, respectively, "Area Code", "Exchange", and "Number".

TECHNIQUES

Sufficient Techniques

- G131: Providing descriptive labels *using one of the following techniques:*

    - ARIA1: Using the aria-describedby property to provide a descriptive label for user interface controls
    - ARIA9: Using aria-labelledby to concatenate a label from several text nodes
    - ARIA17: Using grouping roles to identify related form controls
    - G89: Providing expected data format and example
    - G184: Providing text instructions at the beginning of a form or set of fields that describes the necessary input
    - G162: Positioning labels to maximize predictability of relationships
    - G83: Providing text descriptions to identify required fields that were not completed
    - H90: Indicating required form controls using label or legend
    - FLASH10: Indicating required form controls in Flash
    - PDF5: Indicating required form controls in PDF forms

- H44: Using label elements to associate text labels with form controls
- FLASH32: Using auto labeling to associate text labels with form controls
- FLASH29: Setting the label property for form components
- FLASH25: Labeling a form control by setting its accessible name
- PDF10: Providing labels for interactive form controls in PDF documents
- SL26: Using LabeledBy to Associate Labels and Targets in Silverlight
- H71: Providing a description for groups of form controls using fieldset and legend elements
- FLASH8: Adding a group name to the accessible name of a form control
- H65: Using the title attribute to identify form controls when the label element cannot be used
- SL8: Displaying HelpText in Silverlight User Interfaces
- G167: Using an adjacent button to label the purpose of a field

Note: The techniques at the end of the above list should be considered “last resort” and only used when the other techniques cannot be applied to the page. The earlier techniques are preferred because they increase accessibility to a wider user group.

Advisory Techniques

- G13: Describing what will happen before a change to a form control that causes a change of context to occur is made
- SL19: Providing User Instructions With AutomationProperties.HelpText in Silverlight
- Providing linear form design and grouping similar items

Failures

- F82: Failure of Success Criterion 3.3.2 by visually formatting a set of phone number fields but not including a text label

3.3.3 – ERROR SUGGESTION

If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content.

UNDERSTANDING

The intent of this Success Criterion is to ensure that users receive appropriate suggestions for correction of an input error if it is possible. The WCAG 2.0 definition of "input error" says that it is "information provided by the user that is not accepted" by the system. Some examples of information that is not accepted include information that is required but omitted by the user and information that is provided by the user but that falls outside the required data format or allowed values.

Success Criterion 3.3.1 provides for notification of errors. However, persons with cognitive limitations may find it difficult to understand how to correct the errors. People with visual disabilities may not be able to figure out exactly how to correct the error. In the case of an unsuccessful form submission, users may abandon the form because they may be unsure of how to correct the error even though they are aware that it has occurred.

The content author may provide the description of the error, or the user agent may provide the description of the error based on technology-specific, programmatically determined information.

BENEFITS

- Providing information about how to correct input errors allows users who have learning disabilities to fill in a form successfully. Users who are blind or have impaired vision understand more easily the nature of the input error and how to correct it. People with motion impairment can reduce the number of times they need to change an input value.

EXAMPLES

- **Additional Help for Correcting An Input Error**
  The result of a form that was not successfully submitted describes an input error in place in the page along with the correct input and offers additional help for the form field that caused the input error.
- **Suggestions from a Limited Set of Values**
  An input field requires that a month name be entered. If the user enters "12," suggestions for correction may include
  - A list of the acceptable values, e.g., "Choose one of: January, February, March, April, May, June, July, August, September, October, November, December."
  - A description of the set of values, e.g., "Please provide the name of the month."
  - The conversion of the input data interpreted as a different month format, e.g., "Do you mean 'December'?"

TECHNIQUES

Sufficient Techniques

Situation A: If a mandatory field contains no information:

- G83: Providing text descriptions to identify required fields that were not completed
- ARIA2: Identifying a required field with the aria-required property
- PDF5: Indicating required form controls in PDF forms
- SL35: Using the Validation and ValidationSummary APIs to Implement Client Side Forms Validation in Silverlight

Situation B: If information for a specific data field is required to be in a specific data format:

- ARIA18: Using aria-alertdialog to Identify Errors
- G85: Providing a text description when user input falls outside the required format or values
- G177: Providing suggested correction text
- SCR18: Providing client-side validation and alert
- SCR32: Providing client-side validation and adding error text via the DOM
- FLASH12: Providing client-side validation and adding error text via the accessible description
- PDF22: Indicating when user input falls outside the required format or values in PDF forms

Situation C: Information provided by the user is required to be one of a limited set of values:

- ARIA18: Using aria-alertdialog to Identify Errors
- G84: Providing a text description when the user provides information that is not in the list of allowed values
- G177: Providing suggested correction text
- SCR18: Providing client-side validation and alert
- SCR32: Providing client-side validation and adding error text via the DOM
- FLASH12: Providing client-side validation and adding error text via the accessible description
- PDF22: Indicating when user input falls outside the required format or values in PDF forms

Advisory Techniques

- G139: Creating a mechanism that allows users to jump to errors
- Making error messages easy to understand and distinguishable from other text in the Web page
- Validating form submissions on the server
- When mandatory information has not been provided, including descriptions or examples of correct information in addition to identifying the field as mandatory
- Repeating and emphasizing suggestions for correcting each input error in the context of its form field
- Providing a way for the user to skip from each item in a list of suggestions to its corresponding form field
- Providing additional contextual help for the form field requiring change
- Accepting input data in a variety of formats
- G199: Providing success feedback when data is submitted successfully
  - Advisory techniques for providing suggestions to the user:
    - Providing a text description that contains information about the number of input errors, suggestions for corrections to each item, and instructions on how to proceed
    - Providing a text description that contains suggestions for correction as the first item (or one of the first items) of content, or emphasizing this information in the content

    - Displaying errors and suggestions in the context of the original form (for example, re-displaying a form where input errors and suggestions for correction are highlighted and displayed in the context of the original form)
- Advisory HTML techniques:
    - Providing "correct examples" for data and data formats as initial text in mandatory form fields
    - Providing links to suggested correction text "close to" form fields, or providing the suggested correction text itself directly on the Web page "next to" form fields
- Advisory client-side scripting techniques
    - SCR18: Providing client-side validation and alert
    - Providing client-side validation and adding error text via the DOM
    - Calling a function from the submit action of a form to perform client side validation

3.3.4 – ERROR PREVENTION (LEGAL, FINANCIAL, DATA)

For Web pages that cause legal commitments or financial transactions for the user to occur, that modify or delete user-controllable data in data storage systems, or that submit user test responses, at least one of the following is true:

Reversible: Submissions are reversible.

Checked: Data entered by the user is checked for input errors and the user is provided an opportunity to correct them.

Confirmed: A mechanism is available for reviewing, confirming, and correcting information before finalizing the submission.

UNDERSTANDING

The intent of this Success Criterion is to help users with disabilities avoid serious consequences as the result of a mistake when performing an action that cannot be reversed. For example, purchasing non-refundable airline tickets or submitting an order to purchase stock in a brokerage account are financial transactions with serious consequences. If a user has made a mistake on the date of air travel, he or she could end up with a ticket for the wrong day that cannot be exchanged. If the user made a mistake on the number of stock shares to be purchased, he or she could end up purchasing more stock than intended. Both of these types of mistakes involve transactions that take place immediately and cannot be altered afterwards, and can be very costly. Likewise, it may be an unrecoverable error if users unintentionally modify or delete data stored in a database that they later need to access, such as their entire travel profile in a travel services web site. When referring to modification or deletion of 'user controllable' data, the intent is to prevent mass loss of data such as deleting a file or record. It is not the intent to require a confirmation for each save command or the simple creation or editing of documents, records or other data.

Users with disabilities may be more likely to make mistakes. People with reading disabilities may transpose numbers and letters, and those with motor disabilities may hit keys by mistake. Providing the ability to reverse actions allows users to correct a mistake that could result in serious consequences. Providing the ability to review and correct information gives the user an opportunity to detect a mistake before taking an action that has serious consequences.

User-controllable data is user-viewable data that the user can change and/or delete through an intentional action. Examples of the user controlling such data would be updating the phone number and address for the user's account, or deleting a record of past invoices from a website. It does not refer such things as internet logs and search engine monitoring data that the user can't view or interact with directly.

BENEFITS

- Providing safeguards to avoid serious consequences resulting from mistakes helps users with all disabilities who may be more likely to make mistakes.

EXAMPLES

- Order confirmation:
  A Web retailer offers on-line shopping for customers. When an order is submitted, the order information—including items ordered, quantity of each ordered item, shipping address, and payment method—are displayed so that the user can inspect the order for correctness. The user can either confirm the order or make changes.

- Stock sale:
  A financial services Web site lets users buy and sell stock online. When a user submits an order to buy or sell stock, the system checks to see whether or not the market is open. If it is after hours, the user is alerted that the transaction will be an after-hours transaction, is told about the risks of trading outside of regular market hours, and given the opportunity to cancel or confirm the order.

TECHNIQUES

Sufficient Techniques

Situation A: If an application causes a legal transaction to occur, such as making a purchase or submitting an income tax return:

- G164: Providing a stated time within which an online request (or transaction) may be amended or canceled by the user after making the request
- G98: Providing the ability for the user to review and correct answers before submitting
- G155: Providing a checkbox in addition to a submit button

Situation B: If an action causes information to be deleted:

- G99: Providing the ability to recover deleted information
- G168: Requesting confirmation to continue with selected action
- G155: Providing a checkbox in addition to a submit button

Situation C: If the Web page includes a testing application:

- G98: Providing the ability for the user to review and correct answers before submitting
- G168: Requesting confirmation to continue with selected action

Advisory Techniques

- Informing the user what irreversible action is about to happen
- SCR18: Providing client-side validation and alert

- SL35: Using the Validation and ValidationSummary APIs to Implement Client Side Forms Validation in Silverlight
- Placing focus in the field containing the error
- Avoiding use of the same words or letter combinations to begin each item of a drop-down list
- G199: Providing success feedback when data is submitted successfully

4. ROBUST

Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

4.1 COMPATIBLE

Maximize compatibility with current and future user agents, including assistive technologies.

It is advisable to avoid deprecated features of W3C technologies and to avoid displaying content that relies on technologies that are not accessibility-supported when the technology is turned off or not supported.

4.1.1 – PARSING

In content implemented using markup languages, elements have complete start and end tags, elements are nested according to their specifications, elements do not contain duplicate attributes, and any IDs are unique, except where the specifications allow these features.

Note: Start and end tags that are missing a critical character in their formation, such as a closing angle bracket or a mismatched attribute value quotation mark are not complete.

UNDERSTANDING

The intent of this Success Criterion is to ensure that user agents, including assistive technologies, can accurately interpret and parse content. If the content cannot be parsed into a data structure, then different user agents may present it differently or be completely unable to parse it. Some user agents use "repair techniques" to render poorly coded content.

Since repair techniques vary among user agents, authors cannot assume that content will be accurately parsed into a data structure or that it will be rendered correctly by specialized user agents, including assistive technologies, unless the content is created according to the rules defined in the formal grammar for that technology. In markup languages, errors in element and attribute syntax and failure to provide properly nested start/end tags lead to errors that prevent user agents from parsing the content reliably. Therefore, the Success Criterion requires that the content can be parsed using only the rules of the formal grammar.

The concept of "well formed" is close to what is required here. However, exact parsing requirements vary amongst markup languages, and most non XML-based languages do not explicitly define requirements for well formedness. Therefore, it was necessary to be more explicit in the success criterion in order to be generally applicable to markup languages. Because the term "well formed" is only defined in XML, and (because end tags are sometimes optional) valid HTML does not require well formed code, the term is not used in this success criterion.

With the exception of one success criterion (1.4.2: Resize text, which specifically mentions that the effect specified by the success criterion must be achieved without relying on an assistive technology) authors can meet the success criteria with content that assumes use of an assistive technology (or access features in use agents) by the user, where such assistive technologies (or access features in user agents) exist and are available to the user.

BENEFITS

- Ensuring that Web pages have complete start and end tags and are nested according to specification helps ensure that assistive technologies can parse the content accurately and without crashing.

TECHNIQUES

Sufficient Techniques

- G134: Validating Web pages
- G192: Fully conforming to specifications
- H88: Using HTML according to spec
- Ensuring that Web pages can be parsed *using one of the following techniques:*
    - H74: Ensuring that opening and closing tags are used according to specification **AND** H93: Ensuring that id attributes are unique on a Web page **AND** H94: Ensuring that elements do not contain duplicate attributes
    - H75: Ensuring that Web pages are well-formed
- SL33: Using Well-Formed XAML to Define a Silverlight User Interface

Failures

- F70: Failure of Success Criterion 4.1.1 due to incorrect use of start and end tags or attribute markup
- F77: Failure of Success Criterion 4.1.1 due to duplicate values of type ID

4.1.2 – NAME, ROLE, VALUE

For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.

Note: This success criterion is primarily for Web authors who develop or script their own user interface components. For example, standard HTML controls already meet this success criterion when used according to specification.

UNDERSTANDING

The intent of this Success Criterion is to ensure that Assistive Technologies (AT) can gather information about, activate(or set) and keep up to date on the status of user interface controls in the content.

When standard controls from accessible technologies are used, this process is straightforward. If the user interface elements are used according to specification the conditions of this provision will be met. (See examples of Success Criterion 4.1.2 below)

If custom controls are created, however, or interface elements are programmed (in code or script) to have a different role and/or function than usual, then additional measures need to be taken to ensure that the controls provide important information to assistive technologies and allow themselves to be controlled by assistive technologies.

A particularly important state of a user interface control is whether or not it has focus. The focus state of a control can be programmatically determined, and notifications about change of focus are sent to user agents and assistive technology. Other examples of user interface control state are whether or not a checkbox or radio button has been selected, or whether or not a collapsible tree or list node is expanded or collapsed.

Success Criterion 4.1.2 requires a programmatically determinable name for all user interface components. Names may be visible or invisible. Occasionally, the name must be visible, in which case it is identified as a label. Refer to the definition of name and label in the glossary for more information.

BENEFITS

- Providing role, state, and value information on all user interface components enables compatibility with assistive technology, such as screen readers, screen magnifiers, and speech recognition software, used by people with disabilities.

EXAMPLES

- Accessible APIs – A Java applet uses the accessibility API defined by the language.

RELATED RESOURCES

- Dynamic Accessible Web Content Roadmap
- Role Taxonomy for Accessible Adaptable Applications
- States and Adaptable Properties Module
- Microsoft Active Accessibility, Version 2.0
- Adobe Flash accessibility design guidelines

TECHNIQUES

Sufficient Techniques

Situation A: If there are session time limits:

- G133: Providing a checkbox on the first page of a multipart form that allows users to ask for longer session time limit or no session time limit
- G198: Providing a way for the user to turn the time limit off

Situation B: If a time limit is controlled by a script on the page:

- G198: Providing a way for the user to turn the time limit off
- G180: Providing the user with a means to set the time limit to 10 times the default time limit
- SCR16: Providing a script that warns the user a time limit is about to expire **AND** SCR1: Allowing the user to extend the default time limit
- FLASH19: Providing a script that warns the user a time limit is about to expire and provides a way to extend it
- FLASH24: Allowing the user to extend the default time limit
- SL21: Replacing A Silverlight Timed Animation With a Nonanimated Element

Situation C: If there are time limits on reading:

- G4: Allowing the content to be paused and restarted from where it was paused
- G198: Providing a way for the user to turn the time limit off
- SCR33: Using script to scroll content, and providing a mechanism to pause it
- SCR36: Providing a mechanism to allow users to display moving, scrolling, or auto-updating text in a static window or area

Advisory Techniques

- Using a script to poll the server and notify a user if a time limit is present (Scripting)
- Using sounds to focus user's attention

Failures

- F40: Failure of Success Criterion 2.2.1 and 2.2.4 due to using meta redirect with a time limit
- F41: Failure of Success Criterion 2.2.1, 2.2.4, and 3.2.5 due to using meta refresh to reload the page
- Failure F58: Failure of Success Criterion 2.2.1 due to using server-side techniques to automatically redirect pages after a time-out