

Proceedings of the 2016 ICT Accessibility Testing Symposium: Section 508, WCAG, and beyond

Baltimore, Maryland, USA

November 2 & 3, 2016

Hosted by

The National Federation of the Blind
Center of Excellence in
Nonvisual Access to Education, Public Information, and Commerce (CENA)



www.ictaccessibilitytesting.org

Contents

| | |
|--|-----|
| Introduction from the Chair | 1 |
| Keynote. Testing: A Critical Accelerator for Accessibility, and How to Get There | 3 |
| Workshop. Open Source Tools for Evaluating and Inspecting Web Accessibility of Organizations to Individual Pages..... | 5 |
| Workshop. Trusted Tester Program and Section 508 Compliance in the Agile Development Setting | 9 |
| Panel. ACT Now: Accessibility Conformance Testing for WCAG | 13 |
| How Can Nontechnical Individuals Assist in Web Accessibility Testing? | 17 |
| OpenAjax Evaluation Library for Evaluating WCAG 2.0 Level A and AA Requirements | 21 |
| The Unbearable Inaccessibility of Slideshows | 29 |
| Testing for Accessibility Regulations..... | 39 |
| QA Enabled Testing as the Hook for Organizational Change | 47 |
| Four Things You Should Know Before Testing with VoiceOver on Mac | 57 |
| Writing Accessibility Tests | 63 |
| Providing An Agile, Mobile Touch To Section 508 Testing | 71 |
| Improving Testing for Survey and Exam Accessibility..... | 75 |
| Social Media and Accessibility in Employment | 81 |
| A11y Facts: An Accessible Playing Field for Instructional Materials | 93 |
| The Accessibility Switchboard: A New Web Portal Resource and Community of Practice for Tackling Accessibility at the Organizational Level..... | 105 |
| Symposium Committee..... | 113 |
| Author Index | 115 |

Introduction from the Chair



Dr. Chris M. Law

Chair, The 2016 ICT Accessibility Testing Symposium

This event marks the first time that accessibility community has convened to specifically focus on the testing of Information and Communications Technology for accessibility.

Testing is a bedrock element of any accessibility program. Without test results, it is not possible to accurately gauge where you are now, or to track progress towards meeting goals. In recent years, the number of different types of testing tools and testing methods has grown rapidly. This growth reflects the increasing levels of interest in the inclusion of accessibility as an integral part of mainstream ICT development.

With growth comes an abundance of ideas, diversity in approaches, variations in scope, and so forth. In early 2016, we recognized the need for a forum to foster discussion as well as the presentation and exchange of varied ideas relating to testing tools and methods, and the management of testing personnel and processes in organizations. To fulfill this need, the twenty-one member committee came together to discuss the concept of a new event to provide such a forum.

This event is the result of the hard work of the committee. I would like to thank everyone who volunteered their time providing guidance and assessment of initial plans, and for providing peer review of submitted proposals. I would also like to thank Norman Robinson, for his design, build and management of the event website and acting as Co-Chair, and John Rempel for organizing awards. A very big thanks go out to our hosts for this event, the National Federation of the Blind Center of Excellence in Nonvisual Access to Education, Public Information, and Commerce (CENA), specifically Anil Lewis for providing us with the location, and Clara Van Gerven for the logistical support.

We started this event with the goal of starting what would become an annual symposium with practical benefits for members of our community. We hope that you will consider participating in the this endeavor, in order to help us collectively evolve and enhance the accessibility testing field over the coming years.

—Chris Law

Keynote.

Testing: A Critical Accelerator for Accessibility, and How to Get There



Judy Brewer

Director of the Web Accessibility Initiative, at W3C

Judy Brewer directs the Web Accessibility Initiative (WAI) at the World Wide Web Consortium (W3C). Her work includes coordinating development of standards and guidelines for accessibility of the Web, including mobile, digital publishing, and other technologies that are converging on the Web; improvement of resources for evaluation of Web sites; development of education and outreach resources; exploration of research which may impact future Web accessibility; and promoting harmonization of web accessibility standards internationally. WAI standards and guidelines have been taken up by many organizations and governments around the world to help ensure equal access to the Information Society for people with disabilities.

Brewer is a principle research scientist at MIT's Computer Science and Artificial Intelligence Laboratory. Prior to joining W3C, she worked on initiatives to increase access to assistive technology and mainstream information and communications technology. She has a background in management, technical writing, education, applied linguistics, disability advocacy, and biotechnology. Recent awards include the Newell Perry Award from the US National Federation of the Blind in 2014, and the Migel Medal from the American Foundation for the Blind in 2015.

Workshop. Open Source Tools for Evaluating and Inspecting Web Accessibility of Organizations to Individual Pages

Jon Gunderson, PhD.

Disability Resources and Education Services
University of Illinois at Urbana/Champaign
jongund@illinois.edu

Abstract

The workshop will explore the world of open source web accessibility evaluation and inspection tools and their ability to illuminate our current understanding of how to comply with W3C Web Content Accessibility Requirements 2.0 and Section 508 Requirements. The workshop will include an overview of the features of open source tools developed at the University of Illinois based on the OpenAjax Evaluation Library.

Overview of Workshop

The main purpose of the workshop is to explore the utility of open source tools in helping us understand how to make web resources more accessible to people with disabilities. The workshop will begin by demonstrating several open source tools that have been developed at the University of Illinois to support both quality assurance testers, development teams and managers in understanding the accessibility of individual web pages to entire organizations. The tools are based on the requirements of the W3C Web Content Accessibility Guidelines and the accessible authoring practices of the W3C Accessible Rich Internet Applications specification and HTML5. The second half of the workshop will be a discussion with the participants on the direction of open source tools and how they can best benefit the web accessibility community to support accessible design and providing a resource for training the next generation of web developers.

Open Source Tools Demonstrated in the Workshop

AInspector Sidebar for Firefox

AInspector Sidebar for the Firefox browser supports the evaluation of web pages for the accessibility features required by the W3C Web Content Accessibility Guidelines 2.0 Level A and AA Success Criteria. The tool provides both summary information and detailed information on the accessibility requirements that apply to a web page (e.g. WCAG 2.0 requirement filtering). Users can highlight and inspect element information on the page associated for each

requirement. Each result has information on the requirement, techniques that can be used to implement the requirement and links to more information.

Available from Mozilla Add-on Website

<https://addons.mozilla.org/en-us/firefox/addon/ainspector-sidebar/>

Functional Accessibility Evaluator (FAE)

The Functional Accessibility Evaluator (FAE) 2.0 provides accessibility information on an entire website for WCAG 2.0 Level A and AA Success Criteria. FAE 2.0 provides both website summary and page level detail on the accessibility requirements that applied to the pages with in a website. FAE 2.0 uses the same open source OpenAjax Evaluation Library, Rules and Rulesets as AInspector Sidebar. FAE 2.0 and AInspector Sidebar are designed to complement themselves. It can also be used in procurement to help compare the accessibility of competing products.

Functional Accessibility Evaluator 2.0 (Individual Version)

<https://fae.disability.illinois.edu/>

FAE Auditor

The FAE Auditor is design to provide information on an entire organization. Based on the FAE technology Auditor can run multiple websites with in the same organization and allow reports to be compared at a high level and to report on the implementation of rules across the entire organization and within sub-groups of an organization. Websites can be collected in groups and the results within that group can be analyzed and compared to other groups of websites. FAE Auditor is designed to give organizations an overall picture of the accessibility of accessibility of their web resources and help with planning training programs to improve the accessibility knowledge of development groups.

Bookmarklets for Accessibility Visualization

Bookmarklets can be added to any browser to provide visualizations of specific accessibility information that can helps people understand the functional accessibility of a web page by making hidden accessibility information visible to sighted web developers. Current Bookmarklets include:

- ARIA Landmarks
- Headings (H1-H6)
- Lists
- Images
- Forms

NOTE: Browsers have limited the bookmarklets use to non-HTTPS websites, apparently to improve browser security

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Workshop. Trusted Tester Program and Section 508 Compliance in the Agile Development Setting

Allen Hoffman

Department of Homeland Security Office of Accessible Systems and Technology (DHS OAST)
245 Murray Lane SW/Room 4636-14, Mailstop 0135, Washington, DC 20528 USA
accessibility@hq.dhs.gov

Cynthia Clinton-Brown

Department of Homeland Security Office of Accessible Systems and Technology (DHS OAST)
245 Murray Lane SW/Room 4636-14, Mailstop 0135, Washington, DC 20528 USA
accessibility@hq.dhs.gov

Katherine Eng

United States Department of Justice
145 N St, NE; 3W.134B, Washington, DC 20530 USA
Katherine.Eng@usdoj.gov

Kenneth Moser

DHS U.S. Citizen and Immigration Services (USCIS)
245 Murray Lane SW/Room 4636-14, Mailstop 0135, Washington, DC 20528 USA
kenneth.s.moser@uscis.dhs.gov

Abstract

In 2012, The U.S. Department of Homeland Security (DHS) and the U.S. Social Security Administration (SSA) accessibility teams collaborated to create the *Harmonized Testing Process for Section 508 Compliance: Baseline Tests for Software and Web Accessibility for evaluating Section 508 Compliance for Software and Web*. Currently, an interagency committee representing seven federal agencies works together on updating the Baseline Testing document with changes to keep up with current technologies.

The Baseline Tests supports:

- Unified testing for the Section 508 technical requirements for Software and Websites, as well as Functional Performance Criteria (FPC) across government.
- Provides a minimum set of Tests for agencies to streamline/enhance their existing test processes.
- Aligns Section 508 accessibility testing with most of the World Wide Web Consortium (W3C) Web Content Accessibility Guidelines 2.0 (WCAG 2.0) success criteria, in preparation for the upcoming Section 508 refresh.

In the workshop, participants will learn the issues that drove the need to establish common tests for compliance and how the baseline tests were developed.

Overview of Presentation

In a half-day workshop, the following topics will be presented with demonstrations and group activities:

1. What is the harmonized testing process for Section 508 compliance?
2. What is the Trusted Tester Process, Program and certification?
3. What is the Interagency Trusted Tester program?
4. What is a best practice for accessibility management within an agile development organization?

Themes in the activities include:

- Testing tools
- Testing methods
- Management of testing
- Testing trends
- Standards
- Procurement
- Product Development
- Research
- Federal programs
- Section 508
- Section 508 Refresh
- WCAG 2.0

Trusted Tester Program

DHS has developed the Trusted Tester Program to establish Section 508 conformance evaluation skill sets across the IT community, and through a repeatable set of self-assessment and planning activities foster optimal accessibility management practices across the Federal landscape. In the workshop, participants will learn the history and benefits of Trusted Tester training and certification, how to become a Trusted Tester, and how they can participate in the accessibility community by joining the Trusted Tester Community of Practice (TTCoP).

Interagency Trusted Tester Program (ITTP)

To promote the adoption of unified testing with a common set of tools, test procedures, and evaluation criteria across the federal government, DHS is leading the effort to stand up the Interagency Trusted Tester Program (ITTP). ITTP is modeled after the DHS OAST Trusted Tester Program and is a strategic solution with emphasis on:

1. Promoting across the federal government common evaluation processes and procedures for Section 508 testing of electronic and information technology (EIT).
2. Increasing the workforce skillset and capability by implementing a certification program for Section 508 conformance testing.
3. Creating a central repository containing trusted tester test results that can be shared amongst government agencies to reduce redundant testing.

In the workshop, participants will learn about the six stages of the Unified Testing for Accessibility Project (UTAP), that agency stakeholders follow to develop their own design and implementation plans for adopting Trusted Tester. The six stages that will be discussed are listed below:

- | | |
|----------|---|
| Stage 1: | Set your goals |
| Stage 2: | Get familiar with your own agency |
| Stage 3: | Familiarize yourself with available options |
| Stage 4: | Design the system |
| Stage 5: | Develop an implementation plan |
| Stage 6: | Implement and monitor |

Accessibility Management in an Agile Environment

The United States Citizenship and Immigration Service (USCIS) has pioneered reliance on the Trusted Tester skill sets, and established a set of accessibility management practices for agile, even and dev-ops environments to keep pace with rapid development and delivery of IT applications. Practices include:

- An engagement model that matches activities to the level of risk
- Aggressive recruitment of development Trusted Testers
- Development contract changes that strengthen accessibility
- Top Ten Issues training for software developers
- Strategies to move testing to the left in the development cycle
- Close collaboration with DHS OAST to expand testing options

Outcome

The expected outcome for the audience is:

- Awareness of what a code-inspection-based repeatable, accurate, and scalable Section508 conformance test process is composed of;
- Awareness of the Trusted Tester program and certification;
- Awareness of the Interagency Trusted Tester Program and how to plan for and optimize integration of IT accessibility management practices within an organization;
- An understanding of how one organization manages accessibility in the agile development setting.

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Panel. ACT Now: Accessibility Conformance Testing for WCAG

Shadi Abou-Zahra

W3C Web Accessibility Initiative (WAI)
2004, Route des Lucioles BP94, 06902 Sophia-Antipolis, France
shadi@w3.org

Abstract

The W3C Web Content Accessibility Guidelines (WCAG)¹ continues to gain widespread recognition around the world. It is available freely from W3C, is available as ISO/IEC 40500, and is part of the European standard EN 301 549. It is also referenced in many governmental and organizational policies internationally.

While WCAG 2.0 has been designed and written to be far more testable than its prior version, sometimes there are still differences in testing approaches and their resulting interpretations. This includes fully automated and semi-automated testing tools, and testing procedures for manual (human) evaluation.

Recently WCAG Working Group launched the “Accessibility Conformance Testing (ACT) Task Force”², to help develop a specification for web accessibility testing procedures. This envisioned W3C standard should complement WCAG, and allow the development of testing procedures for automated, semi-automated, and manual testing, for a more consistent interpretation of WCAG.

Background

Evaluating the conformance of web content – including dynamic web and mobile applications – to the WAI Web Content Accessibility Guidelines (WCAG) is often a non-trivial task. In particular, some WCAG success criteria are broad and need to be broken-down systematically for evaluation, or they require qualitative analysis within the specific context of the web content being evaluated. Thus, varying interpretations are manifested in evaluation tools and testing methodologies, with often-conflicting results. That is, the same web content might be deemed to have ‘passed’ accessibility requirements by one method, yet ‘failed’ by another. This contributes to confusion within the field. In some cases, it also leads to loss of business opportunities and legal disputes, which is not supportive to the cause of accessibility. It is critical to pursue a common understanding of WCAG as well as to harmonize interpretations, in the context of testing and evaluation practices.

¹ <https://www.w3.org/WAI/intro/wcag>

² <https://www.w3.org/WAI/GL/task-forces/conformance-testing/>

For many years, researchers, tool developers, and individual experts have attempted to address this issue. Several initiatives and research projects have been undertaken in Europe, the United States, and elsewhere. These resulted in a number of different testing approaches, each with their own collection of advantages and disadvantages. However, until recently these activities did not seem to get a lot of traction among different stakeholders. Yet with the growing uptake of web accessibility standards, there is an increased need for common testing approaches within the field. In fact, some organizations requested that the W3C Web Accessibility Initiative (WAI) undertake standardization activity in this area, to provide a vendor-neutral, authoritative, and openly available testing methodology.

Previous Work

Work on tool-supported web accessibility evaluation started soon after the publication of WCAG 1.0 in 1999. Early attempts include the meanwhile obsoleted “Techniques for Accessibility Evaluation and Repair Tools”³, which flowed into the “Techniques for WCAG 2.0”⁴. In 2005 the European Commission funded three research projects to develop the “Unified Web Evaluation Methodology”⁵. This also contributed to the development of “WCAG 2.0 Test Samples”⁶, to help benchmark the accuracy of web accessibility evaluation tools.

Numerous developments have also taken place outside of the W3C. This includes the test rules development of the “Open Ajax Alliance”⁷, “aXe”⁸, and many more, some of which may not be publicly available. Thus, a goal of this new effort is to build on and merge these activities rather than to create another separate approach. The central role of the W3C as the leading body in web accessibility standardization is critical to this purpose.

Current Work

More recently work started in the W3C Community Group “Automated WCAG Monitoring (auto-WCAG)”⁹. Community groups are pre-standardization fora with no formal standing in W3C status. Anyone can join such groups and they serve as important vehicles to collect input from the community and incubate new work. Despite its name, the work of this community group is not limited to automated testing only, but also explores semi-automated testing. The work of this group, including a list of testing procedures, is publicly available¹⁰.

Based on this initial work, representatives of W3C member organizations proposed the creation of a new task force as part of the WCAG Working Group. The purpose of this “WCAG Accessibility Conformance Testing (ACT) Task Force”¹¹ is to standardize the format and the benchmarking approach of the testing procedures, to facilitate the development of a consistent set of procedures that aligns with WCAG 2.0 and beyond. The development of the actual testing

³ <https://www.w3.org/TR/AERT>

⁴ <https://www.w3.org/TR/WCAG20-TECHS/>

⁵ <http://www.wabcluster.org/>

⁶ <https://www.w3.org/WAI/ER/tests/>

⁷ <http://www.openajax.org/>

⁸ <https://www.deque.com/products/axe/>

⁹ <https://www.w3.org/community/auto-wcag/>

¹⁰ https://www.w3.org/community/auto-wcag/wiki/Test_overview

¹¹ <https://www.w3.org/WAI/GL/task-forces/conformance-testing/>

procedures will continue to take place in the auto-WCAG community group, to facilitate participation and contribution by the broadest audience possible.

Together the two groups, auto-WCAG CG and ACT TF, will produce the following deliverables:

1. **ACT Framework:** A W3C Recommendation defining how to write and validate test rules for accessibility testing. [ACT TF]
2. **ACT Benchmark**
 1. **ACT Benchmark Method:** A description of how to test the accuracy of rules (e.g. false positives and false negatives). [ACT TF]
 2. **ACT Benchmark Tool:** Implementation of the benchmark method. [ACT TF]
3. **ACT Rule Suite**
 1. **ACT Rule Suite Repository:** A collection of rules that have passed the validation and benchmarking requirements. [auto-WCAG]
 2. **ACT Rule Suite Frontend:** Website for the above repository. [auto-WCAG]

Panel

This panel will introduce the work of the WCAG Accessibility Conformance Testing (ACT) Task Force and invite feedback from the audience, to help shape the effort. It will highlight opportunities to contribute and to actively participate in the development of internationally recognized testing procedures for WCAG. The panel will also explore some of the key challenges that need to be addressed, including the following questions:

- How to define deterministic algorithms for qualitative accessibility checks? Not all checks can be fully automated, and some test rules are easier to define than others.
- How to manage the sheer amount of possible tests for different web technologies and combinations of situations that need to be checked, especially for web applications?
- How to stay current with evolving web technologies, support for accessibility features in web browsers and assistive technologies, as well as with evolving web design practices?
- How to support organizations to port their existing testing procedures and contribute to a common set of test rules that aligns with an authoritative interpretation of WCAG?

Join us in this panel discussion to learn about the latest developments and to share your views about them. Panelists include:

- Shadi Abou-Zahra, W3C Web Accessibility Initiative (WAI)
- Katie Haritos-Shea, Deque Systems
- Alistair Garrison, SSB BART Group
- Jon Gunderson, University of Illinois
- Kathy Walhbin, Interactive Accessibility

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

How Can Nontechnical Individuals Assist in Web Accessibility Testing?

Cyndi Rowland

WebAIM; National Center on Disability and Access to Education
Utah State University
6801 Old Main Hill
Logan Utah, 84322-6801, USA
Cyndi.Rowland@usu.edu

Jonathan Whiting

WebAIM; National Center on Disability and Access to Education
Utah State University
6801 Old Main Hill
Logan Utah, 84322-6801, USA
Jonathan@WebAIM.org

Abstract

There is little doubt that the need for web accessibility testing has grown logarithmically in the past few years. During this same time, we see increases in nontechnical individuals who engage in testing. This is especially true in education settings. While resources are present to assist nontechnical individuals in the testing process, the field benefits from a broader conversation about the relative advantages and challenges present in this phenomenon.

Introduction

Web accessibility testing is at an all-time high in American society and it will likely increase over the coming decade as awareness of legal and ethical responsibility for web accessibility continues to grow. In many sectors of society, web accessibility is not the sole domain of technical individuals. Nowhere is that seen more than in education. This may be due to the convergence of two critical circumstances. The first is the paucity of technical individuals who can match the evaluation need for postsecondary institutions, or K-12 school districts who often have thousands, or tens of thousands, of pages. The second is a recommendation from professionals in the field that the responsibility for web accessibility should be in the purview of all content creators; this would include faculties and staffs (e.g., Rowland, Whiting, & Smith, 2015). This recommendation maps onto resources available for nontechnical content creators of non-HTML materials (i.e., Word, PowerPoint, PDF, videos); one example of resources are the “[Cheatsheets](http://ncdae.org/resources/cheatsheets/)” (<http://ncdae.org/resources/cheatsheets/>) available through the National Center on Disability and Access to Education (NCDAAE n.d.). Yet even when content creators follow tips and tools so they can develop accessible content, it still benefits from testing. Whitney

Quesenbery stated (2016), “It’s not that you can’t create an accessible pdf from an accessible Word file. It’s that it’s so easy to get it wrong”

As the field requests that content creators test what they contribute, testing, by default, is often performed by those who often do not have a strong technical background in web development, let alone web accessibility. These are individuals who do not know any markup language, nor could they interpret WCAG success criteria. This is being done at the same time the W3C (n.d.) recommends that only experts and users engage in web accessibility testing. Nontechnical staffs are put in a position to provide cursory evaluations of web content. They may sample or monitor accessibility, specifically test that which they create, or help evaluate accessibility for procurement purposes. While not specifically technical, some of these individuals are web accessibility advocates such as personnel in the Disability Services Office or the Special Education Office or Department.

Because of these issues, we have seen a rise in individuals who desire to know what they can do to engage in evaluation of web accessibility. The web accessibility field acknowledges that it is desirable for content creators to check the accessibility of their own web content. The web accessibility field may also agree that faculty and staff should be able to engage in some preliminary evaluation of linked content before making decisions to do so. Yet these practices create a unique tension. Under what conditions should nontechnical individuals evaluate web accessibility? What are the benefits of having them do so? What are the challenges or unintended consequences of such an approach?

What are these individuals doing when they test?

While we may not know exactly what it is that nontechnical individuals do when they perform testing, we can look to information they may use. There are several sources of information intended to assist nontechnical individuals as they engage in web accessibility testing. For example, WebAIM, in its partnership with the National Center on Disability and Access to Education, created a [simple rubric](http://ncdae.org/resources/cheatsheets/accessibility.php) (<http://ncdae.org/resources/cheatsheets/accessibility.php>) for evaluating some aspects of web accessibility. It walks the user through four main steps to take a peek at some accessibility issues. Their advice is to

1. Use the WAVE tool to identify “red” (i.e., programmatically determinable) errors,
2. Navigate without a mouse,
3. Zoom a page and enlarge text, and
4. Engage in other recommendations such as
 - a. Checking for captions of media,
 - b. Looking for instances of generic link text (i.e., “click here”),
 - c. Checking that the page title is unique and descriptive, and
 - d. Contacting individuals to see if they would be willing to review the page or site for possible issues.

WAI created [Easy Check First \(https://www.w3.org/WAI/eval/preliminary\)](https://www.w3.org/WAI/eval/preliminary) with a similar end in mind. This rubric is slightly more technical (it requires basic understanding of HTML), but it is also more complete. They walk users through nine simple “checks” for accessible web content. Under the WAI Easy Check First rubric, they ask users to check for

1. Page title
2. Image text alternatives
3. Headings
4. Resize text
5. Contrast ratio (i.e., color contrast)
6. Keyboard access and visual focus
7. Forms, labels, and errors
8. Multimedia alternatives, and a
9. Basic structure test

In both resources, the authors are careful to mention that passing these simple tests does not mean that a page is accessible. Certainly other groups have developed similar resources that are available to assist nontechnical personnel as they conduct a web accessibility evaluation.

In addition to assessing web pages, many resources are available to assist content creators who develop non-HTML based materials as they test the accessibility of what they have created (i.e., Word, PPT, PDF). For example, MS Word contains an accessibility checker and Microsoft published a [tutorial \(https://support.office.com/en-us/article/Use-the-Accessibility-Checker-on-your-Windows-desktop-to-find-accessibility-issues-a16f6de0-2f39-4a2b-8bd8-5ad801426c7f?ui=en-US&rs=en-US&ad=US\)](https://support.office.com/en-us/article/Use-the-Accessibility-Checker-on-your-Windows-desktop-to-find-accessibility-issues-a16f6de0-2f39-4a2b-8bd8-5ad801426c7f?ui=en-US&rs=en-US&ad=US) on its use; WebAIM also authored a brief [resource \(http://webaim.org/techniques/word/#checker\)](http://webaim.org/techniques/word/#checker) to capitalize on the Word checker. Yet the presence of testing resources such as these does not ensure that accessibility is achieved.

Issues for the field

As a field concerned with web accessibility outcomes it is important that we recognize what is happening with respect to testing and monitoring. This phenomenon in education may well creep into other sectors, if it has not already. Without a doubt, complete web accessibility testing must be accomplished by technical individuals who understand web accessibility principals and practice. However, given what is occurring now, we benefit from a discussion regarding this practice. Is it a good thing that more individuals are engaged in testing, no matter the level in which they can engage? This could have positive impacts not only for direct accessibility, but to help build awareness in others, as well as help those individuals develop needed skills. On the other side of this issue is the potential damage caused if limited testing conducted by nontechnical individuals is viewed to be as useful as true accessibility testing. Moreover, it could be disastrous if failure to detect errors from a limited set is considered to mean “accessible”. This is not unlike the problems faced in the field when individuals rely solely on automated testing, which is well known to be insufficient for accessibility determination if used alone. Perhaps “limitations” statements should be created and paired with the tests conducted by nontechnical

individuals? There may well be other unintended consequences of using these personnel during evaluation processes? What are the comfort levels of the web accessibility or education communities to this trend? It is probably a good time to involve the broader field in conversations such as these.

References

- Microsoft (n.d.) Use the Accessibility Checker on your Windows desktop to find accessibility issues. Retrieved from <https://support.office.com/en-us/article/Use-the-Accessibility-Checker-on-your-Windows-desktop-to-find-accessibility-issues-a16f6de0-2f39-4a2b-8bd8-5ad801426c7f?ui=en-US&rs=en-US&ad=US>
- National Center on Disability and Access to Education (n.d.), Cheatsheets. Retrieved from <http://ncdae.org/resources/cheatsheets/>
- National Center on Disability and Access to Education (n.d.), Identifying Web Accessibility Issues. Retrieved from <http://ncdae.org/resources/cheatsheets/accessibility.php>
- Quesenbery, W. (2016, September 6). Why can't we make it easier to be accessible? Retrieved from: <http://rosenfeldmedia.com/announcements/cant-make-easier-accessible/>
- Rowland, C. , Whiting, and Smith, J. (2015), What Do You Need to Create and Maintain Web Accessibility?, in Dave L. Edyburn (ed.) *Accessible Instructional Design (Advances in Special Education Technology, Volume 2)* Emerald Group Publishing Limited, pp.13 – 45
- WebAIM (n.d.) Microsoft Word Creating Accessible Documents: Accessibility Checker. Retrieved from <http://webaim.org/techniques/word/#checker>
- Web Accessibility Initiative (n.d.). Easy Checks – A first Review of Web Accessibility. Retrieved from <https://www.w3.org/WAI/eval/preliminary>
- W3C (n.d.). Accessibility Testing. Retrieved from https://www.w3.org/wiki/Accessibility_testing#Who_should_test.3F

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

OpenAjax Evaluation Library for Evaluating WCAG 2.0 Level A and AA Requirements

Jon Gunderson, PhD.

Disability Resources and Education Services
University of Illinois at Urbana/Champaign
jongund@illinois.edu

Abstract

The OpenAjax Evaluation library is designed to evaluate web pages for WCAG 2.0 Level A and AA requirements based on the accessibility techniques available in the HTML5 and ARIA 1.1 specifications. The purpose of the library is to help people understand the accessibility features they need to consider in the design of their web based resources and to provide quality assurance in the implementation and production resources. The rules also provide an outline for developing training programs in web accessibility.

Automated Tools

Automated tools are useful for two things: raising awareness about accessibility issues and in quality control making sure all the ‘i’'s have been dotted and the ‘t’'s crossed for accessibility. Many automated accessibility tools focus on the later use case, resulting in a set of pass/fail rules that require little understanding or knowledge of accessibility on the part of the development team. Development teams using these types of rulesets can be characterized as a “Whack-A-Mole” approach to accessibility, basically go through each identified violation and changing the code to get it to pass. Many times the changes in the code remove the accessibility violation, but the accessibility problem still persists. For example, adding an ALT attribute to an image, but not really describing the image in a meaningful way or as a decorative image. That is why so many of these types of rulesets include rules that look for common errors like authors using the “filename” in the ALT attribute content or decorative images have the ALT text of “thin blue line”.

Only a subset of accessibility requirements defined by the W3C Web Content Accessibility Guidelines 2.0 can be evaluated with rules that result in a pass or fail result. This limitation in a ruleset often leads development teams to a false understanding that their web resource is accessible or meets some level of conformance to WCAG 2.0 requirements when all the violations identified by the ruleset have been addressed. The “Whack-A-Mole” approach also leaves little time or incentive to understand the accessibility of a web page. Limited resources are focused on eliminating or minimizing the “known” accessibility violations, even though there are often more important accessibility issues that should be addressed that a ruleset does not identify, since it would require the development team to know something about the accessibility issue.

Rulesets that are more comprehensive of the WCAG 2.0 requirements help address the “awareness” issue and help development teams understand which WCAG 2.0 requirements apply to the specific markup they use to create a web resource. The evaluation results of comprehensive rulesets therefore are a filter on WCAG 2.0 requirements to reduce what development teams need to know about accessibility. For example, if no elements that can render video are found, development teams do not need to understand the WCAG requirements for captioning and audio description, or control over video. More comprehensive rulesets add the results of “not applicable” and “manual check” to the result vocabulary. Development teams can then focus their limited resources not only in fixing accessibility problems, but also understanding the accessibility of their web resources. The key to this is providing a reporting interface that help development teams easily understand which requirements apply to their web resources and where they can learn about the accessibility requirements. More comprehensive rulesets therefore need to provide a framework for training to move development teams from a remediation mode (a.k.a. Whack-a-mole) to accessible design (a.k.a understanding accessible user experience) for accessibility.

OpenAjax Evaluation Library and Rules

The OpenAjax Evaluation (OAE) library is designed to help web development teams understand the accessibility issues of the web resources based on the requirements of the W3C Web Content Accessibility Guidelines 2.0 Level A and AA requirements and the accessibility techniques associated with the use of the W3C Accessible Rich Internet Application (ARIA) and HTML5 specifications. One of the primary features of OAE is to filter WCAG requirements based on the content found in a browsers Document Object Model (DOM). It is important to analyze the DOM since modern websites at least some, if not all, content is rendered using Javascript (e.g. dynamically updating content on the page). OAE is therefore written entirely in Javascript and can be used in both browser and server based tools. Each ruleset in OAE has at least one rule for each WCAG 2.0 Level A and AA requirement. The OAE evaluation results help development teams identify which set of WCAG requirements and HTML elements they need to understand based on the markup found on their pages and templates. In a typical evaluation only 30-40% of the evaluation results are pass/fail results, so 60-70% require some human understanding and decision making (e.g manual checks) to determine if accessibility requirement apply and/or have been met. Usually only 40-50 of the 126 rules apply to a page so most rules are identified as “not applicable”, helping development teams focus on only the requirements that apply to their page.

The OAE library focuses on helping development teams learn about accessibility by including meta information about the purpose of the rule (e.g. why is it important for accessibility), techniques based on web standards that can be used to satisfy the rule and links to external resources to learn more about the accessibility requirements and solution techniques. OAE is based on the requirements of the ARIA Authoring Practices Guide 1.0 and accessibility features in HTML5, which is different from other automated tools that are typically based on WCAG 2.0 failure criteria. OAE is designed to support a authoring approach to accessibility, rather than remediation.

You can view the OAE rule details at: <https://fae.illinois.edu/rulesets/>

High-Level OpenAjax Concepts

Table 1 describes the high level concepts and vocabulary of the OpenAjax Evaluation (OAE) Library. The center of the OAE is the rule, a rule is designed to represent an important accessibility requirement for a set of target resources. For example, labeling form controls and provide information on what labeling techniques could be used. Rules are organized into rulesets. Rulesets represent a particular accessibility approach to meet WCAG 2.0 requirements. OAE currently has two rulesets. The primary ruleset is “HTML5 and ARIA Techniques” which requires the use of ARIA techniques in meeting WCAG 2.0 Requirements and is designed for development teams building new resources or using HTML5 technologies in their web resources. The “HTML4 Legacy Techniques” ruleset is for people that are still learning ARIA or are using the legacy accessibility techniques available in HTML4, for example working on a legacy code base that will not be moving to HTML5 technologies. Evaluation results in both element level results and rule level results. Rule level evaluation results help development teams know if they understanding or have fully implemented the requirements of a rule.

| Term | Definition |
|-----------------------|--|
| Rule | A low-level accessibility compliance requirement with numerous components, the most basic of which are a definition, an evaluation function, and result messages. When a rule is evaluated, it is assigned a rule result value. |
| Ruleset | A set of rules that embodies specific standards compliance goals. Each rule within a ruleset is designated as either required or recommended, according to its relationship to the compliance goals of the ruleset. |
| Evaluation | The application of all evaluation functions of all rules in the selected ruleset to their corresponding target resources in a web page, and the subsequent collection of all evaluation results and messages, at both the rule and element levels, into a dataset. |
| Rule result | The aggregate result of a rule evaluation with respect to its target resources (all of its target elements and/or the entire page). |
| Element result | The result of a rule evaluation with respect to one of its target elements. |

Table 1 – High Level Concepts of OpenAjax Evaluation Library

Rule and Element Result Values

Table 2 describes the values of the evaluation of a rule. When a rule is applied to the target resources of a web resource there will be individual element results and an aggregate rule result. The rule result is assigned the lowest element result. For example, if there are five IMG elements on a page and 4 of the IMG elements have an ALT attribute and the 5th IMG element has no ALT attribute there will be five element results (4 Pass and 1 Violation) and since there is one violation the aggregate rule result for the page is a “Violation”. If the page has no IMG elements the aggregate rule result is “Not Applicable”.

| Value | Description |
|-----------------------|---|
| Violation | One or more target elements fail the requirements of a required rule |
| Warning | One or more target elements fail the requirements of a recommended rule |
| Manual Check | Unable to programmatically determine a result - evaluation requires human inspection |
| Pass | All rule requirements were satisfied |
| Hidden | Target elements were not evaluated because they are hidden (either visually or from assistive technologies) |
| Not Applicable | Rule was not applied because there were no relevant target elements (Rule result only) |

Table 2 – OpenAjax Evaluation Rule and Element Result Values

Rule Features

A rule is the central object of the OpenAjax evaluation library and contains a rich set of information to help development teams understand the accessibility feature identified by the rule. The goal of the rule is to help development teams understand the accessibility requirement and how their implementation of the rule requirements affects the experience of people with disabilities. Table 3

| Term | Definition |
|-------------------------|--|
| Summary | A concise or abbreviated statement of the rule definition |
| Definition | The markup patterns or requirements that must be present in, or met by, the target resources |
| Success Criteria | <ul style="list-style-type: none"> • Primary WCAG 2.0 Success Criteria associated with the rule (required) • Secondary WCAG 2.0 Success Criteria associated with the rule (optional) |
| Required | <ul style="list-style-type: none"> • Identifies the rule as required or recommend accessibility feature for meeting WCAG 2.0 requirements • This is defined at the ruleset level |

| Term | Definition |
|--------------------------------|--|
| Rule Scope | <ul style="list-style-type: none"> • The scope helps the developer understand what they need to consider in satisfying a rule. • Element level rules require looking at element's features for content, labeling, properties and states. • Page level rules require looking at the page to help users understand the purpose and structure of the content on a page. • Website level rules require looking at the set of pages in a website for consistent ordering and labeling of content and making sure users understand the unique content of each page through effective titling of the pages. |
| Target Resources | <ul style="list-style-type: none"> • The set of HTML elements associated with the rule • For rules with scope of page, the page is also a target resource |
| Purpose | How does satisfying the rule benefit people with disabilities (e.g. why is this important) |
| Techniques | HTML4, HTML5, CSS and Javascript techniques that can be used to satisfy the rule |
| Informational Resources | Pointers to other resources including W3C Specifications, examples or other resources that provide information on understanding and satisfying the rule requirements. |

Table 3 – OpenAjax Evaluation Rule Properties

Rule Categories

OpenAjax Evaluation Library provides an alternative to the WCAG 2.0 Guideline/Success Criterion for organizing rules and rule results. Table 4 describes each of the current 12 Rule Categories that align closely with the organization of the HTML5 specification. Rule categories allow development teams to have rules grouped more by their relationship to markup to make it easier to identify closely related accessibility concepts. Many users of the tools using the OAE library have found the rule categories easier to understand than grouping by WCAG Guidelines. WCAG Guideline organization are also supported for people that are familiar or more comfortable with the WCAG 2.0. In all views of individual rules and evaluation results provide information on the Success Criterion they are associated with.

| Category | Definition |
|------------------------|--|
| Landmarks | Use ARIA landmark roles to structure the content of each page and identify major sections of content, thus making them more findable and navigable. The use of landmarks will, in many cases, reflect the visual styling and page layouts that web designers utilize to set apart various sections of content. |
| Headings | Use heading elements (H1-H6) to provide appropriate labels for landmarks, and to identify subsections of content within landmarks. |
| Styles/Content | Use proper HTML markup to identify the semantics and language of text content. Ensure that text is readable by adhering to color contrast guidelines, and that information is not conveyed solely by the use of color, shape, location or sound. |
| Images | Provide appropriate text alternatives for static images and graphics. |
| Links | Use link text that properly describes the target of each link. Ensure consistency and uniqueness for links that are usable, predictable and understandable. |
| Tables | Provide table captions or other meta-information as needed. Provide row and column header references for data cells of data tables. Ensure that tables used for layout properly linearize text content. |
| Forms | Provide meaningful labels for form elements and usable and understandable error feedback as needed. |
| Widgets/Scripts | Use appropriate event handlers on elements to support native interactivity using JavaScript. Ensure that custom widgets created using JavaScript support keyboard interaction and include ARIA markup to describe their roles, properties and states. |
| Audio/Video | Provide appropriate text transcripts, captions or audio descriptions for elements used in rendering audio and video content. |
| Keyboard | Provide logical and sequential keyboard navigation among interactive elements such as links and form controls. Use standard models of keyboard interaction for custom widgets. |
| Timing | Eliminate accessibility problems caused by time limits on input and by content that moves, scrolls, flashes or auto-updates. |
| Site Navigation | Ensure the consistent labeling and ordering of recurrent page sections across all pages within a website. Provide a meaningful title for each page within a website. |

Table 4 – OpenAjax Rule Categories

Next Steps for Evaluation Library

One of the uses of the OAE library rules is to support training since the rules define functional requirements for ARIA, HTML5, CSS and Javascript for implementing WCAG 2.0 requirements. The University of Illinois is in the process of creating a badging program based on the rule categories and the rules in those categories. The OAE library and the training will complement each by sharing a common vocabulary for reinforcing training on accessibility concepts. The rules are a database of our current understanding of web accessibility and provides a framework for discussing what we mean by a resource being “accessible”.

The version 1.0 of the OpenAjax Evaluation Library was published in August of 2016. Work will continue to add improve the rules based on changes in the ARIA and HTML specification and implementations in browsers. The OpenAjax Accessibility Task Force meets monthly to discuss improvements to the rules and rulesets. The group is collaborating with the W3C AutoWCAG community group to share the OpenAjax rules with that group and harmonize the rules defined by the community group with the rules provided by the OAE library. The task force is open for public participation and all of the work of the group is open source and freely available to use under the Apache 2.0 license.

More information on the OpenAjax Evaluation Library and Accessibility Task Force can be found at:

<http://www.openajax.org/member/wiki/Accessibility>

Tools using the OpenAjax Evaluation Library

The following open source tools use the OpenAjax Evaluation library to evaluate web resources for accessibility.

AInspector Sidebar for Firefox

AInspector Sidebar for the Firefox browser supports the evaluation of web pages for the accessibility features required by the W3C Web Content Accessibility Guidelines 2.0 Level A and AA Success Criteria. The tool provides both summary information and detailed information on the accessibility requirements that apply to a web page (e.g. WCAG 2.0 requirement filtering). Users can highlight and inspect element information on the page associated for each requirement. Each result has information on the requirement, techniques that can be used to implement the requirement and links to more information. AInspector Sidebar uses the open source OpenAjax Evaluation Library, Rules and Rulesets.

Get Latest Build and Documentation

<http://ainspector.github.io>

Available from Mozilla Add-on Website

<https://addons.mozilla.org/en-us/firefox/addon/ainspector-sidebar/>

Functional Accessibility Evaluator

The Functional Accessibility Evaluator (FAE) 2.0 provides accessibility information on an entire website for WCAG 2.0 Level A and AA Success Criteria. FAE 2.0 provides both website summary and page level detail on the accessibility requirements that applied to the pages with in a website. FAE 2.0 uses the same open source OpenAjax Evaluation Library, Rules and Rulesets as AInspector Sidebar. FAE 2.0 and AInspector Sidebar are designed to complement themselves. FAE 2.0 provide a overview of accessibility of a website that can be used in design, planning and quality assurance for project management and AInspector Sidebar provides detailed information and inspection capability of a particular page.

Functional Accessibility Evaluator 2.0 (Institutional Version)

<http://fae.illinois.edu/>

Functional Accessibility Evaluator 2.0 (Individual Version)

<https://fae.disability.illinois.edu/>

References

Berjon, R., Faulkner, S., Leithead, T., Doyle Navara, E., O'Connor, E., Pfeiffer, S. (editors) (2014) HTML5, <https://www.w3.org/TR/html5/>.

Caldwell, B., Cooper, M., Guarino Reid, L. and Vanderheiden, G. (editors) (2008) Web Content Accessibility Guidelines (WCAG) 2.0., <https://www.w3.org/TR/WCAG/>.

Craig, J., Cooper, M. (editors) (2104) Accessible Rich Internet Applications (WAI-ARIA) 1.0, <https://www.w3.org/TR/wai-aria/>.

King, M., Nurthen, J., Cooper, M., Bijl, M. (2016) WAI-ARIA Authoring Practices 1.1, <https://www.w3.org/TR/wai-aria-practices-1.1/>.

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

The Unbearable Inaccessibility of Slideshows

Gian Wild

AccessibilityOz
New York, New York, USA
gian@accessibilityoz.com

Matt McLeod

AccessibilityOz
Melbourne, Victoria, Australia
matt@accessibilityoz.com

Abstract

Slideshows (or carousels or sliders) are everywhere on the web, and mostly they aren't accessible. Often there is no way to pause the slideshow, which is a major accessibility violation of WCAG2. In most cases slideshows can only be manipulated by the mouse. Making them accessible requires following five principles: allow the user to stop all movement; provide visible controls accessible to the keyboard, mouse and touch; provide a valid and understandable focus order through the slideshow; use valid coding and style sheets; and provide meaningful alternatives.

Slideshow. Carousel. Slider

Whatever you call them, they're ubiquitous on organizational home pages across the web. And almost all of them are inaccessible.

There are two main types of slideshows that we often see. The first one (Figure 1) has circles to indicate the slides and provides a method of moving between slides. The second one (Figure 2) has arrows at the left and right edge of the slideshow to allow users to choose the next or previous slide.



Figure 1 - Circles to indicate slides, and as a method to move between slides



Figure 2 - Left and right arrows on the sides of the slideshow to allow users to move between slides

The biggest problem with these slideshows is that they cannot be paused by the user. This is a violation of one of the non-interference clauses of WCAG2 Success Criterion 2.2.2, Pause, Stop, Hide. The next most serious problem is that neither is keyboard accessible, a violation of WCAG2 Success Criterion 2.2.1, Keyboard Accessible. In the first example the keyboard skips the slideshow entirely. In the second example the arrows are not keyboard accessible, but the link within the two slides in the slideshow is. This means that the keyboard focus disappears when it is focused on a slide not currently visible.

We are going to talk about the five principles in making an accessible slideshow.

Principle 1: Allow the user to stop all movement

One of the reasons marketing departments like slideshows is because there's movement on the page, which draws the viewer's attention to that area.

Allowing users to control the movement of a slideshow is one of the four "non-interference" clauses in WCAG2. These four success criteria apply to all content on the page – even content that is classified as inaccessible by the owner.

The easiest solution is to provide a link that stops the movement. We'll talk more about the best way to implement this in the Controls section under Principle 2.

Principle 2: Provide visible controls accessible to the keyboard, mouse and touch

Far too often the controls on a slideshow are very small, or quite low-contrast compared to the content they overlay. And focus / hover is often not too obvious either. There are four requirements to think about here.

A. Are there controls to stop, start and move between slides?

As mentioned in Principle 1, it is absolutely essential that users be able to stop all movement in the slideshow, via the mouse, the keyboard or touch alone. Also necessary is the ability for users to restart a slideshow if they have stopped it. It is not an actual requirement that users be able to move between slides, but it's a good idea.

B. Are the controls highly visible (large enough to be seen and meet colour contrast requirements)?

We see problems with this requirement on **a lot** of slideshows. Either the controls are teeny-tiny or the controls are overlaid on the content and colour contrast is entirely dependent on the underlying slide. There are a number of solutions to this, including having a solid background behind the control which provides the contrast, and using a solid border around the control which provides the contrast.

C. Is the focus / hover indication sufficiently different to the default state while also having enough contrast with the content?

We mostly see focus indicators only appear on mouse hover (and not keyboard focus) and often it consists of only a slight colour change of the controls. It is important that if you are going to use a colour change you need to meet colour contrast requirements.

An even better solution is to change the shape of the control on focus, like we have on the [AccessibilityOz](http://www.accessibilityoz.com) (<http://www.accessibilityoz.com>) slideshow (see Figure 3 and 4). But remember it needs to appear on keyboard focus too!

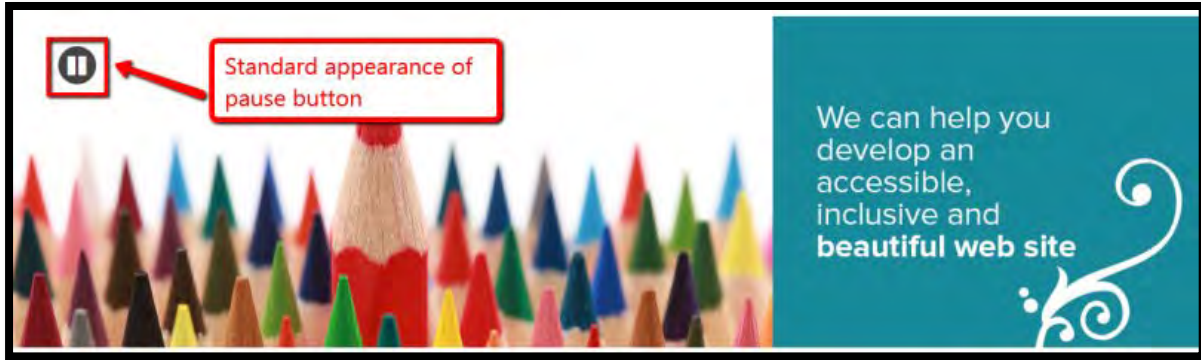


Figure 3 - Standard appearance of pause button is a white pause icon on a circular black background



Figure 4 - The pause button appears as a black pause icon on a square white background on both mouse hover and keyboard focus

D. Can the controls be manipulated by the mouse, the keyboard and touch?

Most slideshow controls are completely inaccessible to the keyboard. All the controls that make up the slideshow need to be made accessible by keyboard alone. If the controls are done using buttons or links, then they should be implicitly keyboard accessible unless you muck around with TABINDEX.

When it comes to mobile, firstly, there must be an **obvious** method to pause the slideshow (it's not sufficient to pause the slideshow on touch). Providing an actual pause button is necessary, like in [Rooted in Rights](http://www.rootedinrights.org/) (<http://www.rootedinrights.org/>) in Figure 5. It is important that this button is large enough for people to tap easily (at least 7 – 10 mm). If the user can swipe to see the next slide, this needs to be indicated visually on the slideshow. And all actions; swipe, scroll and touch, must be triggered on the **removal** of touch, not on the initiation of touch (ie. when the user removes their finger from the device, not when the user first touches the device).

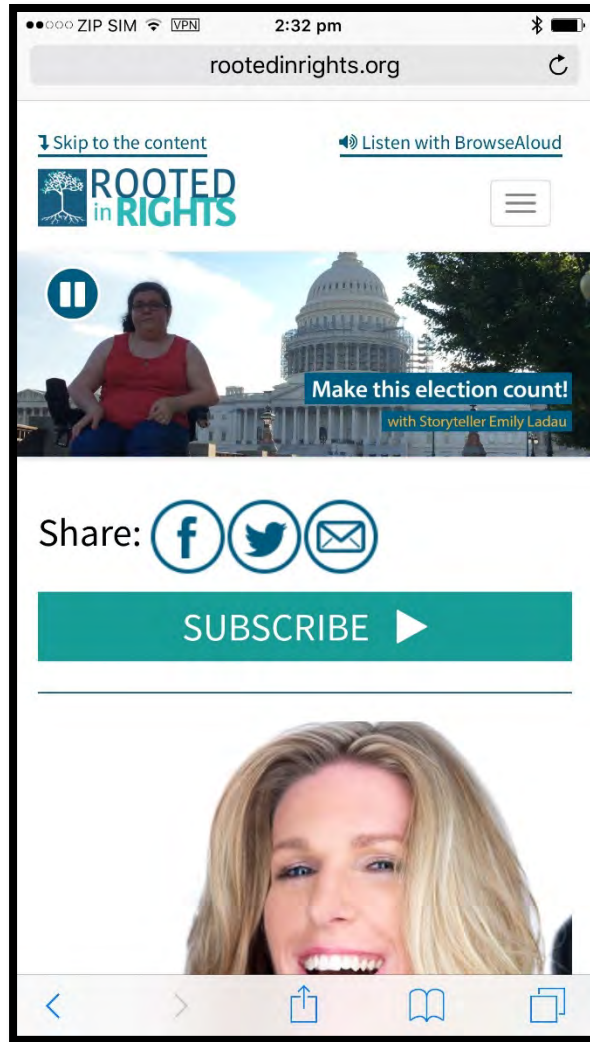


Figure 5 – Rooted in Rights slideshow with a large pause button

[City of Melbourne](http://www.melbourne.vic.gov.au) (<http://www.melbourne.vic.gov.au>) is a great example of proper controls on the mobile. The slideshow moves by default and there is a clear pause button as well as controls to move between slides (Figure 6).

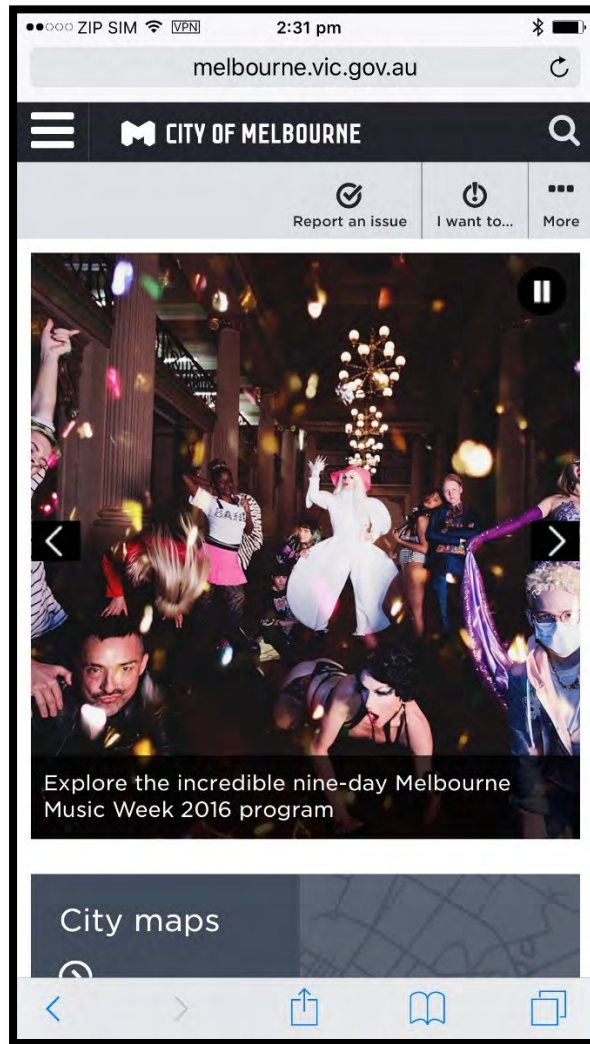


Figure 6 – City of Melbourne slideshow with a large pause button and arrows

Principle 3: Provide a valid and understandable focus order through the slideshow

Focus order is an important accessibility requirement. The best way to explain it is that the order that items receive focus when tabbing through items, should match the underlying HTML (ie the site with style sheets disabled), which should match the visual order of the page.

However, there is an additional requirement, which is that when you make a change to the page, it only changes content **after** the current focus. So a slideshow with an arrow on the right fails this requirement as change occurs **prior** to the current focus. There is an exception to this, and that is if the user is warned about the content change.

So there are three requirements here:

- the focus order must match the source and visual layout;
- the controls must come before the content, or at least the “pause” control appears before anything else; and
- ensure activating a control doesn’t change content before the current focus.

This is a place where some compromise is inevitable, but in general try to put things like a pause/play button and position indicators at the top of the slideshow and have them come before the content in the tab-focus order.

Principle 4: Valid coding and stylesheets

This brings us to proper coding and stylesheet usage of your slideshow. The slideshow does not need to **function** with style sheets disabled, but all the content needs to be **available**.

Most of our clients just show one slide after the other. It’s not pretty, but it’s accessible.

One of the things you must be very careful to avoid is slideshow movement when style sheets are disabled. Often what we see are all the slides displayed, one after the other, and then the first slide disappears, and all the slides move up, then the next slide disappears, and all the other slides move up, etc.

This can be apparent to screen reader users; they read a slide then when they go back to try and find it, it is gone. The other thing we often see is overlapping content when style sheets are disabled.

If there is any text on the slideshow it needs to increase when the user increases text size in the browser. On a mobile device, your slideshow needs to support pinch zoom.

Principle 5: Provide meaningful alternatives

The sad truth is that no matter how much effort you put in to this, there are going to be some edge cases – combinations of older browsers and assistive technology – where your spiffy slideshow simply won't be accessible.

So you'll need to provide an alternative to the slideshow, assuming for a moment that there is any meaningful content in it. And, quite frankly, if there is no meaningful content in your slideshow then **you should get rid of your slideshow**.

The simplest way to provide an alternative is to have the text from each slide repeated in a "screen-reader only" section along with any links. Then include a "skip past the slideshow" link. This is done by creating a "screen-reader only" class.

It's important (and an accessibility requirement) that this alternative displays at the site of the slideshow and not elsewhere on the page. Also remember that the controls reliant on JavaScript should not be visible in the case where JavaScript is disabled.

It is necessary that the alternative is **equivalent** to the slideshow. Therefore, if you have links in your slideshow to areas in your site, you need to replicate these links in the alternative.

In some cases, the images in the slideshow will need to have valid ALT attributes. If the images are purely decorative (as in the AccessibilityOz example) then you can use alt="". However, if the images do convey information then they need a valid ALT attribute.

However, in the Rooted in Rights site (see Figure 7) the text is part of the image and therefore needs an alternative. In this example the ALT attribute is "Emily Ladau speaks to the camera in front of the U.S. Capitol. Text: Make this Election Count with Storyteller Emily Ladau".

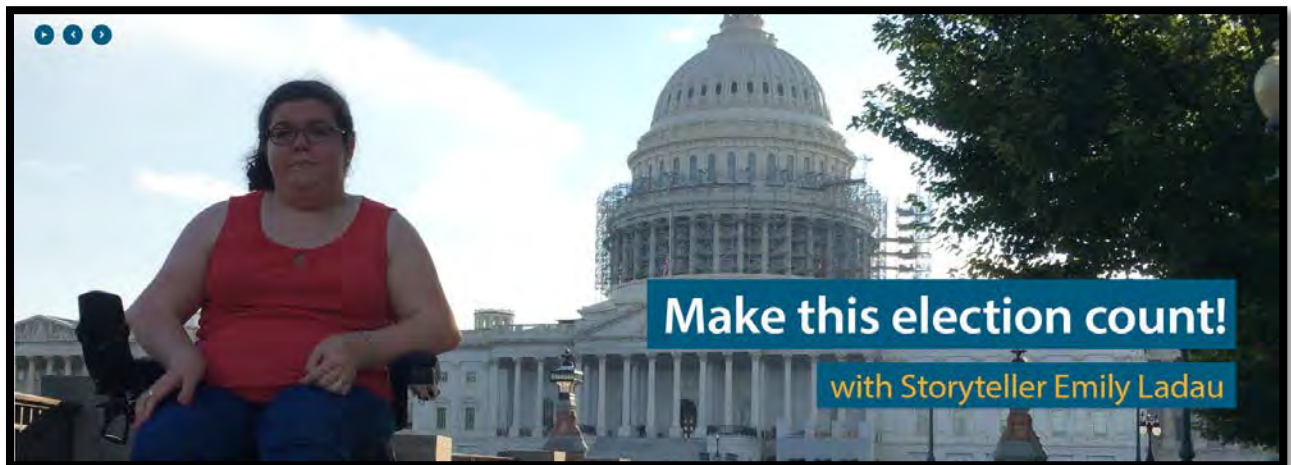


Figure 7 – Slide requiring a valid ALT attribute

Conclusion

So now you understand why it is so difficult to make an accessible slideshow! And before you build one, have a look at www.ShouldIUseACarousel.com.

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Testing for Accessibility Regulations

Jonathan Avila

Chief Accessibility Officer, SSB BART Group
1593 Spring Hill Rd. Ste. 720, Vienna, VA 22182
jon.avila@ssbbartgroup.com

Abstract

This paper will discuss broad testing strategies based on accessibility regulations as they apply to various digital products and services. If products/services are sold into the US Federal government, it may be clear that testing must follow the Section 508 standards. But which accessibility standards must be followed if you don't sell directly into the government but instead you accept Medicare funding or you offer health products on the Federal Healthcare market? What if your products or services include email, a browser, text messaging, video conferencing, or VOIP calls? Or what if you make point of sale devices, ATMs, or fare machines? What about products sold into private or state supported educational institutions? What standards do you use to test for these regulations? This paper will outline the testing approach standards used and how functional objectives can be used to measure outcome based goals for access by persons with disabilities.

Determine Applicability of Regulations

In order to determine what accessibility standards must be met, you must first determine the applicability of regulations to your products or services. This may be based on the product or service type, the manufacture date of the product, the jurisdiction, where the funding source comes from, or other factors such as organizational or other procurement policies.

Our recommendation is to complete a functionality and regulatory coverage questionnaire and perform a system survey and analysis to determine the scope of what must be covered. From that questionnaire and analysis, a process can be put in place to include the correct regulatory requirements into the business requirements, design checklist, implementation guidance, testing documents, and ultimately address the support and documentation provided to the end user.

- Regulatory/functionality questions are arranged around industry. A sample is provided below:
- Education: Does the institution receive Federal funding? Does the institution receive State or local funding?
- Healthcare: Does the program receive Federal funding? Is the program part of the Federal Healthcare Marketplace? Does the program provide service through Medicare or Medicaid? Does the program provide patient access to electronic health records?

- Telecommunications/Hardware: Does the product contain a browser? Does the product playback or receive video programming? Does the product or service provide two-way voice communications?

Places of Public Accommodation and State and Local Governments

Litigation, structured negotiations, and threats of lawsuits under the Americans with Disabilities Act (ADA) have been steadily increasing. These lawsuits are filed by the Department of Justice, advocacy groups, or ADA-focused attorneys representing clients with disabilities. The Department of Justice has said consistently over the past 10 years that websites are covered under the ADA titles II and III. While the ADA does not specifically reference websites or a specific technical standard, litigation is still occurring and compliance with WCAG 2.0 A/AA standards can minimize risk. In most of the agreements reached over ADA website accessibility, WCAG 2 Level A and AA is chosen for web, mobile, and document content. When other applicable accessibility guidelines exist, these may also be included. For example, when a product or service allows for authoring of content, the Authoring Tool Accessibility Guidelines may be part of an agreement reached by both parties.

For ATMs and fare machines, the ADA guidelines last updated in 2010 are clear about the specific technical features that must be implemented to make sure they are accessible to persons with disabilities. The technical standards provide details about floor space, operable parts, privacy, input, speech output, and additional technical requirements. While other interactive transaction machines (ITMS) aren't specifically covered by the technical standard, they are covered broadly under the ADA and the DoJ provides advice on how these ATM standards can be leveraged for ITMS.

Section 508 Standards

The Section 508 standards apply to EIT that is developed, procured, maintained, or used by the Federal government. It is typically enforced through the procurement process since Section 508 standards are a part of the Federal Acquisition Regulations (FAR). While Section 508 doesn't require organizations outside of the government to test or make their digital product or services accessible, it does set Federal procurement standards. Thus, if you want to sell into the Federal government, you must follow their procurement requirements for accessible EIT.

Agencies must procure the most accessible product that meets the business needs of the organization. If you are selling into the government, you will likely need to create a Voluntary Product Accessibility Template (VPAT) or complete a Government Product Accessibility Template (GPAT) based on the test results of the product as part of the solicitation and procurement process. The VPAT or GPAT should address the applicable areas that are covered by the product or service. For example, for a mobile phone, applicable standards likely include desktop and portable computers, software, self-contained, closed products and services, as well as telecommunication features, functional performance criteria for anything not addressed by the technical standards, and document and support standards. To help organizations determine which standards from Section 508 are applicable, the Buy Accessible Wizard can generate a GPAT

document based on the type of product or service selected in the wizard and the listed parts can then be tested and completed.

Section 508 is in the final stages of being refreshed and will adopt by reference the Web Content Accessibility Guidelines (WCAG) 2.0 Level A and AA for web content, software, and documentation. Web content that is also an authoring tool will have some additional standards, and software that acts as a platform or operating system has some additional standards as well that must be addressed. The hardware standards will also be updated as part of the refresh. Thus, testing should be updated to cover WCAG 2 Level A and AA for most content types.

Finally, many agencies have their own Section 508 checklist of requirements that may or may not be identified during the procurement process. If the product is being sold into one of these agencies (the Department of Veterans Affairs, Health and Human Services, Department of Homeland Security, Social Security Administration, and others) you will want to consult the agency's Section 508 website or contact the agency to make sure you have all of the agency specific Section 508 checkpoints that must be met for successful delivery. Many agencies have Section 508 program offices that often will evaluate a percentage of deliverables against their checklist and the Section 508 standards.

Section 508-like Standards found at the US State and Local Level

Many US State governments have adopted website and procurement standards similar to Section 508. For example, Texas, Minnesota, and Virginia are among the states that have Section 508-like procurement and web accessibility policies. The states adopt particular technical standards by requiring the same technical standards found in Section 508 for websites, software, and other technology or by adopting WCAG 2.0 standards such as WCAG 2 Level A and AA as their state standard. Depending on the state, vendors must make sure their products are accessible to the standards. These procurement standards generally apply across the state and include procurements made in state funded education including higher education as well as state health and human services programs. Some states may require VPATs or other procurement documents describing how a product meets the standards.

Education

Even private education institutions must comply with the ADA Title III as they are places of public accommodation. State and local institutions are covered under Title II of the ADA as they are funded by state and local governments.

Recent litigation, settlements, and structured negotiations in the education space have adopted a number of accessibility standards. The Department of Education, along with the Department of Justice, has been involved in these enforcement efforts. WCAG 2 Level A and AA have been proposed in many of these agreements as methods to meet accessibility of web and mobile web content and documents. In the 2016, the Supplemental Advanced Notice of Proposed Rulemaking the US Department of Justice proposes WCAG 2 Level A and AA be required for state and local government's web content.

Air Travel

The Air Carrier Access Act (ACAA) requires that core features of air carrier sites that service the US are accessible. As of December 2016, these sites will have to be fully accessible to persons with disabilities. The ACAA standards which were implemented by the Department of Transportation require WCAG 2 Level A and AA standards for web and mobile web content including embedded content and documents. They do not apply to native mobile apps.

In addition, the ACAA requires the kiosks owned solely or jointly by the airport or air carriers are accessible. The initial kiosk requirements go into effect in December 2016, requiring 25% of new kiosks at each kiosk location be accessible and requiring 25% of all kiosks to be accessible by 2022. The kiosk standards are based on a combination of the Section 508 standards for self-contained, closed products and the ADA 2010 guidelines standards for kiosks and point of sale devices.

Telecommunication, Video Playback/Recording Devices/Computer Hardware

The CVAA provided broad updates to the Communications Act with regulations promulgated by the Federal Communications Commission (FCC). The CVAA updates help to ensure modern forms of communication are accessible to persons with disabilities. The updates require captions for TV programming distributed online, support for caption and audio description rendering or pass through on video apparatus, control of the display of captions, access to emergency information, text to 911, access to browsers, email, text messaging, VOIP, two-way voice communication, and the user interfaces of video apparatus designed to receive or playback video programming.

The two main areas that the CVAA covers are video programming and advanced communication services (ACS). The video programming requirements are targeted at manufacturers of hardware that can receive or playback video programming while the ACS features are aimed at products as well as service providers where no physical device is manufactured.

The video playback requirements are technical requirements around what feature must be present. However, there is latitude in how it can be implemented. For example, the user must be able to change the channel, turn on captions with a key, icon, or button, and power on/off the device, among other functions. The ACS requirements on the other hand are functional requirements similar to those in Section 255. The ACS requirements focus on the availability of information and the operation of controls by users with disabilities.

ACS requirements can be met functionally by coupling the technology with third-party software such as assistive technology or accessibility features of the platform as long as they are of nominal cost and supported for the life of the product. For web-based ACS, the FCC indicates that following the WCAG 2 Level A and AA success criteria would likely meet the bar of the functional performance objectives but it's no guarantee that the FCC could not specify technical standards for the ACS based on the requirements set forth by congress in the CVAA.

Health Care

Section 504 of the Rehabilitation Act of 1973 (“Section 504”), and the Americans with Disabilities Act (ADA) are civil rights laws to prohibit discrimination against persons with disabilities and apply to programs delivered through the marketplace, federally conducted marketplace structure, and organizations that assist in helping consumers make marketplace choices. The ADA applies to organizations that provide accommodations to the public while Section 504 applies to programs that are federally funded.

Section 508 of the Rehabilitation Act (“Section 508”) applies to federal agencies and cover EIT developed, procured, maintained, or used. Section 508 is not a civil rights law but a procurement enforced policy to have EIT that is accessible out of the box to employees and members of the public with disabilities in the same manner as people without disabilities. Under Section 508, the Federal Healthcare Insurance Marketplace must be compliance to accessibility technical standards and functionally usable by people with disabilities.

Electronic Health Records (EHR) require conformance with WCAG 2.0 Level A. For this requirement, the technical standards of WCAG 2.0 should be applied and tested to end user facing health records.

Section 1557 of the Affordable Care Act (ACA) prohibits discrimination based on disability in programs that are funded by or established by the ACA. The ACA addresses websites as well as other information and communication technologies. The ACA, however, does not specify technical standards that must be used to ensure access.

Health and Human Services and Centers for Medicaid and Medicare (CMS) have some additional standards that apply when receiving federal funding or if a program is federally administered. These standards often include functional evaluations by users with disabilities as well as requiring following the technical standards of Section 508 and the HHS Section 508 checklist among other things.

Tables of Potentially Applicable Standards & Testing Type

| Potentially Applicable Regulations | Type | Standard |
|---------------------------------------|--------------------------|---|
| Healthcare | | |
| Affordable Care Act (ACA) | Civil rights | <ul style="list-style-type: none"> Prohibits discrimination against persons with disabilities but not set specific technical standards |
| Section 508 | Technical and functional | <ul style="list-style-type: none"> Section 508 technical, function, and support documentation standards HHS Section 508 Checklist |
| Americans with Disabilities Act (ADA) | Civil Rights | <ul style="list-style-type: none"> Providing auxiliary aids and services to meet the communication needs of persons with disabilities |

| Potentially Applicable Regulations | Type | Standard |
|---|--|--|
| | | <ul style="list-style-type: none"> No specific technical standards for websites or mobile <ul style="list-style-type: none"> De facto WCAG 2 Level A& AA |
| Electronic Health Records (EHR) | Technical | <ul style="list-style-type: none"> WCAG 2 Level A |
| Additional HHS rules and regulations | Mix of civil rights, function, and technical | <ul style="list-style-type: none"> Requires that consumer education materials, websites, or other tools utilized for consumer assistance purposes be accessible to persons with disabilities HHS Section 508 checklist Functional review by users with disabilities Providing auxiliary aids and services |
| Places of Public Accommodation | | |
| Americans with Disabilities Act | Civil rights, some technical standards for ICT devices such as ATMs, fare machines | <ul style="list-style-type: none"> Providing auxiliary aids and services to meet the communication needs of people with disabilities No specific technical standards for websites or mobile <ul style="list-style-type: none"> De facto is WCAG 2 Level A & AA ADA 2010 requirements for ATMs and fare machines |
| Private Educational Institutions | | |
| ADA | Civil rights | <ul style="list-style-type: none"> Prohibits discrimination based on disability |
| State Supported Educational Institutions | | |
| Section 504 | Civil rights | <ul style="list-style-type: none"> Prohibits discrimination against persons with disabilities |
| ADA | Civil rights | <ul style="list-style-type: none"> De facto is becoming WCAG 2.0 based on structured negotiations, settlements, litigation, etc. |
| State Section 508-like standards | Technical | <ul style="list-style-type: none"> Section 508-like or WCAG 2.0 based procurement standards |
| Air Travel | | |
| Air Carrier Access Act (ACAA) | Technical | <ul style="list-style-type: none"> Websites/mobile web and embedded content WCAG 2.0 Level A and AA Kiosk standards based on Section 508 and ADA 2010 guidelines for kiosks and point of sale devices |

| Potentially Applicable Regulations | Type | Standard |
|--|--------------------------|---|
| Federal Government and Products sold into Federal Government | | |
| Section 508 | Technical and functional | <ul style="list-style-type: none"> Section 508 technical, function, and support documentation standards HHS Section 508 Checklist |
| State and local Government and Products sold into State and Local Government | | |
| Section 508-like standards at the state level | Typically Technical | Section 508-like technical standard or WCAG 2.0 |
| Products that include Telecommunications (SMS, fax, phone, modem, etc.) | | |
| Section 255 | Functional | Functional requirements for the availability of information and operation of controls |
| CVAA | Functional | Functional requirements for the availability of information and operation of controls |
| Video Programming Ownership or Distribution | | |
| CVAA | Functional | Functional requirements for the availability of information and operation of controls |
| Product or Service Provider that includes advanced communication services (email, VOIP, instant message, browser, video conference, etc.) | | |
| CVAA | Functional | Functional requirements for the availability of information and operation of controls |
| Digital apparatus design to playback or record video programming | | |
| CVAA | Technical | Describe which functions must be accessible to people who are blind or visually impaired and key, button, or icon to enable captions for people who are Deaf or hard of hearing |

Testing by Users with Disabilities

No matter what regulations are in place with functional or technical standards, testing with users that have disabilities is an essential part of the process. Sometimes individual components are accessible, but when they are put together into a process accessibility issues can arise. Also, testing with real people that use assistive technology and accessibility features can help organizations uncover accessibility support issues. For example, this type of testing may uncover that the techniques used along with the user agent (e.g., browser) and assistive technology combination have barriers to access that were not previously documented.

Testing should ideally be performed for core tasks of the system with assistive technology that is appropriate. For example, for CVAA it must be native or nominal cost.

Conclusion

Organizations should complete a regulatory and functionality questionnaire to determine what regulations impact their products and services. When the scope of accessibility requirements is known, organizations can build these requirements into all of their processes including the

testing and validation process. Testing to the standards that are applicable (both technical and functional) as well as testing with users that have disabilities will help organizations break down barriers and reduce risk to the organization while creating benefits for all of society.

References

21ST Century Communications and Video Accessibility Act. (n.d.) Retrieved September 30, 2016, from <https://transition.fcc.gov/cib/dro/cvaa.html>

ACAA in the Federal Register. (n.d.) Retrieved September 30, 2016, from https://www.federalregister.gov/documents/2016/05/18/2016-11458/nondiscrimination-in-health-programs-and-activities?utm_campaign=subscription+mailing+list&utm_medium=email&utm_source=federalregister.gov

ADA Title II SANPRM. (n.d.) Retrieved September 20, 2016, from <https://www.ada.gov/regs2016/sanprm.html>

Electronic and Information Technology Accessibility Standards; Final Rule, 65 Fed. Reg. 246 (December 21, 2000) (Codified at 36 CFR pt 1194)

HHS Checklist. (n.d.) Retrieved September 30, 2016, from <http://www.hhs.gov/web/section-508/making-files-accessible/checklist/>

Public Law 111-260 Signed by President Barack Obama On October 8, 2010. (n.d.) Retrieved September 30, 2016, from http://hraunfoss.fcc.gov/edocs_public/attachmatch/DOC-303939A1.doc

W3C World Wide Web Consortium (2008), Web Content Accessibility Guidelines 2.0 Level A & Level AA Success Criteria. (n.d.) Retrieved from <http://www.w3.org/TR/200X/REC-WCAG20-20081211/>

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

QA Enabled Testing as the Hook for Organizational Change

Chris M. Law

Accessibility Track Consulting, LLC
McLean, VA, USA
chrismlaw@accessibilitytrack.com

Abstract

In this paper we argue for introducing Quality Assurance (QA) in accessibility testing as a first step, and that this can serve as a useful hook for wider organizational change with respect to accessibility. QA-Enabled testing methods are defined in terms of code inspection testing tools that are usable by programmers and existing QA test teams. We make comparisons with a traditional form of accessibility testing, namely the use of assistive technology as the primary tool, arguing that this is a form of ineffective Quality Control (QC) because it does not permit QA throughout development. Links to existing QA-Enabled test methods are provided.

Quality Testing

We all have an instinctive idea of ‘quality’. Better ingredients and better preparation make for a tastier meal. A gadget with clean lines and polished surfaces seems better than one with rough edges. Beyond our basic notions, there is a scientific field of study around quality in product development. Quality is about providing better products, and lowering the risks associated with delivering less than optimal products. Two key concepts in quality testing are:

- **Quality Assurance (QA):** Ensuring that the integrity of the design is upheld throughout the process of building the product; and
- **Quality Control (QC):** Inspecting the product at the end of the development process to ensure that it meets acceptable standards, and so can be safely delivered to the consumer.

Applying rigorous QA methods throughout development increased the likelihood of discovering and correcting mistakes early on, and thus facilitates a relatively incident-free final QC validation step. QC works best at the end when QA has been previously applied; but QC without prior QA is a recipe for finding mistakes after the product has been fully assembled, just before the product is about to be packaged for delivery. This is the worst time to find mistakes, and at this stage, the desire to deliver on time can mean that faulty products reach the consumer.

The most attention grabbing consequences of poorly applied QC are those that cause fatalities (think of jumbo jets, buildings, bridges), or very hazardous (think of self-combusting lithium batteries in gadgets, spontaneous airbag inflations). For regular products and services, the consequences are less dramatic but they are detrimental to businesses (think of customers leaving

bad online reviews and taking their business elsewhere... the chef who tries to do QC but ignores QA is quickly out of a job).

What does this have to do with accessibility testing of Information and Communications Technologies (ICT)? For a long time, accessibility teams in business have gravitated towards models of operation that could only be used for QC. This has resulted in a 'chase your tail' culture of accessibility and have proven ineffective in many organizations. Newer testing methods are available that are QA-Enabled, and implementing them during development facilitates accessibility teams taking on a valid and practical QC role.

What are 'QA Enabled' Test Methods?

Products are comprised of subcomponents. For an airplane, the subcomponents are things like the engines, the fuselage, the cockpit. Each subcomponent has further subcomponents: the cockpit has seats, controls, doors. Eventually you get to individual components such as nuts and bolts, labels, and lights. The airplane only works as a product when all of the subcomponents are assembled, but subcomponents can be QA tested throughout the stages of development from conceptual design to final delivery of each subcomponent. The tools for testing aircraft components include environmental chambers for extreme temperatures, vibration tables, and wind tunnels. Aircraft are expensive to build, partly because the risks associated with product failure are so severe, and the product requirements mean that testing tools are highly specialized.

Software usually differs in scale and in the level of risk, but otherwise is no different: available subcomponents (subroutines, APIs, etc.) are combined to form other subcomponents (the database, the data capture method, the data processing, the user interface, etc.), and this is assembled into the software product. The typical method of testing and inspection for programmers is to employ other software that reveals how their new code works. For inspecting code syntax and adherence to rules, code inspection software can raise flags for missing, repeated, or otherwise incorrectly formatted elements, much the same as a spell checker operates in a word processing application. For software security, known vulnerabilities can be highlighted with code inspection tools, helping programmers ensure that they cover up any likely routes and guard against hackers. Programmers may be using code inspection tools for their own QA during development, or testing teams may be using similar code inspection tools as a QC check before passing a subcomponent to the next area of the build.

Over the past decade or more, a large number of software-based code inspection tools have been developed for checking the accessibility of software, web, and mobile device interfaces. The tools provide a representation for the programmer that has the source code overlaid with errors and omissions, such as missing alternate text on an image, or a missing target for a 'skip navigation' link. Successful use by programmers and software testing teams of accessibility code inspection in QA and QC requires a number of essential elements to be implemented:

1. Tools for conducting accessibility code inspection tasks.
2. Procedures for using the tools to find coding errors.
3. Reporting mechanisms (bug tracking).
4. Training on how to use the procedures and tools and reporting mechanisms.
5. Test requirements throughout the systems development life cycle (SDLC).
6. Gate review criteria for the SDLC.
7. Staff position(s) with responsibility and accountability for the accessibility elements of QA and QC in the SDLC.

For this last item on the list, the staff overseeing accessibility in the SDLC would be responsible for ensuring that the test procedures and tools were up to date and appropriate for the type of development and testing activities being done by the programmers and testers. The responsible staff would usually track and report on the number of defects being found with bug tracking, track the results of gate reviews, and provide final QC sign-off of final products.

Note: To find available QA-Enabled test tools and methods, see the section at the end of this paper.

Comparisons to a Traditional Test Method

An odd request

Manager: “Jeremy, I just found out that this software interface has to work in the arctic. Can you take it down to the environmental chamber and run an extreme cold weather test?”

Jeremy: “Um. No. I’m just a programmer. I have no idea how to use the environmental chamber, and I don’t really want to get frostbite.”

Manager: “That’s okay. Here’s a manual. And there’s some videos on YouTube on how to do it. You should be up and running by lunchtime, and then you can run all the tests from now on.”

Jeremy: “Didn’t we have an environmental tester who had six months of specialist training and three years of experience?”

Manager: “We did, but he left last week. You’ll have to do the best you can. Let me know the results tomorrow.”

This sounds like a very odd request. What does this have to do with accessibility? Let’s consider the same sort of scenario, but with a traditional accessibility test method:

Manager: “Jeremy, I just found out that this software interface has to work for people who are blind. Can you go the accessibility lab and run it through a screen reader?”

Jeremy: “Um. No. I’m just a programmer. I have no idea how to use a screen reader. What is a screen reader anyway?”

Manager: “Oh, it’s software that speaks what’s on screen for people who are blind. I think there’s an online manual. And there’s some videos on YouTube on how to do it. You should be up and running by lunchtime, and then you can run all the tests from now on.”

Jeremy: “Didn’t we have an accessibility tester who had six months of specialist training and three years of experience?”

Manager: “We did, but they he last week. You’ll have to do the best you can. Let me know the results tomorrow.”

Not QA-Enabled

Using a screen reader for testing is a specialist task. It requires:

- Learning how screen readers work (there are long lists of keyboard commands to learn), and learning how screen reader are used by blind and low vision users;
- Choosing appropriate user settings for the screen reader;
- Making sure the screen reader software is up to date;
- Learning how to use the screen reader tool to find errant utterances of the screen reader, from which you can infer errors in coding;
- Learning when errant utterances are the result of screen reader user settings, or version compatibility issues with screen readers.
- Learning how to document errors so that they can be remediated by programmers.

It takes a long time to learn how to use screen readers and other assistive technologies (AT) in order to be able to test. The likelihood of Jeremy succeeding in his testing task as an untrained novice is low. Programmers do not use specialist tools like environmental chambers. Neither should they be expected to use specialist tools such as AT. In the days before code inspection tools and established test methods to use those tools, testing with AT was the norm, and by necessity was conducted by specialists operating outside of the mainstream programming and testing teams. The problem is that testing with screen readers and other AT is inefficient, time consuming, and is not QA-Enabled.

Illustrative comparisons

Training to become a tester

In training terms, both AT inspection and code inspection methods have a high initial level of effort. After the high level of initial effort and a short amount of time the code inspection trainee can become proficient, but the same is not true for AT inspection. For AT inspection, there is a very long period of high effort just to learn all of the various specialist aspects of testing tasks.

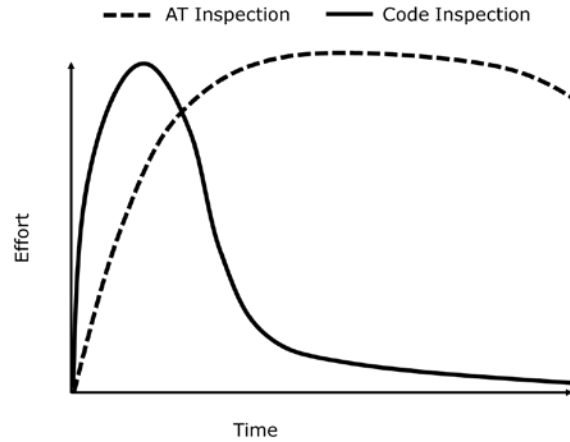


Figure 1 – Time and effort to become a proficient tester

Time to complete tests

Screen readers speak one word at a time. Even if you speed up the speech rate, they still speak only one word at a time. The time to assimilate and test a screen’s worth of information is long because the data pipe is akin to a soda straw. Add to that the time that it takes for to remember, from a very long list of commands, the specific keystrokes to control the screen reader. Further add the time it takes for the tester to compare what is displayed on screen with what is spoken by the screen reader, and then make their evaluation. Testing with AT takes a much longer time per test than with code inspection. In code inspection, the tester chooses a test command (e.g., show the code for headings), the code is instantly revealed (e.g., two headings are tagged; but three other headings fail the test), and the bug report is made. Immediately after training, the amount of time that it might take to test a whole product may take a novice just as long as it takes an AT tester to go through their checklists. However, this amount of time rapidly drops once the code inspection test tools and methods are learned. The same is not true for the one-word-at-a-time AT inspection.

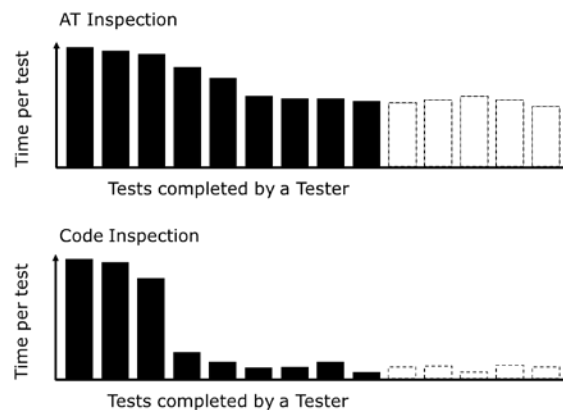


Figure 2 – Time per completed test

Completeness of the product

When your specialist AT inspection team resides outside of your mainstream development team, there is usually no ‘room’ in the SDLC for stopping, waiting for the AT inspection to occur, and then resuming with development. The long time that it takes to test individual products, and the need for the AT test teams to be supporting several product development efforts, means that scheduling accessibility tests during development is often difficult. An additional consideration is that many software elements will not work with screen readers until they become ‘complete’ products. As a result, accessibility testing occurs only at the end of the development project, at or near the time that the product is due to ship. Testing with AT, effectively, is adding a QC step to a development process that features little or no QA.

Locus of responsibility

Programmers are much more likely to adopt (own) a code inspection tool and test method if it means they can test their own code during development, and have less risk of a hold-up caused by an external accessibility test team close to the deployment stage. With ownership comes responsibility.

When the locus of responsibility for accessibility testing falls *outside* of the mainstream development team, as is typically the case when AT inspection methods dominate, the mainstream test team has to relinquish control and responsibility. Most programmers are not good at doing this (Cooper, 1999).

When the locus of responsibility for accessibility testing falls *within* the mainstream development team, as can be the case with QA-Enabled testing, the *outside* accessibility testing team can then assume a QC and QA support role. In this role, they assist developers to achieve their assigned tasks of making their products accessible. This is a much more effective use of the accessibility team’s time than the ‘tail chasing’ option of trying to test products that are beyond the point of incorporating major updates.

Impact on product design

“[Accessibility Program Offices] demonstrated very little control over design decisions that directly affected the accessibility of the final product” (NCD, 2004, p.190).

The above quote is over a decade old, but unfortunately it is still just as relevant today.

The only way to impact the accessibility of a given software product is to change the code that drives it. Programmers are the ones who control the code, not management, not executives, and not separate specialist accessibility teams. The programmers have to be required and responsible, and more importantly have to *want* to change the code (Cooper, 1999).

By relinquishing accessibility testing tasks to programmers and other developers, the existing accessibility team acting in a QC and QA support role should have a lot more time available to conduct other tasks that could positively affect the accessibility of final products. For example, the accessibility team could spend more time on awareness and training efforts, and could use their AT expertise to test the usability of products by people with disabilities. (In many organizations there is a usability testing team, and an accessibility testing team. However, the usability team typically works on nondisabled interaction issues, while the accessibility team

only tests for standards conformance. Actual testing for usability of products with AT used by people with disabilities is rare.)

Testing as the Logical Hook for Organizational Change

There are a small but growing number of published resources targeted at accessibility professionals and those wanting to adopt accessibility as a strategy throughout their organization. These resources address many different facets of business (policy, training, needs requirements, legal requirements, the business case, etc.). However, deciding where to make a start can be a difficult choice in practice.

What seems to work most effectively as a starting point is the introduction of code inspection as the responsibility development teams, with those teams supported by a QA team (DHS, 2015a). Once development teams realize that (a) accessibility testing is much more straightforward and much less specialized than in the past, and (b) they can successfully adopt it, we find that development team members are then much more amenable to other organizational changes.

Testing is the best hook because it works on what programmers already know and do. In any organizational change process, making the case that developers can and should test their own products is a useful first step from which other organization-wide changes can follow.

Resistance to change

Even though code inspection based testing is a useful first hook, the advice from those working in the area of organizational change is that executive support is actually *the* first step (DHS, 2015b). The reason is that change does not come from someone in the accessibility team politely asking the development team to adopt new testing methods. Instead it comes from having the responsibility and accountability to get jobs done, and that comes about as a result of executive decisions.

In our experience, resistance to the introduction of QA-Enabled testing will come from two areas of the organization. First, the developers can have initial reflexive objections to anything that on face value sounds like more work. Second, established accessibility teams that have relied on AT inspection methods to date may incorrectly perceive that the need for their services and skills is becoming obsolete. Careful management of any change process is needed for heading off and tackling any resistance to change from those involved with the development and testing of products.

Sacrificing the Purple Cow

Noted marketing author Seth Godin describes the concept of the 'Purple Cow' (Godin, 2003). The concept is that a cow is a cow is a cow, and no matter how many cows you pass by, there's little reason to stop and take a picture. But if you encounter a Purple Cow, now that's *remarkable*. Godin's point is that in order to stand out in the marketplace, you need something remarkable, and therefore you need to be developing the next Purple Cow.

Many accessibility specialists have seen looks of amazement and curiosity from mainstream developers as they have sat in on accessibility testing experience sessions employing AT inspection. They are initially amazed at the concept of screen readers. They may wonder aloud “why didn’t they learn about this in undergraduate computing class?” To them, the screen reader is a Purple Cow. (Watch the mainstream news long enough, and you will see articles on ‘amazing’ technologies that people with disabilities use. More remarkable Purple Cows.)

We need to make accessibility testing *unremarkable*. It should be part of day to day business in the same way that checking for code syntax or security is a day to day task for programmers. Screen readers are great for testing usability of interfaces for use by people with low vision and blindness. They used to be one of the only ways to test for accessibility, and even though they took a long time and had to be used by specialists, it was better than nothing. But now code inspection has overtaken the screen reader in terms of proficiency and effectiveness. It’s time to drop the screen reader for the Purple Cow that it is. Getting mainstream developers to test their own accessibility as a matter of course and without a word of objection? *That* would be remarkable.

Finding QA-Enabled Test Methods

The World Wide Web Consortium (W3C) Web Accessibility Initiative (WAI) provides a list of tools with search filter capabilities. The filters can be used to narrow down the list to find code inspection tools:

[Web Accessibility Evaluation Tools List](http://www.w3.org/WAI/ER/tools/)
(<http://www.w3.org/WAI/ER/tools/>)

The Trusted Tester program at the Department of Homeland Security (DHS) is a code-inspection based package comprising:

[Trusted Tester Program Overview](https://www.dhs.gov/trusted-tester)
(<https://www.dhs.gov/trusted-tester>)

[Section 508 Compliance Test Process for Applications](https://www.dhs.gov/publication/dhs-section-508-compliance-test-processes)
(<https://www.dhs.gov/publication/dhs-section-508-compliance-test-processes>)

[Trusted Tester Online Certification Training](https://www.dhs.gov/publication/trusted-tester-resources)
(<https://www.dhs.gov/publication/trusted-tester-resources>)

References

Cooper, A. (1999). *The inmates are running the asylum: why high tech products drive us crazy and how to restore the sanity*. Indianapolis, IN: SAMS.

DHS. (2015a). *Conducting your own Unified Testing for Accessibility Project*. Washington, DC: Department of Homeland Security

DHS. (2015a). *The Unified Testing for Accessibility Project (UTAP): Pilot*. Washington, DC: Department of Homeland Security

Godin, S. (2003). *Purple Cow: Transform Your Business by Being Remarkable*. Portfolio.

NCD. (2004). *Design for Inclusion: Creating a New Marketplace*. Washington, DC: National Council on Disability.

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International ([CC BY-NC-ND 4.0, https://creativecommons.org/licenses/by-nc-nd/4.0/](https://creativecommons.org/licenses/by-nc-nd/4.0/)).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Four Things You Should Know Before Testing with VoiceOver on Mac

Jonathan Whiting

WebAIM.org
6807 Old Main Hill
jonathan@webaim.org

Abstract

The VoiceOver screen reader is a very capable screen reader that is used by many developers to test for screen reader accessibility. However, there are four important things that web developers and accessibility evaluators should understand before testing with VoiceOver:

1. The majority of desktop screen reader users are on Windows.
2. VoiceOver does not have a forms or application mode.
3. Default keyboard accessibility settings must be changed.
4. ARIA descriptions are read after a very long pause in older versions of VoiceOver.

Introduction

Web professionals are increasingly using Macs. As they become more aware of the importance of web accessibility, more and more of them are using VoiceOver to test for screen reader accessibility. This is excellent news, but there are at least four important differences between VoiceOver and most Windows screen readers that accessibility testers should understand.

1. The majority of desktop screen reader users are on Windows.

WebAIM regularly conducts surveys of screen reader users. The most recent [WebAIM screen reader user survey \(webaim.org/projects/screenreadersurvey6/\)](http://webaim.org/projects/screenreadersurvey6/) was conducted in July 2015, with over 2500 respondents. Of the respondents, 93% use a screen reader due to a disability. While this sample was not controlled and may not represent all screen reader users, it is still a valuable resource since more rigorous large-scale studies are not available at this time.

When respondents were asked which screen readers they use on their desktop or laptop, 8% of respondents indicated that VoiceOver was their *primary* screen reader and 31% reported that they *commonly* use VoiceOver (rounded to the nearest whole number). While these numbers suggest VoiceOver is a fairly common screen reader, the fact is that over two-thirds of respondents do *not* commonly use the Mac Operating System. Although you can gain a great

deal of insight when testing with VoiceOver, relying on it alone will not give you a complete picture of screen reader accessibility.

Although this paper deals with VoiceOver for Mac, it is worth noting that VoiceOver is by far the most common *mobile* screen reader. In this same survey, 57% of respondents indicated they commonly used VoiceOver for their mobile needs. TalkBack for Android was a distant second at 18%.

2. VoiceOver does not have a forms or application mode.

Windows screen readers have different interaction modes when they are reading through content vs. interacting with forms and widgets (Watson, 2014). These modes have different names in each screen reader, but for the sake of consistency, they will be called ‘read mode’, ‘forms mode’, and ‘application mode’.

Read mode

In read mode, the arrow keys are used for reading. For example, the down arrow reads the next item or section while the right arrow reads the next letter. There are also a number of shortcut keys that can be used to jump to common page elements. For example, the H key moves to the next heading and T moves through tables. On most websites, screen reader users spend the majority of their time in this mode.

Forms mode

When navigating into a form control, the screen reader typically uses an audio tone to indicate that it has entered forms mode. This mode allows the user to interact with form controls using the keyboard. Pressing the H key in forms mode, for example, will type this letter into a text box instead of moving to the next heading. Pressing the down arrow will select the next radio button in a group instead of reading the next item. A different audio tone announces when a screen reader leaves forms mode and returns to read mode. Most screen reader users are familiar with this mode.

Application mode

In order to interact with many online “widgets,” the screen reader must stop using keystrokes for reading and instead pass them on to the browser. For example, in an accessible tree menu, users should be able to move between sections with the up and down arrow keys and expand or collapse these sections with the right and left arrow keys (keystrokes normally reserved for reading).

Functionally speaking, forms and application mode are the same. In fact, they are the same in some screen readers. However, for developers and testers, there is still an important difference: HTML form controls trigger this mode automatically, and this process is understood by most screen reader users. For application widgets, this mode must be triggered with the correct ARIA roles. This is not as well understood by many screen reader users, and this confusion is at the heart of many accessibility issues.

ARIA, screen reader modes, and VoiceOver

Certain ARIA roles will tell a screen reader to change from reading mode to application mode. These roles include `menu`, `menubar`, `tab`, `tablist`, `grid`, `toolbar`, and of course `application`. ARIA roles that trigger application mode are often misused by well-meaning developers. The misuse of ARIA and lack of understanding of screen reader modes currently results in significant accessibility issues on the web. The prevalence of these issues is on the rise.

For example, if a group of links at the top of a webpage have incorrectly been assigned an ARIA `role="menu"`, this triggers application mode and the screen reader user cannot read through this area. Alternatively, if a developer creates a menu that is controlled with the arrow keys and it does not have the appropriate ARIA roles, a screen reader user may move right past it without knowing what it is or how to interact with it.

VoiceOver does not have the single-key controls that are common in Windows screen readers. For instance, to move to the next heading in a Windows screen reader, you press the H key. To move to the next heading in VoiceOver, you use `command + control + option + H`. To read the next item in VoiceOver, the user presses `control + option + right arrow`. Because of these required key combinations, there is no need for VoiceOver to toggle between modes. This is good news for VoiceOver users, who may not experience issues when the above roles are misused. However, this is not ideal for testing because an incorrect ARIA role that can make a webpage highly inaccessible for the majority of screen reader users may go undetected when testing with VoiceOver.

3. Default keyboard accessibility settings must be changed.

In modern Windows browsers, pressing the tab key will move through all links and form controls (text fields, checkboxes, buttons, etc.). But by default, not all of these elements receive keyboard focus on a Mac. Mac keyboard accessibility can also vary when VoiceOver is running and when it is disabled. To ensure consistent accessibility across all major Mac browsers, do the following two things:

Enable full keyboard accessibility in System Preferences.

To enable full keyboard access, select **System Preferences > Keyboard > Shortcuts**. At the bottom of the page, there is a setting that reads “Full Keyboard Access: In windows and dialogs, press Tab to move keyboard focus between:”. By default, the “Text boxes and lists only” option is selected. Change this to “**All controls**”.

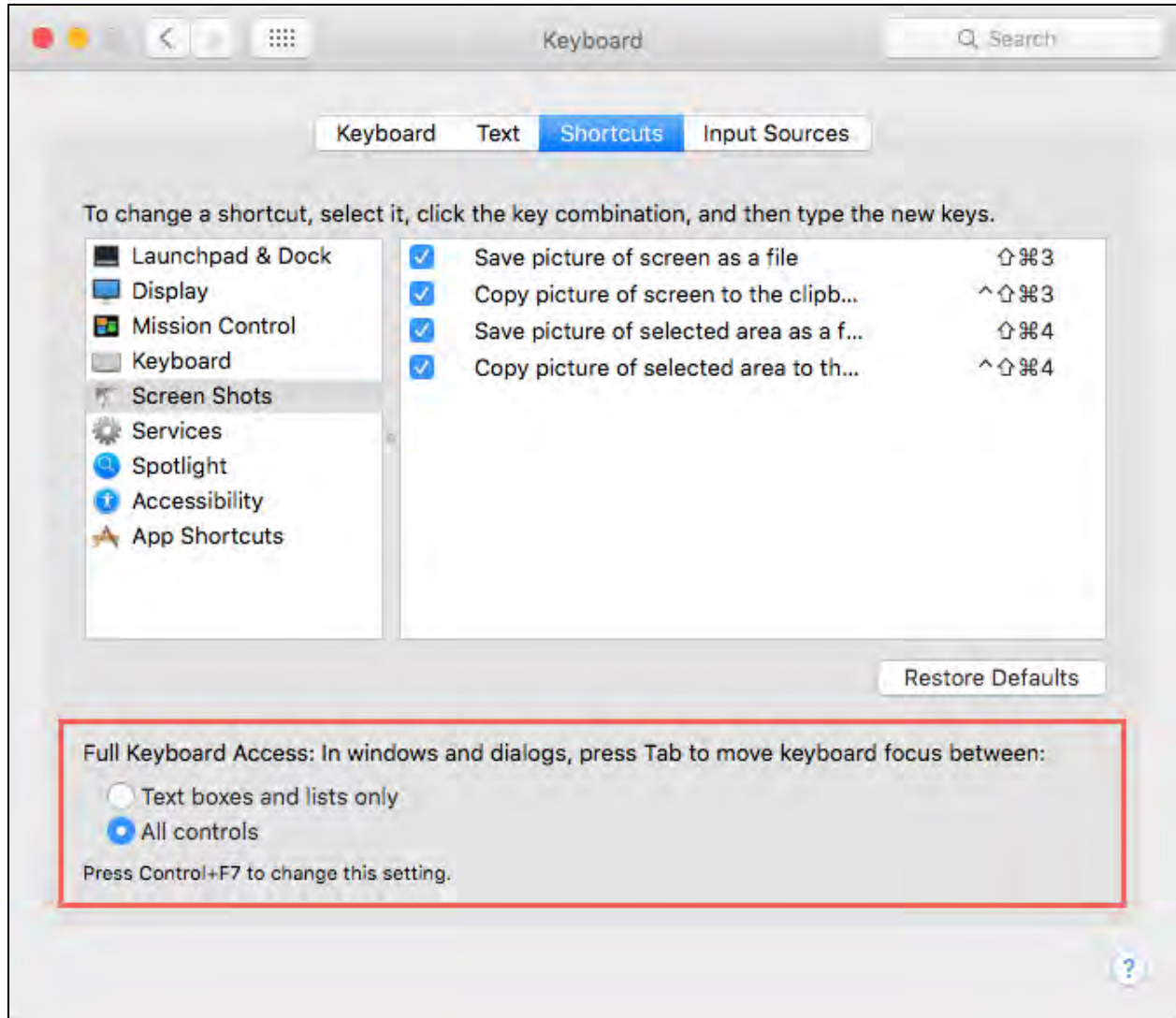


Figure 1: Screenshot of keyboard access settings.

Note: If this setting is not changed manually, it will automatically change to “All controls” every time VoiceOver opens and then revert back to “Text boxes and lists only” when VoiceOver closes. That change in behavior can be very confusing when testing keyboard accessibility without a screen reader.

Enable full keyboard accessibility in Safari.

Historically, accessibility support has been best in the Safari browser. However, links cannot receive keyboard focus by default in Safari, even with the above system preference enabled. To address this, in the **Safari** menu, select **Preferences** > **Advanced** > **Accessibility** and check “**Press Tab to highlight each item on a webpage**”.

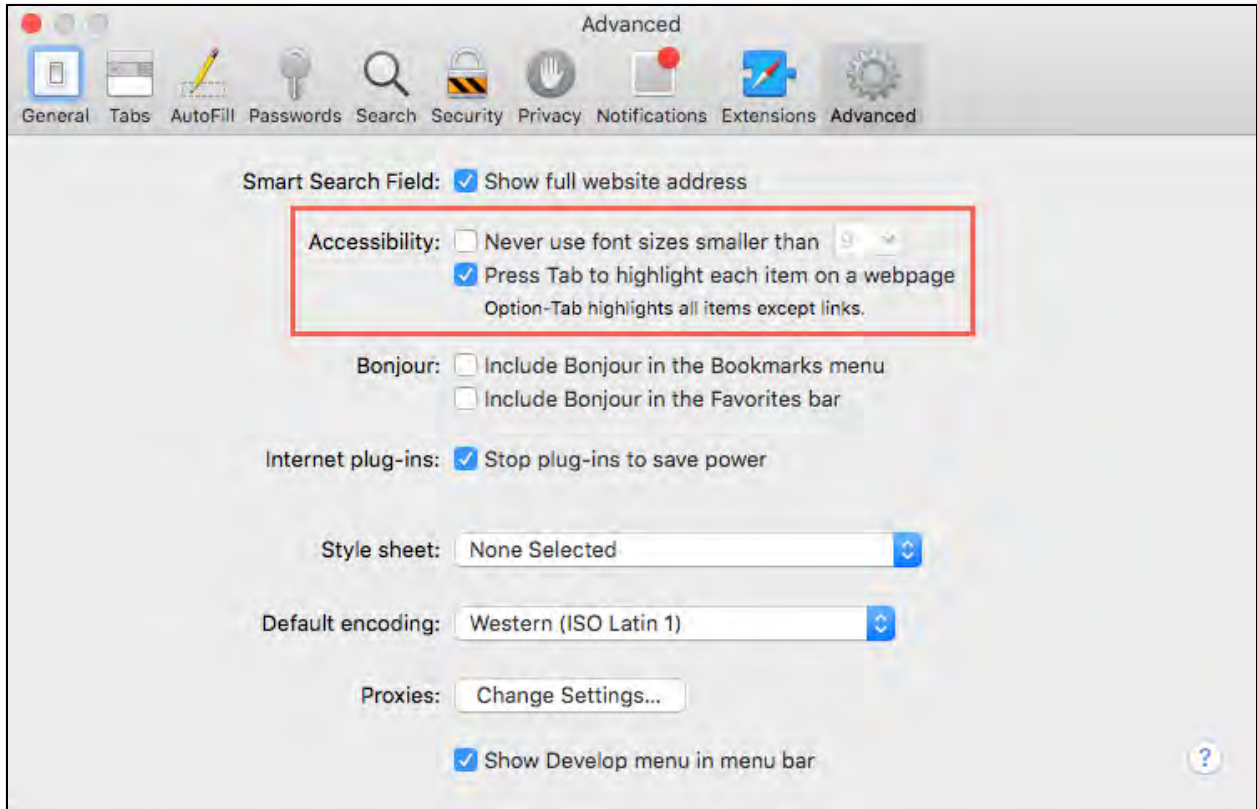


Figure 2: Screenshot of Safari keyboard accessibility settings.

(Beckall, 2014)

4. ARIA descriptions are read after a very long pause in older versions of VoiceOver.

Often there are times when a text description should be associated to an element (usually a form control). For example, below in Figure 3 there is a form field with a message of “Must be 8-15 characters and have letters and numbers”:

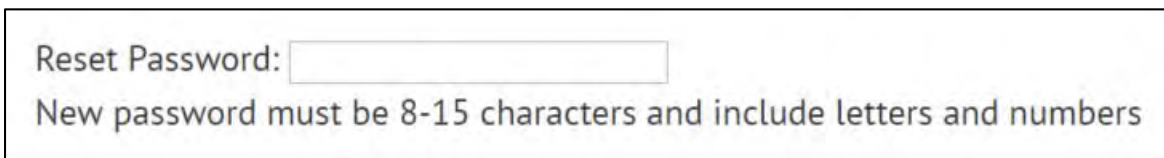


Figure 3: An example of a form field that has a label and description.

This additional information should be read by a screen reader when the password field is encountered, which is accomplished with the `aria-describedby` attribute.

In older versions of the Mac operating system, there is an issue when reading ARIA descriptions—there is a very long seven second delay after a control is focused before reading

ARIA descriptions. This can cause users and testers to miss this important information. It also leads to a common but incorrect practice of avoiding `aria-describedby` completely and using `aria-label` or `aria-labelledby` to present labels and descriptions. Fortunately, this issue was addressed in a September 2015 update (OS X 10.11, El Capitan).

If you encounter this issue during testing and you cannot update to a newer version of macOS, this delay can be shortened to less than half a second with VoiceOver settings. To shorten this delay, open your **System Preferences** and choose **Accessibility > VoiceOver > Open VoiceOver Utility > Verbosity > Hints**, and then drag the slider down on “**Speak hints after delay**”.

Conclusion

VoiceOver is a very powerful screen reader and its usage is on the rise. In order to most effectively use VoiceOver to test web accessibility and to best ensure an accessible experience for all screen reader users, VoiceOver users should understand these differences and consider additional testing in Windows screen readers.

References

Beca11y (2014, July 7). Keyboard Navigation in Mac Browsers. Retrieved from <http://www.weba11y.com/blog/2014/07/07/keyboard-navigation-in-mac-browsers/>

King, Matt (2016, September 11) Re: Three things you should know before using VoiceOver for testing [Blog comment]. Retrieved from <http://webaim.org/blog/three-things-voiceover/>

Watson, L. (2014, September 21). Understanding screen reader interaction modes. Retrieved from <http://tink.uk/understanding-screen-reader-interaction-modes/>

WebAIM (2015, August 28). Screen Reader User Survey #6 Results. Retrieved from <http://webaim.org/projects/screenreadersurvey6/>

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Writing Accessibility Tests

Alistair Garrison

Senior Accessibility Engineer, SSB BART Group
1593 Spring Hill Rd. Ste. 720, Vienna, VA 22182
alistair.garrison@ssbbartgroup.com

Jonathan Avila

Chief Accessibility Officer, SSB BART Group
1593 Spring Hill Rd. Ste. 720, Vienna, VA 22182
jon.avila@ssbbartgroup.com

Abstract

This paper focuses on the methodology of writing accessibility tests, identifying testing techniques, determining a good testing interface, and describing the role of automatic (unit, continuous, and on demand) and manual testing. The application of machine learning algorithms will be introduced as a means to push the boundaries of what can currently be tested automatically.

The complexity of modern web pages can mean that accessibility testing and reporting takes several hours per page, even when automated tools are used to support manual inspection and testing with Assistive Technology (AT). This presentation will discuss approaches and techniques that can be used to speed up the process and make it more uniform, scalable, and repeatable—techniques such as automated issue detection and remediation, even in live content, based on code or object signatures. Additional concepts such as user story capture / replay and automatic bug logging will also be discussed.

Testing Techniques

Following well-established software testing best practices

Accessibility testing is effectively an extension of User Acceptance Tests—in effect, ascertaining if a product (e.g., web site or web application) enables users, including persons with disabilities—to complete all specified tasks, whether are using Assistive Technology (AT) or not.

Accessibility tests are written to ensure positive accessibility features have been implemented or that negative features have been avoided. Ideally, features are defined in terms of testable statements that can be made about the product.

Positive feature: A skip link is provided at the top of a page that goes directly to the main content area.

Negative feature: No displayed content in a page flashes more than three times per second.

As such, accessibility tests can be written following the same well-established software testing best practices that we use to write general acceptance tests, for example:

1. When covering a feature, write a test procedure that contains several simple test steps rather than a few complex test steps.
2. Ensure each test step checks for one thing.
3. Write each test step in plain English (avoiding pseudo-code).
4. Ensure each test has an expected result that is clearly defined.
5. Ensure each test is independent. It should be able to run on its own, not affect other tests if changed, and not be affected by changes to other tests.

Of course, tests should also be applicable to the technology platform used and should provide specific tools or methods to verify the test and that the expected results have been met.

Testing approach for “often very broad” accessibility requirements

Most accessibility tests are designed to test a product's compliance with a known accessibility standard such as WCAG 2 Level A and AA (W3C World Wide Web Consortium, 2008) or Section 508 standards (Electronic and Information Technology Accessibility Standards; Final Rule, 2000).

These accessibility standards can contain requirements that are often very broad (e.g., WCAG 2 SC 1.1.1 Equivalent Non-Text Elements).

There are at least two possible solutions for testing such “broad” requirements.

The first approach is to try and test the “broad requirement” by having one long test, generally defined as a decision tree, that each relevant node is run through in order to determine if the requirement has been passed or failed. If it is an `img` node, do this; if it is an `input type= image` node, then do that. Processes such as these can be multiple written pages long, and tell a user if the overall requirement is passed, failed, or non-applicable.

Note: The term “node” is used in place of “element,” as the large majority of testing is done on the Document Object Model (DOM), which consists of a tree of nodes. In essence, an `img` element in the mark-up should create an `img` node in the DOM.

The second approach is to determine a set of relevant techniques that can be used to prove a “broad” requirement has been met, and to create sets of smaller tests that can determine if these individual techniques have been properly implemented.

The benefits of the second approach are as follows:

1. **Flexibility:** The addition of new techniques (thick and fast with mobile) can be handled easily, as each technique is supported by sets of independent tests. This contrasts with the tree approach, which is easily fractured if change is needed.
2. **Reporting:** Developers can be informed of which techniques were not properly implemented and asked to re-implement them. This contrasts with being told that content *x* has failed (the very broad) requirement *y*.

Using Known Failures and Sufficient Techniques that are accessibility supported as Guidance

Most accessibility tests are designed to test a product's compliance with a known standard (e.g., WCAG 2 Level A and AA, Section 508 standards). Generally, the tests written by SSB BART Group follow the second approach (described above) and determine if accessibility-supported techniques, known to work well with common user agents and assistive technologies, have been properly implemented to create positive accessibility features in a product and avoid negative features.

For example, when seeking to conform with WCAG 2 SC 1.1.1 Equivalents Non-Text Elements (<https://www.w3.org/TR/WCAG20/#text-equiv>), a developer might choose to implement one of several available WCAG 2 Sufficient Techniques, for example, “H37: Using alt attributes on `img` elements.” (<https://www.w3.org/TR/WCAG20-TECHS/H37.html>) This would ensure a product in which each `img` node has a text equivalent (accessible name).

If the techniques used to implement features are known, it is best to create and follow the test procedures defined for those specific techniques.

However, as testers, we are often divorced from the knowledge of which technique was implemented. In this case, we have to think about the product feature the technique is aiming to create (or avoid), and create a test for that feature.

For example, a properly implemented `aria-labelledby` attribute, `aria-label` attribute, `alt` attribute or `title` attribute can be used to provide a text equivalent (accessible name) for an `img` node. In order to cover all bases, we might create a negative test that allows us to state, “one or more `img` nodes do not have a mechanism that allows an accessible name value to be calculated.”

Note: This test is purely to determine if a text equivalent has been provided, not if it is descriptive. For example, the text equivalent might say, "Image" instead of "A young girl with long brown hair sitting in a tire swing." This would pass an automated test, but does not meet accessibility standards.

That said, if we have sets of tests that cover both known techniques and desired / undesired outcomes, it allows us to say that an outcome was achieved using a tested-for technique or an unknown technique that could theoretically be submitted as a new WCAG 2 sufficient technique?). But, more importantly, it means we can say to a developer, "We didn't achieve the

outcome using this technique, but we think we'd show better practice / achieve greater reach with this other technique instead."

Complexity of Seemingly Straightforward Tests

The text equivalent (accessible name) for an `img` is calculated using the accessible name calculation [http://rawgit.com/w3c/aria/master/acname-aam/acname-aam.html#mapping_additional_nd_te] for the `img` nodes. It takes into account different ways that can be used for providing a text equivalent for images, such as the `aria-labelledby` attribute, `aria-label` attribute, `alt` attribute, or `title` attribute.

However, what seems like a simple check is not such a straightforward process when we consider images may be marked as decorative by using `role="presentation"`; or by using `alt=""`.

In this case, we have to modify our test to become: "One or more `img` nodes (excluding those that have an `alt` attribute set to a null value) does not have a mechanism that allows an accessible name value to be calculated."

Candidate Sets

Optimally, we only want to test content which is present and relevant to the test. At SSB BART Group, we decide if tests are applicable by first collecting a "candidate set of nodes." If there are no nodes collected in the candidate set, the test is not run.

In our `img` node example above, the candidate set would be `img` nodes; however, it excludes those that have an `alt` attribute set to a null value.

Candidate sets can be created by using CSS selectors and are generally written in JavaScript using the `document.querySelectorAll()`, or `document.querySelector()` functions. CSS selectors are powerful methods of including and excluding particular combinations of nodes and attributes.

The selector for our candidate set would be: `img:not([alt=""])`. In other words, `img` nodes, excluding those that have an `alt` attribute set to a null value).

Other Factors

Numerous other factors may also affect which nodes are tested. For example, nodes that are not shown due to being affected by the CSS style of `"display:none"` may be deemed to be excluded from the candidate set.

As such, our test would need to be modified to "one or more `img` nodes (excluding those that have an `alt` attribute set to a null value), available in the DOM, does not have a mechanism that allows an accessible name value to be calculated."

Here “available in the DOM” would include nodes that are not affected by the CSS style of `display:none` .

In likelihood these `display:none` nodes may be shown only in some defined circumstances (e.g., based on user action or in different responsive views). Thus, tests may need to be run in different states of the page (e.g., different DOM states), in different responsive breakpoints, and in different browsers (user agents). These different states may be testable through Behavior-Driven Development QA tools that run tests written in Cucumber or Jasmine; or they can be queued up for automatic testing by manually collecting the different states.

Machine Learning

Certain Machine Learning (ML) algorithms within the field of Natural Language Processing (NLP) allow us to test if text is descriptive or not. With regards to accessibility testing, we may be able to utilize such ML algorithms to automatically test the more subjective issues which were previously only testable by humans. For example, we might be able to determine if link text is meaningful in and out of context. We can also determine what text is truly equivalent for an image within the context of the surrounding text, and its placement in the page.

You may have noticed such Machine Learning in your Facebook news feed lately. If your Internet connection is slow and an image cannot load, you will see `alt` text that gives its best guess as to the content of that image. ("Image may contain: person, car.")

We might also allow computers to determine if content is used according to its specification (for example, if a table is used as a data table or for layout). Depending on the type of table, different requirements apply. Thus, if a computer can determine the purpose of the table, the correct tests can be run on the table.

ML algorithms can also be used to group similar images or spot text inside images. This automated categorization of images will allow more and more manual tests to be automated.

Guided Tests

Even with the advances in Machine Learning, there will still be tests that require a defined candidate set to be examined by a human. For example, ML may be able to recognize an image that conveys complex meaning, but we may still have to ask a user if the surrounding text conveys all the information if a full text description is not programmatically identified.

That said, if the full text description is identified programmatically (e.g., by an `aria-describedby` attribute or `longdesc` attribute), we could have an automated test check if the equivalent text was a reasonable summary of the full text description.

The big question moving forward is how we can utilize ML more and more to replace manual tests and also how can we encourage developers to adopt the more inherently auto-testable techniques.

JavaScript Testing Libraries

There are a number of Behavior-Driven Development QA tools that enable a user journey to be codified and then played into a browser, which allows testing to be undertaken on each browser DOM state reached.

The relatively recent realization that such tools could be highly useful when testing accessibility has led to the development of a number of self-contained, pure JavaScript testing libraries.

Once included in a page (or view) through the inclusion of a script tag, these testing libraries allow tests that move the browser through BDD QA tools and make assertions based on their collected accessibility test results.

Unit Testing

The simplest use case for a JavaScript testing library would be to include it in a page that is being built and to use the “run tests on file change” functionality in a test runner such as Karma. When a change to the page is pushed, it will run a Jasmine unit test that asserts “no accessibility issues.”

This would mean that a developer will be alerted immediately if they have just coded something that contains an accessibility error, and they can make the necessary changes immediately, ensuring the lowest possible cost for remediation.

Continuous Integration (CI)

Traditionally, testing for accessibility occurred after the fact and was carried out by accessibility consultants or QA testers. It is important to bring accessibility testing into the full software development lifecycle by incorporating it into the processes of checking in code and build creation.

Another simple use case for a JavaScript testing library would be to ask developers to create BDD tests (e.g., Cucumber tests) that move a browser to each DOM state reach when moving through a user journey, and then run accessibility tests on each.

These tests could be included and run as part of a smoke test prior to committed code being allowed to enter Continuous Integration; or they could be run as part of the regression testing.

If developers are blocked from checking in code that doesn't pass internal or security requirements, then the same standard should be held for accessibility.

The best practice is to run all tests daily (in reality, nightly, as running BDD tests is intensive), to ensure that if accessibility is broken in a daily build that the red flag is raised.

Auditing using a JavaScript Testing Library

When inserted into a system designed for accessibility auditing, a JavaScript testing library should allow its tests to be run in accordance with any filters being applied, such as by standard, technique, severity, failure or warning, or noticeability.

The auditor should also be free to choose to test different things: from something as small as an HTML element or a widget, to part of the DOM tree or even the whole page.

Testing will always be of the live document object model (DOM) of the page and not the source code.

Outside of the JavaScript testing library, the interface for the accessibility auditing tool should allow code issues / image issues to be stored from a page (possibly with JIRA integration); and also provide visual tools to highlight page components assisting in locating issues quickly.

Discussions with testers indicate that different people test in different ways. For example, one tester or developer may look at their page structure first and then evaluate for keyboard access. A second tester may evaluate for keyboard access first. A third may test keyboard and structure for each component separately.

Conclusion

The selection of all failure and relevant sufficient techniques provides a useful way to derive full-test coverage for the “broad” requirements in accessibility standards. Well-described software testing best practices can be used to shape tests and test steps. Machine Learning algorithms will allow more and more manual tests to be fully-automated moving forward.

The adoption of Behavior-Driven Development (BDD) tools in accessibility testing, coupled with JavaScript testing libraries, can improve the accessibility test coverage of a page. User journeys can be codified, then replayed, in order to move the browser to each DOM state.

Unit testing / Continuous Integration frameworks can be used to run JavaScript testing libraries to ensure accessibility issues are found at the earliest possible point in a build and to ensure accessibility issues don't creep into the build on an ongoing basis.

References

Electronic and Information Technology Accessibility Standards; Final Rule, 65 Fed. Reg. 246 (December 21, 2000) (Codified at 36 CFR pt 1194)

W3C World Wide Web Consortium (2008), Web Content Accessibility Guidelines 2.0 Level A & Level AA Success Criteria. Retrieved from <http://www.w3.org/TR/200X/REC-WCAG20-20081211/>

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Providing An Agile, Mobile Touch To Section 508 Testing

Larry L. Lewis, Jr.

Management Analyst
Section 508 Office
Department Of Veterans Affairs
Phone: (216) 381-8107
Larry.Lewis3@va.gov

Abstract:

This paper outlines the challenges of applying Section 508 standards against the process of testing mobile content. It proposes a process which satisfies the requirements of Federal Law while providing remediation guidance to developers of mobile content. Practical examples will be offered when this paper is presented.

Introduction:

Over the past 18 years the current Section 508 regulations have been faced with an ever-changing, technology landscape. Since these regulations were introduced in 1998, the mechanisms and methodologies by which Federal employees and those who access electronic content produced by Federal agencies has drastically changed. In the late 90s, most of us were either tethered to desktop PCs largely driven by Microsoft's Windows Operating systems. Others may have had the luxury or inconvenience of accessing heavy laptop systems that most likely did not perform at the same level or speed as their desktop counterparts. Mobile phones were much larger than the devices being used today. These handsets were powered by proprietary operating systems whose primary function was to send and receive phone calls.

Federal Employees as well as those who access content produced by these agencies have many more options for interacting with this content in 2016. Computers have gotten smaller, faster, less expensive, and have provided a seamless transition to tablets and mobile phones. Two additional manufacturers of palatable OS alternatives have made significant inroads into a sector once dominated by a single manufacturer. Today's mobile devices provide every bit as much power as their desktop and laptop counterparts.

Understanding the Problem

These realities present challenges to ensure that content developed for these devices meets the standards set forth within Legislation crafted prior to the inception of touch screen devices and evolving operating systems being brought to market. In order to create solutions to transform these challenges into opportunities, the following three issues require attention:

- From Desktop to Mobile: Visit a website on a desktop PC and then visit the same site on a mobile phone. Without question, the ways that these different browsers behave are striking. Couple development efforts wishing to leverage the capabilities of these mobile devices through the development and deployment of mobile Apps, and the differences in how this dynamic information is accessed is even more profound. Although the differences in accessing information on desktop and mobile platforms are so drastically different, there are three user expectations that transcend these platforms and must be addressed by developers: a logical reading order for content, the ability to navigate through it, and the expectation to interact with it.
- A shift in how information is accessed: Methods of accessing information have shifted from desktop keyboard and mouse usage to continual use of hardware keyboards coupled with an ever-growing dependence on external Bluetooth keyboards, onscreen keyboards, touch gestures, and voice dictation. While it is essential to continue to consider hardware keyboard operability as a “must” for navigating throughout and interacting with information, touch screen gestures along with the use of one’s voice provide added tools to the user’s technology arsenal when accessing, navigating through, and interacting with this information.
- From Third Party to Integrated: The traditional means of access to electronic content has been for third party companies to develop solutions to provide auditory and/or screen magnification access to the operating system of a device and its respective applications. Google and Apple have set a healthy trend whereby the manufacturer of the operating system assumes responsibility for the integration and enhancements of accessibility tools within their operating systems. Microsoft is beginning to adopt this strategy within Windows 10.

Seizing the Opportunity

The US Department of Veteran Affairs 508 Office has been presented with the Privilege of testing mobile content and Apps designed to serve our Nation’s Veterans. As part of this service, along with outward-facing Apps designed for use by Veterans, content is also tested for employees to leverage off mobile know-how to more effectively provide services to Veterans.

In order to address the challenges presented when testing such content in a timely and responsive manner to VA Stakeholders, an agile test process that provides measurable results and a remediation plan for moving forward must be executed. The following process may be applied across Federal Agencies and embraced by the private sector as well.

How It Works

The following process describes how the 508 Office tests mobile content once the content is submitted for testing. VA Stakeholders fill out a submission form that provides a snapshot of the content to be tested—whether it is a mobile website, a “native App”, or a “hybrid App”. The submission process also identifies the respective media types that are used by the developer for a specific mobile website/App. Content may consist of HTML, PDF, multi-media, and other file types used to accomplish the mission of the developer.

- **Selecting A Platform For Testing:** Before beginning the process of testing mobile content in an agile manner, it's essential to have a vehicle which allows for testers to manage the testing process. This utility should:
 1. Allow for all requests to be captured: the online submission form referenced above is completed, submitted, and automatically entered into this utility.
 2. Project is "scoped" for automatic/manual testing.
 3. Results are captured, saved, and shared with relevant stakeholders.
 4. Developers are granted access to reports and tools to remediate documented defects.
 5. Conformance is achieved or documented to stakeholders that it has not been achieved.
 6. Developer best practices are mapped to Section 508 standards and are stored and maintained.

The VA's Section 508 Office uses SSB Bart's "Accessibility Management Platform" as its preferred testing vehicle, for it satisfies all of these requirements: <https://va.ssbartgroup.com/>. This platform is not only utilized by the Section 508 Test Team, but it provides outward facing resources for all VA employees as well as contractors tasked to complete VA-related business. This platform provides a seamless means of accomplishing the following:

- **Scoping:** The scoping process entails gleaning content from each of the media types within the specific content to create a representative set of modules to test with the goal of discovering any accessibility violations that exist within the content. By properly scoping this content, accurate test results are efficiently provided so that possible remediation efforts may begin. Affective "scoping" also ensures that not every nook and cranny of a particular project has to be tested. Once Section 508 conformance is achieved, stakeholders are advised to apply all development best practice used to achieve conformance within the representative modules tested to ensure that all content is deemed Section 508 conformant.
- **Initial Assessment:** Mobile content is tested using a combination of manual and automated testing methodologies. It is imperative that tests are completed within their intended environments to provide accurate results for development teams. It is also essential that when providing an initial assessment that concrete examples with clear guidance are provided to stakeholders to expedite and optimize remediation efforts. For instance, mobile web content should not be tested using a desktop PC, a desktop Browser, and a third party screenreader that is not compatible with the mobile devices on which this content is expected to be accessed. Finally, it is expected that the development team will provide access to the content being tested in a "frozen state" so as to not negatively impact the accuracy of the result of the initial audit.
- **Remediation:** Depending upon the defects, a Mobile project may be tested several times before all violations are remediated. This is attributed to either the developers' lack of knowledge as to how to best conduct manual tests using the mobile operating system's accessibility features. Other times, the agile development of mobile manufacturers may create situations where the manufacturers inadvertently introduce their own accessibility bugs that impact how their access tools behave when their assistive technologies are

enabled. The process embraced by the VA's Section 508 Office strives to either uncover work-arounds for developers to account for these circumstances, or provides documentation that the violation is currently not able to be remediated at the time of testing due to variables introduced by the manufacturer.

- 508 Conformance: When conformance is achieved, all stakeholders are notified and are directed to the final report that validates conformance. This report may be used to satisfy 508 contractual requirements prior to deployment of the content for general use.
- Sustainment: Especial in regard to mobile Apps, contracts are established to optimize and improve upon previous versions of an App. Also, mobile websites that are deemed compliant are sometimes "wrapped" and redeployed as Hybrid mobile Apps on multiple operating systems. It is imperative that Section 508 Conformance be established for particular versions of Apps on the particular operating systems where they are intended to be used. The proper implementation of an appropriate tracking mechanism satisfies this expectation.

Conclusion

The ultimate goal of this testing process is to provide guidance and direction for bringing into conformance as many Section 508 projects as possible offering to the VA's disabled Veterans and employees who serve them the access they so richly deserve and are entitled to by Federal Law. This mature process is constantly being tweaked to meet the needs of evolving mobile content and Operating Systems that are upgraded several times per year. Best practices are often being adjusted, and the Agency's standards are routinely scrutinized to ensure that the VA as an organization is upholding the Mission of Section 508 as its users' needs evolve, mainstream manufacturers provide enhancements to their operating systems, and developers strive to create content that honors the requirements of Section 508 Regulations. This approach to testing agile content is scalable and can be implemented across Federal Agencies and companies within the private sector wishing to provide the most optimal, accessible user experience for all user wishing to access their mobile content.

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Improving Testing for Survey and Exam Accessibility

Kare Romanski

AMAC/GA Tech
512 Means Street Atlanta, GA 30318, USA
kromanski6@gatech.edu

Abstract

The aim of this paper is to help professionals save time and effort while producing accessible, meaningful surveys and exams. Accessibility tests require considerable planning well before the actual accessibility and or usability testing begins. There are key stages for testing the accessibility of higher education and corporate exam or survey deployments.

Introduction

This paper was inspired by my professional experiences and learning as a seasoned avenger of web and mobile accessibility experiences in corporate, government and higher education sectors.

We'll consider the elements that go into writing a survey or exam before the first question is crafted with accessibility in mind at every phase. Next, we'll discuss some best practices for writing accessibility test items, including decreasing the cognitive load and avoiding bias and stereotypes. After that, we'll discuss pulling items together into an accessible exam or survey form, including developing instructions and testing compliance.

Then, we'll look at evaluating a test's performance to improve the quality and assess reliability, in addition to accessibility.

Planning the Survey or Exam

Reliability and validity should be established right from the start. An assessment's results are considered reliable if they are dependable, repeatable and consistent. The assessment is deemed to be accessible and valid only if is perceivable, understandable, operable and measures the specific knowledge and skills that it is meant to measure regardless of abilities or devices.

Testing Content for Accessibility

Provide Users with Guidance

- Clearly state what the survey or exam is about.
- Provide instructions including how many questions there are.
- Provide expected time for completion of the task or use an accessible progress indicator.
- Allow people to save and return to the survey, especially if it's long.

Question Types to Avoid

Likert-Scale Questions

The Likert Scale is a five (or seven) point scale used to express how much the user agrees or disagrees with a particular statement. Screen readers interpret the Likert scale as a table, with the answer option labels as one row and the buttons as a separate row. This can cause confusion for respondents, so it's best to use radio buttons, set to display horizontally, for all single selection type questions.

JavaScript-based Questions

JavaScript-based questions don't work well with screen readers. We'll review six components which use JavaScript.

Hidden Questions

Avoid hiding questions with logic as they are often hidden using JavaScript and screen readers have trouble deciphering live questions from hidden ones.

Problems with Interactive Question Types

Ensure more interactive questions types that advanced online survey tools and form builders have, are often not keyboard accessible, including image choice, ranking, and drag and drop question types.

Rankings

When asking readers to rank items, ensure words rather than numbers are used as the scale. Too often, people need to repeatedly refer to the legend.

Validation/Warning Messages

Validation and associated warning messages can be problematic for screen readers as they are often not visible. Ensure warning messages are updated then test thoroughly to ensure that they are visible to the screen reader.

Language

Ensure clear and simple language is used. Sentences should be short and jargon-free.

Specify the intended document language and specify any inline languages.

Fonts

Font-Families

Ensure that consistent and familiar San-Serif font families (i.e.; “Arial, Helvetica, sans-serif”) are used that are supported across platforms.

Font-Sizes

Ensure fonts that are based on a fixed pixel size was avoided. It is best to use relative font to better meet the needs of users that are low vision or with cognitive challenges.

Color & Contrast

Ensure color alone is not used to convey information alone. Add another method, in addition to color, to convey information or emphasize text.

Background Colors

Ensure support for higher color contrast to better suit low contrast or color blindness in surveys and forms as a standard.

Form Design

Ensure forms adhere to HCI principles and follow WCAG 2.0, and forms are accessible and easier to use. Not following expected practices will result in problems and an increase in survey abandonment.

Provide Instructions and Visual Cues

Make sure that the order in which form elements are accessed is logical and easy and ensure clear instructions about what information is provided.

Ensure a cue or instruction that the field is required was provided and not represented by color alone so that those who are not using an assisted device may also be alerted is a good practice.

Also provide instruction if you have fields that need to be entered in a certain format.

Form Labels

Improve the layout of your online forms by placing form labels near the associated input field depending on the format of the input field.

Associate Form Labels with Controls

Text labels should generally describe the function of each form control. Ensure the label's position for a textbox field is different than that of checkboxes or radios buttons.

Text Fields

Ensure text precedes the input field text fields, either directly above or to the left.

Radio Buttons and Checkboxes

Ensure the proper element is used based on the response options that can be selected. If only one, use radio buttons. If more than one, use checkboxes.

Group Related Fields

Ensure there is a logical order for reading and navigating the form and like fields were grouped together.

Descriptive text can be associated to a group of form controls using `<fieldset>` and `<legend>`. The `<fieldset>` identifies the entire grouping and `<legend>` identifies the grouping's descriptive text and is read to screen reader users when the grouping is navigated to.

Groupings of form controls, typically groups of related checkboxes and radio buttons, sometimes require a higher level description. You can expect problems if developers stray from the norm.

Messaging

Ensure the form design is simple and clean and users understand what information they need to complete.

Keyboard Accessibility

Ensure that all page functionality is available using only the keyboard. One of the most crucial requirements for accessibility is the ability to navigate a form with only the use of a keyboard.

Visible Focus Indicators

Ensure highly visible focus indicators are consistently provided for all focusable elements. Another design method is to provide a visual change to a button to indicate that that button currently has keyboard focus.

Activating Dropdown Navigation Menus

Ensure JavaScript `onChange` events are avoided navigation, such as non-standard dropdowns menus. Ensure when using a keyboard, the first item in the drop-down-menu can't be activated when users attempt to tab through the list options.

If you are going to use a drop-down-list for navigation, provide a button that allows the user to activate the selection.

Difficulties with Tables

Tables can give online forms or surveys a nice grid layout but if the label and the form element are separated, it becomes hard for screen readers to interpret the field.

Screen readers decipher the table content in the order it appears in the source code. If the text and the form element that the text relates to are split, you will cause readability issues.

Use Alternative Text for Images

Inspect alternative text is present and appropriate or an empty alt attribute was provided for decorative images when you upload your image so that screen readers can present users with a text description in place of the image.

Button Messages

Be sure to include text on your buttons that indicates the purpose of the button. If you do not include text on your button, your users will be confused about its intent regardless if they are blind or sighted.

Click Here Links

Ensure descriptive text is used for links. Links that say “click here” are ambiguous for all users, let alone those who are using screen readers.

Form Validation

Form validation ensures that the user provided necessary and properly formatted information for a successful completion.

The form should be usable regardless of whether scripting is enabled or not. Server-side scripting reliably solves this problem by validating the fields after the form has been submitted. Often forms utilize both client-side and server-side scripting so that validation and error correction can be provided.

User Accessible Forms

All users should be able to interact with your site.

In conclusion, despite the technological advances aimed at making the Web easy to use, Web-based surveys and exams may pose significant barriers to people with disabilities. However, a very important to realize the most common errors are not necessarily the worst errors.

Three recommendations can help to quickly and easily improve the accessibility of a website: Label correctly the form controls with meaningful and descriptive text. This makes it clear to the user what information they should be providing. If a form has many controls and it is too complex, group the controls with fieldset and legend or aria role group. Check the tab order in the form.

Add proper alternative text to images. The text provided as the text alternative should represent the content and function of the image.

Test web pages with a keyboard. Make each page and focusable element navigable by keyboard alone using the tab key.

References

Accessible HTML/XHTML Forms: Beginner Level: Web Standards Project. (2016, Jun 14). Retrieved from <http://www.webstandards.org/learn/tutorials/accessible-forms/beginner/>

BBC - My Web My Way - Change text and background colours: BBC. (2015, July 31). Retrieved from http://www.bbc.co.uk/accessibility/guides/change_colours/

Electronic Accessibility: Guidelines for accessible surveys and forms: UCOP. (2016, Jun 15). Retrieved from <http://www.ucop.edu/electronic-accessibility/web-developers/tools-and-testing/guidelines-accessible-surveys-forms.html>

Shrock, Sharon and Coscarelli, William. United States. 2nd Edition. Publisher: Pfeiffer, 2007. *Criterion-Referenced Test Development Technical and Legal Guidelines for Corporate Training*. San Francisco, California, United States, 2007. Print.”

SurveyGizmo: How to Create Accessible Online Forms: SurveyGizmo. (2015, Nov 17). Retrieved from <http://www.surveygizmo.com/survey-blog/creating-accessible-online-forms-everyone-benefits-from-a-well-designed-web-form/>

WebAIM: Creating Accessible Forms - General Form Accessibility: WebAIM. (2016, July 1). Retrieved from <http://webaim.org/techniques/forms/#intro>

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Social Media and Accessibility in Employment

Gian Wild

AccessibilityOz
New York, New York, USA
gian@accessibilityoz.com

Abstract

Social media is an integral part of the web, and it is becoming even more important when it comes to employment prospects. People with disabilities already find it difficult to access employment and any web sites that are inaccessible to them makes the issue that much harder. LinkedIn is one social network that is becoming essential for employees in the twenty-first century. A subset of requirements from the W3C Web Content Accessibility Guidelines were used to test how accessible LinkedIn is to people with disabilities. A large number of issues were found, indicating that LinkedIn is not an accessible social network and is unlikely to provide the same functionality to people with disabilities using the system as those provided to the general public.

Introduction

Social media is an incredibly important tool in modern society. There are five main reasons people access social media: personal (such as sharing photos on Instagram), work (such as finding jobs on LinkedIn), entertainment (such as following celebrities on Twitter), provision of goods and services (such as responding to user complaints on Facebook) and education (such as watching instructional videos on YouTube) [1]. It is not just the young who access social media, with close to 30% of people over the age of 65 interacting on social networking sites, and 50% of people aged 50 – 64 [2].

Social media and employment

Social media is becoming an essential part of negotiating the current working environment. Of all US adults who use social media, 35% have used social media to look for or research a job, 34% have used social media to inform friends of a job in their current company and 21% have applied for a job found through social media contacts [3]. Thirteen percent of social media users state that their existing social media profile has helped with finding a job [3].

As the percentage of recruiters who use LinkedIn to review job candidates is now 95% [4], this social network is integral to one's employment prospects. With over 414 million users [5] (including over 120 million registered members in the US [6]), two new users every second [5] and 3 million active job listings at any one time [7], it is a formidable network.

Thus the accessibility compliance of these social networks is of paramount importance to people with disabilities. People with disabilities are already discriminated against in the workforce, as the many instances of litigation under the Americans with Disabilities Act attests to [8]. Although the employment rate of people without disabilities has increased slightly over the last 25 years – from 76% to 78%, the employment rate of people with disabilities has almost halved – from 29% to 16%, despite no significant change in the percentage of people with disabilities over the twenty-five years [9]. With the employment participation rate of people without disabilities close to three times that of people with disabilities [10], any barrier to employment must be rectified.

When it comes to LinkedIn, one's success is greatly improved by interacting with the network. For example, adding a professional photo to one's profile means that one is fourteen times more likely to be found in LinkedIn [11]. Adding skills to one's profile increases one's views thirteen times [12]. The number of comments on a LinkedIn post doubles if that post contains an image [13]. Therefore, LinkedIn is not just about creating a profile, but updating it as required. It is therefore essential that these features are fully accessible.

Methodology for testing the accessibility compliance of social networks

Social media web sites and mobile apps were tested against the W3C Web Content Accessibility Guidelines on a desktop computer, an iPhone and an Android phone. The Homepage, Create an account, Login, Profile page, and Add a post pages were tested.

Testing against the W3C Web Content Accessibility Guidelines

A selection of success criteria from the W3C Web Content Accessibility Guidelines were used to assess the accessibility compliance of the various social media web sites. These success criteria are representative of the most serious accessibility issues that people with disabilities encounter when accessing a web site. Success criterion were tested manually on a Firefox browser (Version 44) on Windows 10; using Chris Pederick's Firefox Web Developer Toolbar [14]; the Paciello Group's Colour Contrast Analyser [15]; and the AccessibilityOz Firefox bookmarklet.

Success Criterion 1.1.1: Non-text content (Level A)

This success criterion requires that text alternatives are provided for all non-text content, such as images, for assistive technologies to interpret. Without text alternatives ("ALT attributes") all images would be unavailable to screen reader users. Missing text alternatives to form features, such as image submit buttons, means that assistive technology users, such as screen reader users and speech recognition users (for people with vision impairments and physical disabilities respectively), will be unable to use a form at all.

Images and other non-text content in social media sites were tested for the presence and accuracy of text alternatives. Social media sites were also tested for the ability for a user to add text alternatives to user-generated content.

Success Criterion 1.3.1: Info and Relationships (Level A)

This success criterion requires that important information about the content be coded so that it can be interpreted by assistive technologies. For example, headings can be coded in a particular way so that screen reader users can access a list of headings in the page, and therefore access an overview of the content of the page. Social media sites were tested for the presence and accuracy of headings and for the presence of appropriately coded form elements.

Success Criterion 1.4.3: Contrast (minimum) (Level AA)

This success criterion requires that the colour contrast between foreground text and the background meets certain requirements. People with failing eyesight and people who are colorblind have difficulty reading content with low colour contrast. Approximately 8% of men are colorblind [16], therefore it is a common ailment (although it is not defined as a disability). Due to the high prevalence of colorblindness amongst the general population this success criterion was deemed integral to the accessibility compliance of social media web sites, despite the fact that it is in the Level AA (medium) category, not the minimum category. Items that were deemed mandatory to understanding and interacting with the social media site were tested for adequate colour contrast.

Success Criterion 2.1.1: Keyboard (Level A)

This success criterion requires that all content and functionality in a web site be accessible via the keyboard. A number of different physical disabilities restrict a user's ability to use a mouse, and these people often rely on a keyboard to access a site. Keyboards can also be implemented with mobile devices, for people who have difficulty using the touchscreen feature.

Success Criterion 2.1.2: No Keyboard Trap (Level A)

This success criterion requires that any component that can be entered via a keyboard can also be exited via a keyboard. There are some instances where features, such as video players, trap the keyboard focus and the user cannot escape the feature. In order to continue using the site the user must close the browser and begin again. Success Criterion 2.1.2 is one of the four “non-interference” success criteria in WCAG2 [17]. These four success criteria must be met across an entire site, even if part of the site is deemed to be inaccessible. Failing one of these four success criteria interferes so significantly with some user's interaction with the site that it is deemed a critical failure.

Success Criterion 2.4.1: Bypass Blocks (Level A)

This success criterion requires that users that can only access content sequentially can jump over repeated content such as navigation straight to the body of the page. One of the most common methods of achieving this is to provide “skip links” which provide an anchor link to the content of the page. According to WCAG2, skip links must be the first focusable link on a page [21]. For example, this allows screen reader users to jump past the navigation, which, if there were no skip links, would be repeated on every page that they visit.

Success Criterion 2.4.3: Focus Order (Level A)

This success criterion requires that the order of content is meaningful to the user. There are three different content orders: visual order of content on the page; source order of the code; and the order in which items receive keyboard focus

It is important that these three orders are the same. Often people with disabilities will have access to more than one content order, and if there are differences between these content orders it can cause serious confusion to users. For example, a keyboard-only user of a site will have access to the visual order of the content of the page, as well as the order in which items receive keyboard focus. As another example, people with cognitive disabilities who use screen readers to assist in reading content will have access to the visual order of the content in the page and the source order of the code (which is used by the screen reader).

This success criterion was tested in conjunction with Success Criterion 2.1.1: Keyboard, by manually reviewing the visual order of the content on the page and by displaying the content with style sheets disabled using an internal AccessibilityOz FireFox bookmarklet.

Success Criterion 2.4.7: Focus Visible (Level AA)

This success criterion requires that an element that has keyboard focus is visually indicated to the user (“keyboard focus indicator”). If this is the case, the user can easily follow keyboard movement through the page, and activate appropriate items. Where items do not have a highly visible keyboard focus indicator the user will not know where on the page their focus is located. This makes a site incredibly difficult, if not impossible to use. As this is a feature that is essential to keyboard-only users in interacting with a site it was deemed integral to the accessibility compliance of social media web sites, despite the fact that it is in the Level AA (medium) category, not the minimum category.

The accessibility compliance of social media

Results of testing against the W3C Web Content Accessibility Guidelines

Success Criterion 1.1.1: Non-text content (Level A)

Surprisingly the LinkedIn site does not have many images, however they all include accurate ALT attributes. Unfortunately, there is no way for a user, when posting a status with an image, to add an ALT attribute to that image. As a result, all images added to a status or a post have empty ALT attributes.

Success Criterion 1.3.1: Info and Relationships (Level A)

The forms in LinkedIn have been coded appropriately as specific form elements. This should make it straightforward for screen reader users to access the site.

Headings have been coded in the site, however the hierarchy of the headings are incorrect. In the example below the text ‘Platinum Asset Management Limited shared:’ is a heading 3. The article title ‘Sugar addiction – breaking the cycle’ is a heading 4. However, the names of the commenters on the article (‘Alan Wallace’, ‘Brett Elliott’) are also coded as heading 4 (see Figure 1). These commenter names are sub-headings to the article name and should be coded as a heading 5.

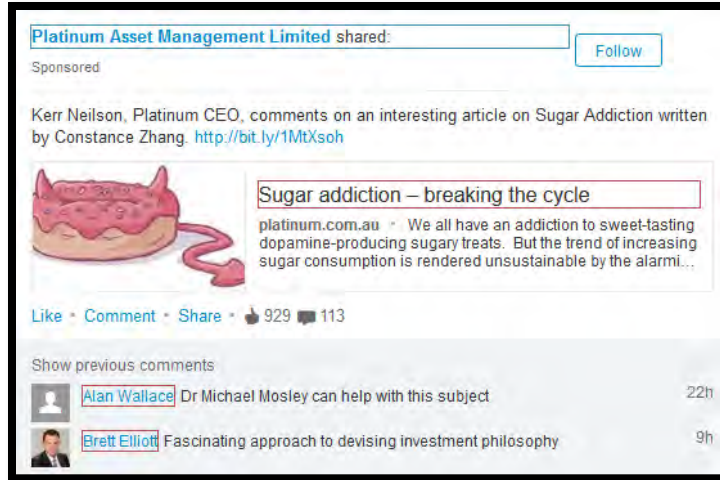


Figure 1 - Heading structure (headings outlined) in the main profile page of LinkedIn

Success Criterion 1.4.3: Contrast (minimum) (Level AA)

Colour contrast is problematic, with some information presented as medium-grey text on a light-grey background on a mobile device. Whether a person is a first degree contact or a second-degree contact is provided as medium-grey text on a light-grey background and fails WCAG2 colour contrast requirements (see Figure 2).



Figure 2 - Colour contrast failures in the login feature

When entering an incorrect password into the login box, the error text is red on a grey background (see Figure 3) and this also fails WCAG2 colour contrast requirements.

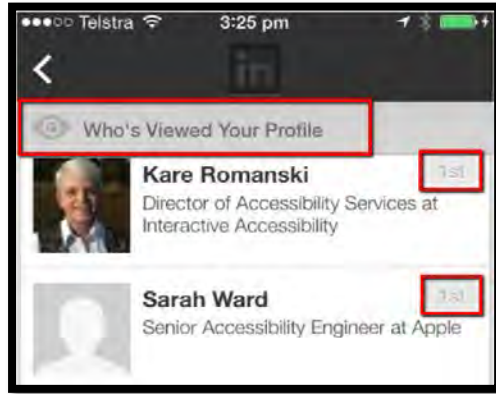


Figure 3 - Colour contrast failures in the iPhone LinkedIn mobile app

Success Criterion 2.1.1: Keyboard (Level A)

When creating an account some important features are not keyboard accessible, including the ability to skip importing contacts (see Figure 4), and sending another email if the original email was not received (see Figure 5).

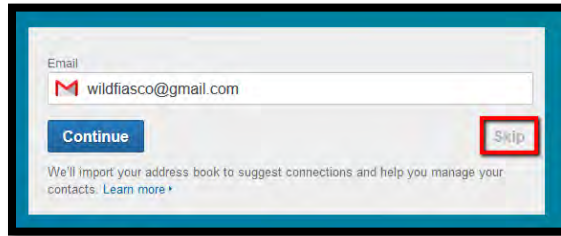


Figure 4 - The 'Skip' importing contacts link can only be accessed via a mouse

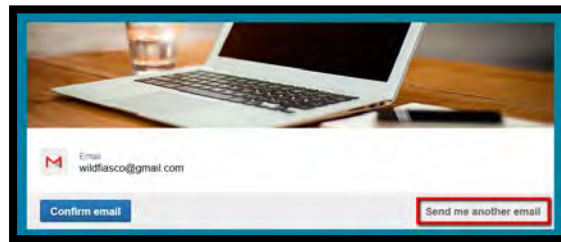


Figure 5 - The 'Send me another email' option can only be accessed by the mouse

Some features are not keyboard accessible, such as saving, publishing or formatting functions when adding a post.

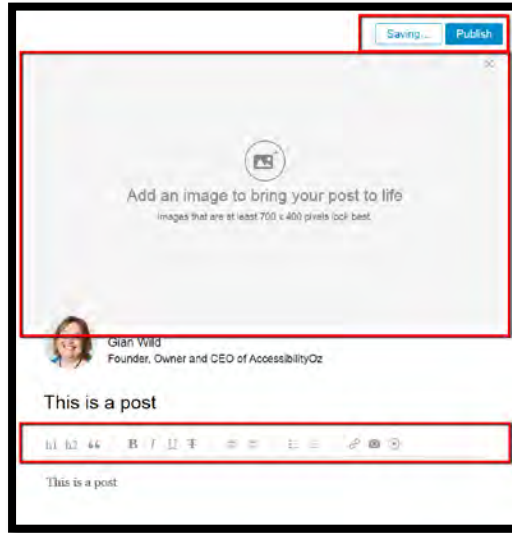


Figure 6 - Many options in the add post feature can only be accessed by the mouse

Success Criterion 2.1.2: No Keyboard Trap (Level A)

When first accessing the site as a new user a popup appears over the top-left navigation with information about messaging (see Figure 7). This popup cannot be closed by the keyboard. In addition, a user tabbing through the navigation will not be able to see the items currently in focus as they are overlapped by the popup. This is referred to as a ‘reverse keyboard trap’ [22] – where content cannot be closed with the keyboard and this content overlaps important information. This can only be closed by leaving LinkedIn and logging in again – on subsequent logins this popup does not appear.

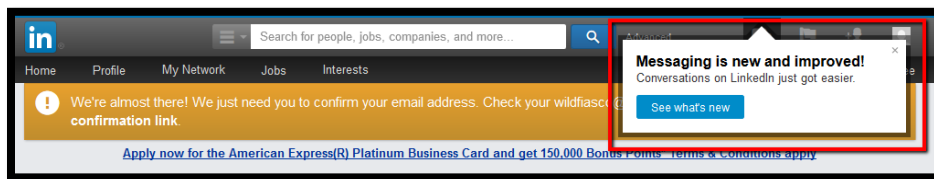


Figure 7 - The messaging popup overlaps important navigation items in the top-left and can only be closed using the mouse

Success Criterion 2.4.1: Bypass Blocks

LinkedIn contains quite detailed skip links. There is a popup available on keyboard focus that allows users to jump to sections such as profile activity, update status, network updates and search (see Figure 8).

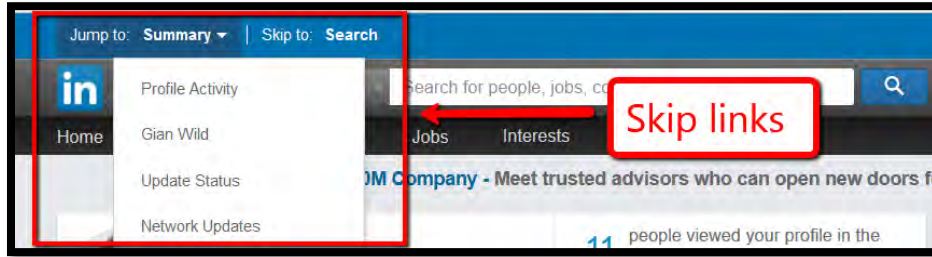


Figure 8 - LinkedIn includes comprehensive skip links that appear on keyboard focus

Success Criterion 2.4.3: Focus Order (Level A)

In the sign up process the source and keyboard focus order do not match the visual order of content (see Figure 9). The first item is the field, however the second item that receives keyboard focus is a 'Learn more' link which appears below the 'Continue' button. The 'Continue' button subsequently receives keyboard focus.

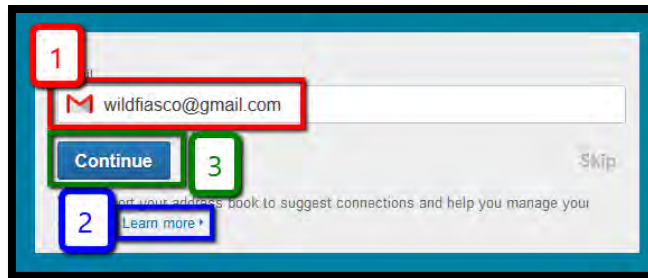


Figure 9 - The source order of content does not match the visual order in the sign up process

Success Criterion 2.4.7: Focus Visible (Level AA)

LinkedIn often has a highly visible keyboard focus indicator so keyboard-only users know where they are positioned on the page. When looking at pending invitations in the menu the focus indicator is a bright blue outline (see Figure 10).

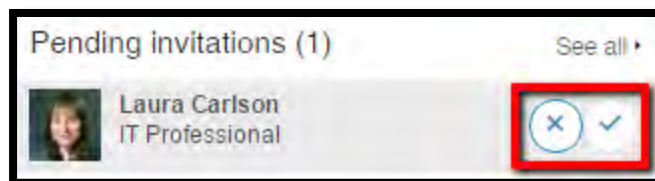


Figure 10 - Some items, such as rejecting an invitation have a highly visible keyboard focus indicator

Unfortunately, some items that have a keyboard focus indicator do not meet colour contrast requirements. In the dropdown for the menu, when an item receives keyboard focus it changes to a deep-blue colour (see Figure 11). This colour, against the dark-grey background does not meet WCAG2 colour contrast requirements.

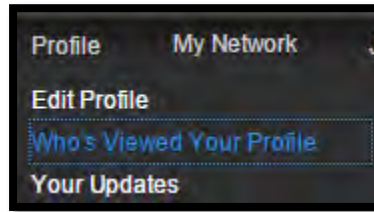


Figure 11 - The keyboard focus indicator is a deep-blue against a dark-grey background and does not meet colour contrast requirements

Conclusions

In conclusion LinkedIn still has many accessibility problems that are likely to cause problems to different groups of people with disabilities. As a result, these groups are less likely to be able to find relevant employment, contributing to the under-employment of people with disabilities.

Why is social media inaccessible?

The main reason why social media is not accessible is that social networking sites and apps are almost continually refreshed. Facebook sometimes changes twice a day [23]. This, coupled with a lack of a formal testing process, means that what may be accessible today may be literally gone tomorrow. Although there have been some improvements in the accessibility of social networks over the last year; namely the removal of a CAPTCHA in the signup process for LinkedIn, any accessibility features can be instantaneously lost as the site or mobile app is updated.

References

1. Kietzmann, J.H., Hermkens, K., McCarthy, I.P., Silvestre, B.S.: Social Media? Get Serious! Understanding the Functional Building Blocks of Social Media. *Business Horizons*, 54(3), 241-251 (2011)
2. Perrin, A.: Social Media Usage: 2005-2015, <http://www.pewinternet.org/2015/10/08/social-networking-usage-2005-2015/> (2015)
3. Smith, A.: Searching for Work in the Digital Era, <http://www.pewinternet.org/2015/11/19/searching-for-work-in-the-digital-era/> (2015)
4. Pratt, S.: How Recruiters and Job Seekers Use Social Media in 2015 (Infographic), <http://www.socialtalent.co/blog/how-recruiters-and-job-seekers-use-social-media-in-2015> (2015)
5. Sordello, S.: LinkedIn's Q4 2015 Earnings, <http://blog.linkedin.com/2016/02/04/linkedins-q4-2015-earnings/> (2016)

6. LinkedIn, <https://press.linkedin.com/about-linkedin>
7. LinkedIn Corporate Communications Team: LinkedIn Announces Second Quarter 2015 Results, <https://press.linkedin.com/site-resources/news-releases/2015/linkedin-announces-second-quarter-2015-results> (2015)
8. United States Department of Justice Civil Rights Division: Information and Technical Assistance on the Americans with Disabilities Act, http://www.ada.gov/enforce_current.htm#TitleI (2016)
9. Brucker, D.L., Houtenville, A.J.: People with Disabilities in the United States. Archives of Physical Medicine and Rehabilitation, 96(5), 771-774 (2015)
10. Bureau of Labor Statistics: Employment Situation Summary: Employment Status of the Civilian Population by Sex, Age, and Disability Status, not Seasonally Adjusted. <http://www.bls.gov/news.release/empsit.t06.htm> (2016)
11. Rao, N.: Celebrating Leadership & LinkedIn Power Profiles 2015 in India, <http://blog.linkedin.com/2015/08/13/celebrating-leadership-linkedin-power-profiles-2015-in-india/> (2015)
12. Bastian, M.: Your Skills Are Your Competitive Edge on LinkedIn [INFOGRAPHIC], <http://blog.linkedin.com/2014/10/16/your-skills-are-your-competitive-edge-on-linkedin/> (2014).
13. Seiter, C.: Master The LinkedIn Company Page: 12 New Data-Backed Tips To Max Out Yours, <https://blog.bufferapp.com/linkedin-company-pages> (2015)
14. Pederick, C.: FireFox Web Developer Toolbar, <http://chrispederick.com/work/web-developer/history/firefox/> (2013)
15. The Paciello Group: Contrast Analyser, <https://www.paciellogroup.com/resources/contrastanalyser/> (2016)
16. Color Blind Awareness, <http://www.colourblindawareness.org/>
17. World Wide Web Consortium: Understanding WCAG 2.0: Understanding Conformance, <https://www.w3.org/TR/UNDERSTANDING-WCAG20/conformance.html> (2008)
18. World Wide Web Consortium Understanding WCAG 2.0: Bypass Blocks: Understanding SC 2.4.1, <https://www.w3.org/TR/UNDERSTANDING-WCAG20/navigation-mechanisms-skip.html> (2008)
19. Wild, G.: Mobile Accessibility Issues - Part One: Hover, <https://www.linkedin.com/pulse/mobile-accessibility-issues-part-one-hover-gian-wild> (2015)
20. Rossi, C.: Ship Early and Ship Twice as Often, <https://www.facebook.com/notes/facebook-engineering/ship-early-and-ship-twice-as-often/10150985860363920> (2012)

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

A11y Facts: An Accessible Playing Field for Instructional Materials

Robert Martinengo

CAMI Project Co-Director
AMAC Accessibility Solutions and Research at Georgia Institute of Technology
512 Means Street, Suite 250
Atlanta, GA 30318, USA
robert.martinengo@amac.gatech.edu

Elaine Ober

Head of Accessibility Advocacy, Pearson Global Product
Pearson
501 Boylston Street
Boston, MA 02116
elaine.ober@pearson.com

Abstract

Instructional materials for higher education are now available in a wide variety of formats. While printed textbooks still dominate the marketplace, products are increasingly digital. Just because a product is digital, however, does not mean students with disabilities will benefit. Even digital products can be made with varying levels of accessibility, and a student purchasing a digital product is not assured of an accessible experience. “Leveling the playing field” is often used to describe the potential for technology to eliminate barriers to information for these students, but making sure all instructional materials are accessible “off the shelf” to all students requires new strategies that include accounting for accessibility at each step of product development. Testing and resulting compliance reporting may help, but to level the playing field requires disclosure of accessibility information.

Today all students expect a choice in how they obtain study materials. Unfortunately, there isn’t a litmus test for accessibility; products may be more or less accessible, but each student’s needs define what is accessible for them. A student with disabilities may have choices, but to make an informed choice they require a transparency of features and access to information about how a product can meet their needs. Accordingly, the Center for Accessible Materials Innovation is proposing the development of a new “A11y Facts Portal” which provides standardized accessibility information for instructional materials. This portal would provide information on the accessibility of content products, reading systems, and assistive technology.

Following an Example: The Nutrition Label Model

The U.S. Nutrition Labeling and Education Act of 1990 provides us with a model of the kind of transformation standardized labeling can perform in the marketplace. While the Act did not allow food manufacturers to label things as healthy or unhealthy, it did define a set of “facts” (percentage composition) and definitions (e.g., “high fiber”) that consumers could use to make their own judgments about the desirability of a given food product for their needs.

The Nutrition Label does not, and cannot, tell a person whether a product is “healthy” for them. Sodium content is an example: a person with high blood pressure must eschew sodium content, but a person with low blood pressure needs to increase sodium intake.

The Center for Accessible Materials Innovation is commencing the development of a conceptual label called the A11y Facts Label (for “Accessibility Facts”) for higher education instructional materials. As the Nutrition Label does not define “healthy” foods, the A11y Facts label will not define a metric for what is accessible and what is not. Instead the label will provide a means to allow facts to be known and rely upon published standards for definitions of characteristics.

The usefulness of the A11y Facts Label is based largely on the concept that students know the characteristics of products that will be amenable to their consumption. While the label cannot attest to whether a product is accessible to students with specific disabilities, it can list characteristics (such as alternative text provided for images) that will let consumers know how likely it is to meet their needs.

Today’s student with a disability will frequently first go to the campus Disability Support Services (DSS) office. The DSS office may attempt to search the market for an alternative, or more likely will go to the publisher for information or files, or assess the need to create an alternative accessible format. All of this takes time, often putting the student at a disadvantage as the semester continues on and the student falls behind. If the student and the DSS had instant access to information on accessible alternatives, this whole process could be more expedient.

In 2011, The Advisory Commission on Accessible Instructional Materials in Postsecondary Education for Students with Disabilities (the AIM Commission) was authorized under the Higher Education Opportunity Act of 2008 (HEOA) to address and seek remedies for the challenges encountered by students with print disabilities enrolled in postsecondary institutions. The conclusion of the AIM Commission was that with increasing alternatives (especially digital) in instructional materials, that the marketplace would evolve to satisfy the needs of students with disabilities. While there has been some progress in the marketplace toward this end, the consumer with a disability may not be driving the change in offerings and, in fact, the consumer has no place to go to understand the accessibility features of the available offerings, and no place to give voice to the market for his/her needs.

The proposed A11y Facts Label would make important information about educational content available to people with disabilities. Current practices support multiple mechanisms (described below) for content producers, publishers, distributors and alternative media producers (organizations which convert digital or paper books to accessible formats) to make accessibility information available. Each mechanism has its own set of standards and metadata requirements.

Currently there is no mechanism that displays this information to consumers even when it is provided. The A11y Facts Label will allow this information to be mapped to one consistent set of definitions across media and format types, and displayed for the consumer in a user-friendly manner.

The Nutrition Facts Label is a level playing field where food producers provide important information about their products to consumers. The A11y Facts Label takes a similar approach to make important information about digital content available to people with disabilities. In contrast to Nutrition Facts, A11y Facts is an online system where the label will be displayed on a screen or mobile device. To assemble the “labels” in one place, the A11y Facts Portal will be developed to house the information, allow search and discovery, and permit market feedback from the consumer.

Who Defines Accessibility?

Unlike the Nutrition Label which reports strictly measurable characteristics, accessibility features lend themselves to more subjective evaluation. A product may indeed have alternative text (alt-text) provided on every image, but whether the alt-text is well written, relevant and useful is a more subjective measure, likely to differ from user to user. Likewise, a statistical graphic map may be converted into a table of data, but does a table of minute details have the same instructional equivalence of a visual presentation of colors and shapes?

The values on the Nutrition Label are not subject to consumer evaluation, but many accessibility factors are. Therefore, we feel that the A11y Facts Portal needs to provide mechanisms which facilitate feedback to the product’s publisher or producer as to the usefulness of product features. Access to facts about the accessibility of instructional materials may encourage more open discussion by consumers. Students with disabilities may enlist existing advocacy groups to amplify their voice to publishers and other vendors, or they may form new organizations that exploit the new potential of social media.

Today, all students want and expect a choice in how they obtain their study materials and how and when they use them. Students with disabilities are no different. The challenge is to ensure the new playing field for instructional materials is not only accessible to students with disabilities, but also empowers them with a voice in the process.

The keys to a level playing field are transparency and equal access to information. Without a complete understanding of what they are buying and what it is capable of, students with disabilities remain at a disadvantage, even as more accessible products come on the market. Students want to know the full range of options available to them. A student may have a choice between accessible non-digital products (e.g., large print, Braille, audiobooks) and multiple digital formats (e.g., an EPUB3 book, an online learning product, etc.). Today, it is difficult to find a singular place that presents all these alternatives to the consumer for a given book, especially since these alternatives may include versions from commercial, limited-distribution, and non-commercial sources.

An additional factor to consider is the reading system or accessibility technology used for consumption. Accessibility standards have been developed for web-based products (WCAG 2.0)

and for EPUB3 ebooks, but what makes a product accessible remains subjective to the consumer and often involves the appropriate combination of the product and an assistive technology. An EPUB3 product that meets the standards for accessing text via a screen reader may still be inaccessible to a dyslexic student when read on a reading system that does not highlight words and sentences as they are spoken aloud. The A11y Facts Portal is intended to be a place where the accessibility of the same EPUB3 file on two different reading systems can be compared.

Challenges

Depending upon the kind of product and the producer of the product, there is more than one way to communicate accessibility information. Currently, the most common mechanism for publisher reporting of features is an XML data feed called ONIX, a standard developed and supported by Editeur. ONIX is used by publishers to communicate product information to distributors and online stores that display product information in their systems. ONIX delivers extremely detailed information about published products. The current ONIX standard, ONIX 3, has been extensively enhanced to transmit information about accessibility. While ONIX 3 is the current standard, only approximately 35% of publishers today are distributing information using ONIX 3, with 65% still supporting ONIX 2.1. There are many reasons for this reluctance to change, including technical effort involved but as mentioned before, there is also a motivational resistance to provide additional information in the new format when there is no venue that will display it.

Publishers and producers of alternative media produce many formats, which can include Braille, Doc files, PDFs, docbook, EPUB, complex digital learning products, and others. Alternative media producers do not generally create ONIX files because they are not generating commercial products. These products can be described by metadata defined in Schema.org's "A11y metadata." Additionally, the most accessible format to date, EPUB3, can imbed this metadata in their packages. Digital resources that comply with IMS Global Learning Consortium's digital resource standards can also imbed "Access for All (AfA)" metadata in their packages. Taken together, ONIX 2.1, ONIX 3, Schema.org, EPUB3, and AfA provide five different reporting mechanisms, each with different (but often similar or even overlapping) metadata.

Students with disabilities are seeking solutions that work for them, and when alternative media producers are included in the mix, there may be a wide range of alternatives available to them. Depending upon the product and the publisher, there may be many versions to consider even on the market. A loose leaf or spiral-bound version of a printed book, for example, may address the needs of certain people with mobility issues. A large print version may be the preferred solution for a low-vision user, and while publishers may sell EPUB3 or web-based online learning products, a student with a disability may still need a PDF because it works best with the assistive technology that they use. Unfortunately, this kind of accessibility information is not universally available through the reporting mechanisms listed above. A publisher-provided large print version may be described in ONIX, a remediated PDF may be developed by an alternative media provider and described on their database, an EPUB3 file from the publisher may contain Schema.org's A11y metadata, and an equivalent online learning package from the publisher may contain Access for All standard metadata. Each of these metadata schemas is different, and each

may have a different set of metadata characteristics as appropriate to their format and reporting mechanisms.

A11y Facts Products

While digital products are often considered the most amenable to servicing the needs of people with and without disabilities, there are still other product alternatives that work best for some people with disabilities. These alternatives include Braille, recorded sound, tactile graphics and large print, which are available both commercially and through alternative media producers. The A11y Facts database is intended to accommodate both digital and physical formats that may meet the requirements for people with disabilities. Likewise, the intent is for A11y Facts to be a place to communicate accessibility information both for commercial and alternative media products.

Digital products are always consumed via a platform, be it a reading system or a browser, either alone or in combination with a screen reader or other assistive technology. Many digital products are bound by DRM to a specific reading system, which is an important component of the accessibility profile. Therefore, in addition to content products, A11y Facts needs to be the one-stop place where a consumer can get information on the accessibility characteristics of reading systems in general, assistive technology, and the combination of a content product with a reading system.

Source Data for Content Product Labels

As outlined previously, ONIX 2, ONIX 3, IMS Access for All and EPUB3, all have mechanisms for conveying information on accessibility features and ability to cooperate with accessibility technology. However, currently there is no place for consumers to view this information. This results in a chicken-and-egg scenario where producers do not supply this information because no one can view it. It is our hope that the A11y Facts Portal will help kick-start the sharing of this information by making it visible to consumers. It is also our hope that, much as has happened in the nutrition industry, making this information visible will inspire competitive pressure to make products accessible and their features visible. We have seen in the food industry that a requirement to disclose trans-fats has resulted in a drive to reduce trans-fats in prepared food products. We hope that similar competitive pressures in higher educational instructional materials will help the AIM Commission's vision of a "market-driven model" become a reality.

Because the objective of the A11y Facts system is to provide broad coverage for accessibility data, we feel we should support the all most commonly available reporting mechanisms. The standards, while similar, are not identical and often use technical terms that might not be meaningful to a consumer. A11y Facts uses its own metadata definition which is consumer-driven. It builds on the ONIX standard commonly used to convey information to retailers but also extends the model to include accessibility features described in EPUB3, schema.org and Access for All. The A11y Facts label gathers this accessibility information in one place and makes it available to consumers in direct, consistent, easy to understand language. This allows the commercial large print book edition to be compared to the commercial digital learning product, or alternative media producer eBook.

The source of accessibility metadata will vary with media type and distribution. ONIX 3 has the broadest metadata model for describing both physical and digital products in terms of accessibility. ONIX 3, however, is used to support commercial distribution and is unlikely to be used by an alternative media producer. Digital metadata data standards for accessibility, such as those in EPUB3, are becoming more robust and descriptive, but physical formats such as Braille and Large Print do not readily benefit from these standards, leaving ONIX3 as the best place to communicate accessibility facts for non-digital commercial products. EPUB3 also has a strong metadata model for describing accessibility, but since it is only applicable to a digital product, it would never be used to describe a Braille or Large Print format. Access for All also has a strong metadata model, but it is mostly applicable to interactive digital products that conform to the IMS Global standard.

Accordingly, the A11y Facts system hopes to support a spectrum of data sources:

- ONIX 2 and 3 for commercial products
- Extracted metadata from ePUB3 for digital products, both commercial and from alternative media producers
- CSV data for publishers, distributors, and alternative media producers who do not produce rich ONIX
- Direct input via a user interface for publishers, distributors, and alternative media producers who wish to declare low-volume products.

A11y Facts Proposed Consumer Label for Content

A11y Facts hopes to make accessibility feature labeling as consumer friendly as possible and intends to interview a range of students with disabilities to test descriptive terms for understanding. Students are seeking all types of accessible alternatives and in the course of a search for an alternative, they may be looking at many renditions or formats of a book. They may desire to compare physical, large print, Kindle, EPUB, PDF or other formats. In order to do this effectively, students would likely prefer a single taxonomy of accessibility features even though the different formats have different features. Therefore, A11y Facts will use its own uniform taxonomy for accessibility features. As currently proposed, the label structure will group content features into 11 major categories of interest to students:

| Navigation Features | Text Enhancements | Speech to Text | Text to Speech |
|-------------------------------|--|---|------------------------|
| Navigation by TOC | Short alternate descriptions | Exact transcript of audio | Optimized for TTS |
| Navigation by index | Full alternate descriptions | Full transcripts of audio | Optimized for language |
| Correct reading order | Visualized data available as non-graphical | Synchronized text with video or audio | |
| Print-equivalent page numbers | Includes annotations from author | | |
| Ability to change TOC | Optimized for dyslexia | | |
| Navigation by bookmark | | | |
| Audio Enhancement | Alternative Presentation | Accessible Control | Access mode |
| Synch'd recorded audio | Large print | Full keyboard control | Auditory |
| Audio descriptions | Audio book | Full mouse control | Color dependent |
| | Adjustable fonts, size, color | Full switch control | Tactile |
| | High contrast display | Full touch control | Visual |
| | Sign language | Full video control | Text on image |
| | Tactile graphics | Full voice control | Olefactory |
| | Tactile objects | Timing control | |
| | Braille | | |
| | Symbolic representation | | |
| Accessibility Hazards | STEM Accessibility | Conformance | |
| Flashing content | Accessible math content | Compliance web page | |
| No flashing hazard | Accessible chem content | Compatibility tested | |
| Motion simulation | Nemeth Braille content | Trusted intermediary web page | |
| No motion sim hazard | | Trusted intermediary contact | |
| Contains sounds | | Publisher's web page for accessibility info | |
| No sound hazards | | Publisher's contact for accessibility info | |
| No olefactory hazards | | Conforms to: | |
| | | Certified by: | |
| | | Digital Rights Management restrictions | |

The taxonomy and feature names shown here are proposed and will be tested with students for viability and understanding. Each of the features shown may, in addition, require driving down to an additional level of detail (e.g., Audio book requires description of media, Visual access mode requires detail on text, math, chemistry, etc.). Each feature name listed here may also be a summary descriptor that leads to more detailed description, for example, an audiobook could be CD, DVD, or downloadable MP3.

The A11y Facts database will allow direct entry of accessibility information either via the user interface, CSV or API. This type of maintenance may often be used by smaller publishers or producers of alternative media. Or the accessibility features may be provided by mechanized sources such as ONIX, EPUB3 metadata or Access for All package. Our intent is to create systems that can ingest ONIX feeds, or ingest an EPUB3 for metadata extraction only (not to retain the file) as well as any other mechanism that can provide large amounts of reliable data. To enable this strategy, we must face the challenge of mapping values from disparate schema's into the A11y Facts label, however we feel this provides a more user-friendly result. A few examples of the mapping show the challenge:

Navigation Features

Navigation by Table of Contents: True if ONIX3 <b333> is A202,A203,A204,A205,A206,A208,A209,A210,A211,A212, or <b335> is 11, or Ally (ePUB3)metadata is tableOfContents or Access for All metadata is StructurePresentation

Corrected Reading Order: True if ONIX3 <b334> is 9 and <b335> is 13 or Ally metadata is readingOrder or if Ally metadata is taggedPDF

Alternative Presentation

Large Print: True if ONIX3 and <x419> is LTE or ULP, or if ONIX2 and <b051> is LTE or ULP, or Ally metadata is largePrint

Braille: True if ONIX3 and <x419> is BRL, or if ONIX2 and <b051> is BRL, or if <b333> is B704 or B705 or B706, or if Ally Metadata is braille or if AfA metadata is Braille

In addition to the mapping, we will need to identify an order of data preference should we get conflicting metadata from two different sources. We feel the most likely source of this kind of data from publishers will be via ONIX. ONIX in many ways is the richest schema, yet it does not include many of the newer metadata identified by Schema.org and supported by EPUB3. It is our hope that we can influence Editeur to add required codes to support all of the metadata which can be identified by Schema.org.

Source Data for Reading System and Assistive Technology Products

The BISG EPUB3 Support Grid is currently the most comprehensive source for accessibility features for EPUB3 reading systems, and the A11y Facts Label system hopes to leverage this work to present this information, either indirectly through linkages, or directly through import into the A11y Facts system. Additionally, A11y Facts would like to be able to present information on the features and capabilities of assistive technology which is not focused on EPUB3. Products such as Kurzweil 3000 and Claro Read are not EPUB3 reading systems, but still relevant choices for many students with disabilities, and may be applicable to formats, such as PDF, outside the realm of EPUB3 standards. A11y Facts should be able to represent the capabilities of these products to students who are seeking information on assistive technology that can be useful to them. For these types of products, A11y Facts plans to support a declarative user interface that allows a developer to input relevant data about product capabilities for prospective customers.

Content/Platform Combinations

The fact that content producers may deliver content which can be accessed by multiple platforms introduces the complexity that A11y Facts may require entries for the same content expressed through multiple platforms. A publisher may produce an EPUB3 file that is fully conformant to the EPUB3 specification, but that has different performance capabilities on different platforms. An EPUB3 file may contain valid MathML or complex tables that a specific reading system cannot render. Or there may be features that a reading system supports, like self-voicing text-to-speech, but a given ebook may have digital rights constraints (DRM) that suppresses the feature due to licensing restrictions. Therefore, we feel that a given book may require an A11y Facts Label on a per-platform basis. So, we feel that it is important the A11y Facts Data Model can support multiple representations of content and content within platform. It is conceivable that a publisher creates an EPUB3 version of *Biology*, with a set of accessibility criteria on one label, but there could also be a label for that same text as rendered by VitalSource, ReadHear, or EasyReader.

The Role of Testing

Early in our development of the A11y Facts concepts, we considered potential mechanisms that could test a digital product for accessibility features. However, upon further consideration we determined this would not be our best approach. While CAMI strongly endorses the concept of testing accessibility features and claims, we first and foremost want to invite and encourage publishers, alternate media producers, and others to display their accessibility features within the A11y Facts system. Furthermore, not all accessible formats are digital. We feel that the real sea change will be providing visibility to this data which will both (1) encourage publishers and producers to provide the data and (2) encourage consumers to review the data. Therefore, at least initially, CAMI is not planning its own testing of accessibility features and claims, instead it will focus on providing a venue in which such claims can be stated. ONIX 3, and proposed standards of EPUB3 already provide mechanisms for documenting and expressing conformance and conformance certification. The intent of the A11y Facts Label is to display the conformance that is reported in ONIX 3 or the EPUB3 metadata. It is similarly our hope that providing the venue

to express both conformance and certification will be motivational for publishers and other content producers to perform testing that can be reported within A11y Facts.

Publisher Perspective: Pearson

Pearson is the world's largest learning company, developing and producing content and digital products for learners at all stages, from early childhood to professional development. All content and digital products are regularly updated and enhanced to meet the changing needs of our learners. Thus, different products will have different accessibility profiles. While all new digital product and content development aims to be born accessible, Pearson continues to support older, legacy products which are enormously valuable to our customers but which have varying degrees of accessibility. Pearson evaluates whether older products should be refreshed for accessibility taking into account the popularity of the product and the proximity of the release of the next edition or refresh.

The A11y Facts Label is an attractive option, as it would provide transparency across all products, both new and old. Like other publishers, Pearson may make available specific versions that appeal to people with specific disabilities, such as loose-leaf editions, which can also be described by the A11y Facts label.

Keeping track of each product's accessibility profile is a challenge for any large publisher. Pearson continues to experiment with documentation formats, such as VPATs (which most consumers can't interpret); so-called "User Guides," which approach accessibility by categories of interaction or asset; and a variety of simple checklists. The VPAT was designed for evaluation by procurement officers and is not a consumer-focused document. Furthermore, there is not a singular place where this information may be found. The A11y Facts Label is an attractive option: it would be standardized, easy to populate and keep current, and provide transparency across all of our products and our competitors' products.

This last aspect is not unimportant. One of the biggest challenges Pearson faces is having outdated, or simply incorrect, information about Pearson products broadcast to the marketplace not only by competitors, but also by "well-meaning" experts who haven't taken the time to verify their information with us. To the extent that improving standards allow systematic testing and reporting of conformance and compliance, Pearson feels that this situation will be improved. The existence of the A11y Facts Portal, which can display the most up-to-date accessibility information, would significantly improve this situation.

Pearson puts considerable effort into making certain that new products are "born accessible," in hopes that students with disabilities can be served in the market like everyone else, instead of relying upon third party remediation by alternative media producers and intervention by Disability Support Service offices. Despite the advantages to students with disabilities being able to obtain accessible materials as easily as non-disabled students obtain their materials, there are some important challenges in adopting the market model.

First, even accessible products may not be easily adjusted to suit a student's unique needs. Students who are used to their campus disability service office tailoring "alternate format" materials to the student's requirements may find the commercial product less user-friendly. The

disability service office often distributes unprotected Word or PDF documents which may be preferred by the student, but which are less appealing to the publisher.

In addition, if a student buys a commercial electronic textbook and finds it is not accessible for their specific situation, they'll have to deal with customer service and potentially obtain a refund, which may be both time consuming and frustrating. A single source for accessibility information – of product, platform, reading system, and assistive technology – that students, instructors, disability support staff, and product evaluators can visit *before* purchasing their materials will help level the playing field.

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International ([CC BY-NC-ND 4.0, https://creativecommons.org/licenses/by-nc-nd/4.0/](https://creativecommons.org/licenses/by-nc-nd/4.0/)).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

The Accessibility Switchboard: A New Web Portal Resource and Community of Practice for Tackling Accessibility at the Organizational Level

Chris M. Law

Accessibility Track Consulting, LLC
McLean, VA, USA
chrismlaw@accessibilitytrack.com

Anil Lewis

National Federation of the Blind Jernigan Institute
Baltimore, MD, USA
alewis@nfb.org

Abstract

This presentation occurs at the mid-point of the first year of a project to create a web portal of resources to help people in industry adopt organization-wide approaches to accessibility, and to help consumers identify and articulate to industry their needs with respect to Information and Communications Technologies (ICT). The project website will serve as a connection point for industry, consumers, and others interested in addressing the needs of people with disabilities with respect to employment, the workplace, and ICT used at home, work, or on the move. The creation of the web portal and resource guides is informed and supported by a twenty-member Community Of Practice consisting of consultants and other subject matter experts from the accessibility field. End-user and thematic considerations are presented, along with descriptions of the types of resource being generated, and the review process. Future accessibility field engagement plans beyond the first year are discussed.

Introduction

Background

The National Federation of the Blind is building a resource portal for adopting organization-wide approaches to accessibility. The name “[Accessibility Switchboard](http://switchboard.nfb.org/)” (<http://switchboard.nfb.org/>) was chosen to represent the idea of connecting people in need to existing published resources and organizations.

The project is being led by The National Federation of the Blind Jernigan Institute (NFBJI). Through its projects and programs, the NFBJI “leads the quest to understand the real problems of blindness and to develop innovative education, technologies, products, and services that help the world's blind to achieve independence.”

In this presentation we will cover the impetus for the project, the methods used, resources being generated over the first year, and proposed future developments.

Impetus

A company that had little or no prior exposure to ICT accessibility issues might decide to make its public-facing web presence accessible. The spark may be from an internal advocate wanting to “do the right thing”. It may be because of anticipation of future regulations imposing requirements for web accessibility. Or it may be as the result of a complaint brought against the company. Regardless of the impetus, the process of analyzing and remediating things like the company’s public facing website, mobile site, and social media posts, is technically quite straightforward. Consultants may be temporarily hired to help with the process. Analysis tools, guidance, examples and so forth are easy for development and testing team members to find from an Internet search. The guidance and tools can then be put into practice to make fixes to web elements.

But what happens a year later? Without turning a one-off fix into changes in day-to-day operations, such a website could easily lapse in its accessibility. A new developer is hired who does not know of the difficulties presented by mouse-only interfaces, and a whole new section of the website is now inaccessible to keyboard-only users. A new content provider is introduced, and they do not use tags for heading formatting, and have never heard of alternate text. New articles get posted with no testing for accessibility.

Many in the accessibility field are recognizing that enculturation of accessibility into various departments and activities in workplaces is needed to bring about sustained change. The technological aspects of accessibility are well documented. The non-technical aspects of introducing accessibility as an organization-wide commitment are in the early stages of dissemination. A small handful of books, webinar presentations and articles have been produced so far, but this is in fact a very broad field of practice.

There is a great deal that can be done by individuals and teams in organizations to incorporate accessibility in their culture (culture being defined as what people say and what they do), their product development processes, and their interactions with customers and suppliers. Experience shows, however, that the pace of change is typically very slow.

An aim of the Accessibility Switchboard project is to help speed up the process of gathering and disseminating knowledge on organization-wide approaches and thus open up a wider marketplace for consultant practitioners in the field. Related aims are to provide industry practitioners with initial guidance that helps them understand these nontechnical issues, and connect them to existing and emerging resources; and to provide consumers with the means to articulate their needs to industry (ICT vendors and others) and disability organizations.

Development Methods

Community of practice

We set out to develop new freely available published resources, tailored to the needs of industry and consumers. To this end, the Jernigan Institute team is supported by a community of practice of twenty subject matter experts, ten from the consulting field, and ten including disability organizations, academics, government agencies representatives, nonprofits, and a disability-focused technology company.

The community of practice convened to review and revise an initial project plan, and came up with a list of target resources for a one-year pilot project. Later, the community members were involved in the review of the resources being generated.

Half of the community of practice members are consultants. This was a purposeful choice, as it was anticipated that they could contribute case-studies based on real life experiences, and that this content would be key to convincing readers that change was achievable, not conjectural.

Resource considerations

End-user considerations

Early on, we recognized that the characteristics of the potential audience will vary by type of reader. For example:

- **Impetus:** Is the reader working for an organization that is interested in gaining a competitive advantage, or wanting to ‘do the right thing’? Or, is she responding to an outside force, such as a complaint or the threat of a lawsuit, because her team has consistently neglected requests to create inclusive products and services?
- **Size:** Does the reader work for a multi-national ICT development company with tens of thousands of employees? Or, is he the manager of a family-owned restaurant whose web presence is under scrutiny?
- **Accessibility Experience:** Is the reader a consumer with low vision, who is familiar only with the assistive technologies she has previously and currently uses? Or, is the reader running an accessibility program, having a generally broad familiarity with various technical aspects of accessibility, but she frustrated at her inability to significantly impact the accessibility of the her organization’s products? Or, is she a product manager or executive who knows very little about accessibility, and has been informed that she has an organization-wide problem to address?
- **Organizational level:** Is the reader a mid-level manager who has time to research and learn the area? Or is he the Chief Information Officer who has to juggle many pressing demands on his time?

Current organizational change resources prescribe differing step-by-step tasks. Various authors use similar-but-different terms. Some resources are based on real-world experiences of those who have helped implement organizational change; whereas other resources are based on logical assumptions, but may be more conjectural. Current published resources are also somewhat

generic. Because they are not geared to different types of reader, figuring out where to begin can be quite a daunting task for the novice.

Confronted by resources that are not tailored to certain cohorts, an industry-based or government-based reader may instead opt for hiring a consultant expert. The approach taken in the clients' project will vary by the chosen consultant. This type of consulting service is becoming increasingly popular, but it is still new, and so there is no established commonality when it comes to providing such services. Of course the reputation and referrals of the more successful consultant firms provide some measure of success, but unfortunately the success stories of their individual client organizations is not available to teach to the wider community. This can be because the work was conducted under a non-disclosure agreement, and/or the consultant is too busy or not inclined to write resources to share their wisdom.

Our problem in consulting is definitely one of scale. Even if each consultancy serves a large handful of clients in a given year, there are more potential clients than can be handled. There are hundreds of federal and state agencies, thousands of ICT development companies, and hundreds of thousands of small business who use ICT. For those who cannot afford to hire consultants, the problem of scale can only be addressed by providing better training and education resources. Resources should be tailored to the diverse needs and characteristics of different potential readers. Resources should be based on the best current knowledge, and should be based on actual success stories produced by people who practice in this area.

Thematic considerations

Following from the above considerations, a set of thematic considerations were generated at the outset. These considerations would guide the creation and review of the new content:

- Assume little to no prior knowledge about the topic areas on the part of the reader.
- Establish strategy first. Strategy leads to plans. Plans lead to tactics.
- Emphasize the importance of readers to understand and document their initial state through the conduct of informal or formal maturity assessments.
- Resourcefulness trumps having resources.
- Real-life case studies that people can learn from will carry more weight than directives.
- There will inevitably be resistance to change, and resources to help readers handle and counter resistance must be provided.
- Building accessibility into mainstream development processes is the aim, and the message must include emphasis that the earlier in the process, the less expensive it is to accomplish.
- This is an information resource site containing pointers to new and existing content; it is not intended as a civil rights or legal website.

Resources generated

Industry-Focused Guidance Resources

Industry is a broad audience for the project, ranging from small businesses and startups all the way up to multinational corporations. Five guides have been created:

1. A roadmap for organizational accessibility for large ICT vendors.
2. How to provide an accessible work environment.
3. Beyond offering employees a complaint process: proactive measures to tackle accessibility issues.
4. Introducing organization-wide accessibility approaches: a guide to making a successful pitch.
5. An introductory guide for small businesses and startups.

Consumer-Focused Guidance Resources

We aimed to provide guidance tailored for consumer audiences on the following three topics:

1. Documenting a day in your life: demonstrating the level of accessibility of the technologies you interact with, and those you cannot interact with.
2. Best practices in organizational commitments to accessibility.
3. How to approach change makers: what are the pros and cons of various options?

Q&A Resources

In addition to the main guides, we authored nine shorter “question and answer” articles that gave background information, and also pointed to existing resources providing further information:

1. Why would I want to address accessibility in my organization?
2. Does accessibility have to be expensive?
3. I’ve been told to make sure the things I procure are accessible. How do I do this?
4. I have a job applicant who has notified me they have a disability. What should I do now?
5. How do I find a knowledgeable consultant?
6. How do I ensure my products work for people with disabilities?
7. What procedures should I use to test my ICT for accessibility?
8. Where do I get good/accurate information on ICT (web, software, mobile, electronic document) accessibility?
9. What are the steps I can take to reduce the likelihood of getting sued over the accessibility of my ICT?

Contact Resources

For consumers, we developed a contact form for seeking help from the National Federation of the Blind, and we developed resources to help them contact vendors directly.

For all users, we provided a means to provide feedback to the authors for future updates to articles, and links to new resources and case study information as it is generated.

Website Resource

The above resources were integrated into a website that was developed in tandem with the written resources, and was evaluated by the community of practice members.

Review process

The Nineteen new resources in the form of short guides and “Q&A” articles were developed by the project lead. The resources were then reviewed by the community members over a three month period. Reviewers would evaluate the new guidance and the linked resources, and provide case study input and general feedback for revision and enhancement.

After the community of practice review edits are done, a wider beta-style review is planned, opening up the draft review to a wider group of consultants and other subject matter experts, as well as industry and consumers. Following this second three-month review processes the resources will be published on the Accessibility Switchboard web portal.

Future Developments and Continued Engagement

The long term aim is to sustain this dynamic web portal by continuing the growth of contemporary content through the active engagement of subject matter experts as well as consumers.

The content was necessarily tailored to US audiences in the pilot. In future years more international content may be facilitated.

In the pilot, the primary audience for the guides has been necessarily limited to industry and consumers. In future years, government, higher education, and other audiences may be added to the Switchboard.

As we progress with future Accessibility Switchboard projects, we welcome engagement in terms of participation, content generation, review, and support. Please contact the authors for more information.

Acknowledgments

The Accessibility Switchboard is being developed by the NFB as a one-year pilot project under the NFB Center of Excellence in Nonvisual Access (CENA) with Non-Visual Accessibility Initiative grant funds from the State of Maryland.

The authors wish to express their sincere thanks to the members of the Community Of Practice for their insights and support in the conduct of this project. (For a list of members, visit the “[Accessibility Switchboard web portal](http://switchboard.nfb.org/)” at <http://switchboard.nfb.org/>.)

Copyright notice

The 2016 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, <https://creativecommons.org/licenses/by-nc-nd/4.0/>).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Symposium Committee

Chris M. Law (Chair)

Organizational Accessibility Consultant, Accessibility Track Consulting, LLC

Norman Robinson (Co-Chair)

Agency Section 508 Program Manager, Department of State

John Rempel (Awards Chair)

Quality Control & Training Specialist, AMAC Accessibility Solutions & Research Center at Georgia Tech

Shadi Abou-Zahra

Chair, Evaluation and Repair Tools Working Group, W3C Web Accessibility Initiative (WAI)

Holly Anderson

Former Chair of the Federal Accessible Electronic Documents Community Of Practice (AED-COP)

Jon Avila

Chief Accessibility Officer, SSB Bart Group

Robert Bohn

Cloud Computing Program Manager, National Institute of Standards & Technology (NIST)

Phil Jenkins

Accessibility Business Development Executive, IBM

Chad Lamb

Testing Program Manager, New Editions Consulting, Inc.

Jonathan Lazar

Professor of Computer and Information Sciences, Towson University

Anil Lewis

Executive Director, National Federation of the Blind Jernigan Institute

Clayton Lewis

Fellow, Institute of Cognitive Science, Professor of Computer Science, University of Colorado

Debra Martin

Federal Engagement Manager, Deque Systems, Inc.

Andrew Nielson

Senior Accessibility Technician, New Editions Consulting, Inc.

Kathleen McCoy

Assistive Technology Committee Co-Chair, The Interagency Committee on Disability Research (NIDILRR/ICDR), & Professor and Chair of Computer Science, University of Delaware

Earl McJett

Section 508 Coordinator, Federal Deposit Insurance Corporation (FDIC)

Bill Peterson

Executive Director, Office of Accessible Systems and Technology, Department of Homeland Security

Madeleine Rothberg

Senior Subject Matter Expert, National Center for Accessible Media / WGBH

Cyndi Rowland

Executive Director, WebAIM, Utah State University

Josef M. Scarantino

Founder, Scarantino Consulting LLC

Paul Schroeder

Vice President, Programs and Policy Group, American Foundation for the Blind

Author Index

Abou-Zahra, Shadi, 13
Avila, Jonathan, 39, 63
Brewer, Judy, 3
Clinton-Brown, Cynthia, 9
Eng, Katherine, 9
Garrison, Alistair, 63
Gunderson, Jon, 5, 21
Hoffman, Allen, 9
Law, Chris, 1, 47, 105
Lewis, Anil, 105
Lewis, Larry, 71
Martinengo, Robert, 93
McLeod, Matt, 29
Moser, Kenneth, 9
Ober, Elaine, 93
Romanski, Kare, 75
Rowland, Cyndi, 17
Whiting, Jonathan, 17, 57
Wild, Gian, 29, 81