opentext™

IT Accessibility Plan

Rachele DiTullio

Contents

Executive Summary

Starting 1 January 2021, the Accessibility for Ontarians with Disabilities Act (2005) requires all public-facing websites and web content for companies headquartered in Ontario conform to the *Web Content Accessibility Guidelines* (WCAG) 2.0 Level AA, a set of 38 success criteria used for testing.

OpenText's digital properties must be surveyed, counted and categorized to determine:

- Legacy properties that require updating
- New properties that require prevention
- Compliant properties that require monitoring

Some properties, like www.opentext.com, have undergone accessibility updates and will be brought into compliance during the OpenText Modernization project in flight.

All other sites must be evaluated for

- Remediation: Keep existing website "as-is" and fix as many accessibility problems as possible.
- Replacement: Rebuild the website with accessible best practices.
- Retirement: Migrate needed content to a compliant website and decommission the old one.

What is digital accessibility?

Digital accessibility refers to the extent that a digital property is inclusive of disabled people, such as those with low vision, deafness, motor or cognitive impairment. It's designing *with* disabled people to know what their needs are. If a website isn't designed with disabled people in mind, then it's not likely to be accessible.

Much in the way coding standards and expectations had to change with the introduction of responsive design, accessible design is a new way of thinking about the entire design and development process.

OpenText employees will need accessibility training to learn what the conformance standards are, how to meet them and how to test for accessibility issues before code is released.

Industry best practice indicates that it takes upwards of two years for large organizations to establish a successful accessibility program. This starts with leadership buy-in to an accessible process including:

- Time to train personnel
- Budget for accessibility software and testing with disabled people
- Accountability for accessibility goals

The preliminary count of sites requiring assessment is 33.

Introduction

A successful accessibility program aims to reduce the number of accessibility issues in the code of websites and web applications. The tendency is to think about accessibility only at the QA testing stage, but accessibility is everyone's responsibility.

- **Manager:** Understand and champion good accessibility
- **Business analyst**: Understand the implications of poor accessibility
- **Designer**: Understand and design for accessibility
- **Developer**: Understand and build for accessibility
- **QA tester**: Understand and test for accessibility issues

Personnel need a broad understanding of accessibility concepts, practices and requirements. The following training and testing information seeks to establish a conprehensive process for preventing, finding and fixing accessibility issues that can be used for every web property at OpenText.

The goal is to standardize on an accessibility process that is centrally managed and available to IT and Digital Marketing teams.

The appendex is the start of an inventory of OpenText web properites supported by IT.

The rest of this document outlines process changes to

- Training on accessible coding methods
- Testing for accessibility
- Remediating accessibility issues

Training

Below are some of the gaps where IT staff require training. Everyone needs to understand what accessibility is as well as how to write good code that doesn't contain accessibility issues.

Semantic HTML

The very foundation of accessible code is using semantic HTML and letting the browser perform as much native functionality as possible, e.g. use a button element instead of creating a custom control. The browser handles click functionality, works with space/enter keys and gets focus—functionality a developer would have to specially add to a custom control using JavaScript.

Developers need to know HTML: Beginner's Guide to Writing Good HTML

Common accessibility issues caused by bad HTML include:

- Interface controls that don't work with a keyboard
- Form inputs that don't have programmatically accessible labels
- Focus management through complicated interactions

Accessibility

Accessibility is essential for individuals and organizations that want to create high-quality websites and web applications–and not exclude people from using their products and services. All IT staff should take a basic accessibility course, such as this one from edX: Introduction to Web Accessibility

Learn about the international standards for web accessibility from the W3C–including Web Content Accessibility Guidelines (WCAG) and WAI-ARIA for Accessible Rich Internet Applications–and first steps in applying them, the broad scope of web accessibility and how disabled people use different assistive technologies and adaptive strategies.

1) **Module 1: What is Web Accessibility**—Introduces stories of people with disabilities, defines and scopes web accessibility and introduces its interrelations with other disciplines.

2) **Module 2: People and Digital Technology**—Introduces some accessibility features and barriers; presents some adaptive strategies and assistive tools; and introduces the components of web accessibility.

3) **Module 3: Business Case and Benefits**—Introduces the business case for web accessibility and presents some benefits, such as enhancing the brand, increasing market reach, driving innovation and minimizing legal risk.

4) **Module 4: Principles, Standards, and Checks**—Introduces the principles of web accessibility, as well as the international W3C Accessibility standards, and how these are developed. Provides hands-on experience checking how web pages have implemented the standards.

5) **Module 5: Getting Started with Accessibility**—Provides an overview of organizational planning and managing considerations to integrate accessibility throughout the web production process, and of different accessibility roles and responsibilities involved.

Web Content Accessibility Guidelines (WCAG)

Accessibility under the AODA is evaluated using the WCAG 2.0 Level AA success criteria (38). These are used for accessibility testing and are also mapped to baseline tests for manual testing. It's important to note that WCAG 2.0 became the standard in December 2008. WCAG 2.1 is the industry standard (12 additional Level A and AA success criteria) and WCAG 3.0 is coming soon.

Using assistive technologies (AT)

Common AT uses include:

- **Screen reader**: NVDA and JAWS
- **Voice to text**: Dragon Naturally Speaking
- **Screen magnification**: ZoomText

At minimum, use of a screen reader is necessary for catching keyboard traps, hidden content that is exposed to AT and the accessible names of controls. The American Federation for the Blind (AFB) has free training on how to use the NVDA screen reader.

Testing

Accessibility testing happens at many points along a release cycle. By "shifting left" the responsibility for identifying and preventing accessibility issues, fewer problems should be detected during QA testing.

Plan: Business analysts, managers and stakeholders ensure accessibility is part of the scope of work for every project.

Design: Designers create color palettes that meet contrast requirements, good font legibility, page reflow for mobile and accessible labels for interface controls.

Develop: Developers ensure all interface controls can be accessed with a keyboard; page state is communicated and page history is managed in single page applications; all links and buttons have focus indicators; focus management.

Test: QA testers perform thorough automated and manual accessibility testing to find any remaining issues.

Tool kit

Use the following tools for accessibility testing on Windows:

Browser: Chrome

Automated testing: axe devtools (Chrome extension)

Screen reader: NVDA (cheat sheet)

Color contrast: Accessible Color Picker (Chrome extension)

Headings: HTML5 Outliner (Chrome extension)

Hidden content: JavaScript bookmarklets for accessibility testing

Mobile: iPhone with VoiceOver on iOS (cheat sheet)

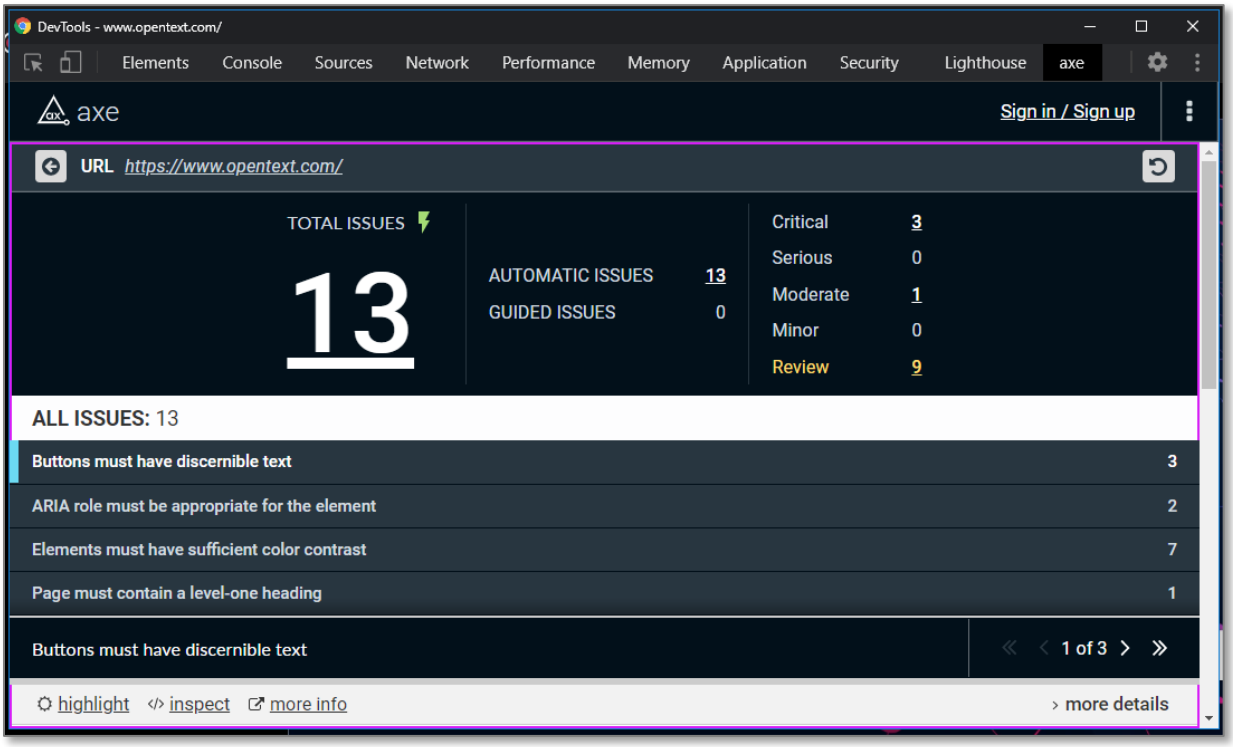
Automated testing

Using axe devtools, scan each page and screen state that has unique components. Track global components like the header and footer separately from the individual page content.

Automated testing reveals some common issues:

- Color contrast
- Buttons and links without accessible names
- Form inputs without programmatically associated labels
- Images without alt text

Automated issues are found and fixed by the developer.



Automated testing can only detect a subset of accessibility issues and must be followed up by manual testing.

Manual testing

Manual testing is required to find more difficult accessibility issues that need human verification:

- Tabbing order of the document is correct

- Alt text makes sense on every image
- Visible focus indication
- Keyboard access to all interface controls, etc.

Assess hidden accessibility information using bookmarklets:

- ARIA attributes
- Tables
- Image alt text
- Tabbing order
- Forms, etc.

Baseline tests

Baseline tests reduce ambiguity, increase consistency of results and emphasize testing of the methods and techniques that can reliably meet the AODA accessibility requirements, given the current state and compatibility of underlying technologies.

Each website and web application flagged for remediation goes through comprehensive manual testing of each page, screen or component to identify issues.

Baseline Tests	WCAG Success Criteria
1) Keyboard Access 1.1. Keyboard access 1.2. No keyboard trap	2.1.1 Keyboard 2.1.2 No Keyboard Trap
2) Focus 2.1. Visible focus 2.2. Focus order 2.3. On Focus	2.4.3 - Focus Order 2.4.7 - Focus Visible 3.2.1 - On Focus
3) Non-Interference 3.1. Non-interference	1.4.2 - Audio Control 2.1.2 - No Keyboard Trap 2.3.1 - Three Flashes or Below Threshold 2.2.2 - Pause, Stop, Hide
4) Repetitive Content 4.1. Bypass blocks	2.4.1 Bypass Blocks 3.2.3 Consistent Navigation

Baseline Tests	WCAG Success Criteria
4.2. Consistent navigation 4.3. Consistent identification	3.2.4 Consistent Identification
5) Changing Content 5.1. Changes in content	4.1.2 Name, Role, Value
6) Images 6.1. Meaningful images 6.2. Decorative images 6.3. Captchas 6.4. Images of text	1.1.1 Non-text Content 1.4.5 Images of Text 4.1.2 Name, Role, Value
7) Sensory Characteristics 7.1. Use of color 7.2. Sensory characteristics	1.3.3 Sensory Characteristics 1.4.1 Use of Color
8) Contrast 8.1. Contrast minimum	1.4.3 Contrast (Minimum)
9) Flashing 9.1. Three flashes or below	2.3.1 Three Flashes or Below Threshold
10) Forms 10.1. Form names 10.2. Form labels descriptive 10.3. On input 10.4. Error identification 10.5. Form has a label 10.6. Error suggestion 10.7. Error prevention (legal, financial, data)	1.3.1 Info and Relationships 2.4.6 Headings and Labels 3.2.2 On Input 3.3.1 Error Identification 3.3.2 Labels or Instructions 3.3.3 Error Suggestion 3.3.4 Error Prevention (Legal, Financial, Data) 4.1.2 Name, Role, Value
11) Page Titles	2.4.2 Page Titled

Baseline Tests	WCAG Success Criteria
11.1. Page titled	
12) Tables 12.1. Data tables 12.2. Layout tables	1.3.1 Info and Relationships
13) Content Structure 13.1. Descriptive headings 13.2. Visual headings are programmatic 13.3. Programmatic headings 13.4. Visually apparent lists	1.3.1 Info and Relationships 2.4.6 Headings and Labels
14) Links 14.1. Link purpose	2.4.4 Link Purpose (In Context)
15) Language 15.1. Language of page 15.2. Language of parts	3.1.1 Language of Page 3.1.2 Language of Parts
16) Audio-Only and Video-Only 16.1. Audio-only 16.2. Video-only	1.2.1 Audio-only and Video-only (Prerecorded)
17) Synchronized Media 17.1. Media player controls 17.2. Media player caption control level 17.3. Media player audio description control level 17.4. Captions (prerecorded) 17.5. Audio description 17.6. Captions (live)	1.2.2 Captions (Prerecorded) 1.2.3 Audio Description or Media Alternative (Prerecorded) *1.2.4 Captions (Live) – not required by AODA* *1.2.5 Audio Description (Prerecorded) – not required by AODA*

Baseline Tests	WCAG Success Criteria
18) CSS Content and Positioning 18.1. Meaningful background image 18.2. CSS positioned content	1.1.1. Non-Text Content 1.3.1 Info and Relationships 1.3.2 Meaningful Sequence
19) Frames and iFrames 19.1. Frames 19.2. iFrames	4.1.2 Name, Role, Value
20) Alternate Versions 20.1. Conforming alternate version	WCAG Conformance Alternate Version
21) Timed Events 21.1. Timing adjustable 21.2. Moving information 21.3. Auto-updating information 21.4. Audio control	1.4.2 Audio Control 2.2.1 Timing Adjustable 2.2.2 Pause, Stop, Hide
22) Resize Text 22.1. Resize text	1.4.4 Resize Text
23) Multiple Ways 23.1. Related web pages	2.4.5 Multiple Ways
24) Parsing 24.1. Parsing	4.1.1 Parsing

WCAG 2.0 Level AA is the floor of compliance. OpenText will benefit by including newer requirements from WCAG 2.1 that may eventually be adopted by the AODA and accessibility laws in other countries.

- Reflow (responsive design)
- Content on hover
- Pointer cancellation
- Orientation (responsive design)

- Status messages

Developers need to perform manual testing against these baseline tests, though all tests are not applicable to every page, screen or component.

User journeys

User journeys are step-by-step processes for reaching a goal utilizing a website or web application. Optimal user journeys are characterized by **independent** and **efficient** completion of sequential steps to reach the intended goal and will differ depending upon the user profile for whom the journey is designed.

User journeys are impacted by the selected platform on which the journey occurs and by the assistive technologies employed by the disabled user (laptop, kiosk, mobile device).

Components of a user journey

- Surveillance: Where am I and what's around me?
- Destination: Where do I want to be?
- Navigation: How will I get there?
- Interaction: What are the tasks that I need to perform along the way?
- Completion: Indication that the goal is achieved

Testing with disabled people

All websites and web applications should be tested by disabled people who are native users of assistive technologies like a screen reader, voice to text, braille output and zoom text across multiple devices.

AccessWorks is a service for finding disabled people to evaluate websites and web applications remotely by testing specific user journeys. Compare plans for Userzoom GO.

Basic plan: $250/month

Remediation

Automated and manual accessibility test results should be maintained over time for spot checks and for comparison during an annual accessibility review of each website and web application. While it is possible to do this with spreadsheets, there is immense benefit to using specialized accessibility software:

- Tracking of issues from discovery to fix
- Analysis of issue severity
- Automatic association of issues to specific success criteria failures

- Best practices to follow
- Integrates a manual testing methodology for accurate and efficient testing

Accessibility software

OpenText should explore using accessibility software for finding, tracking and fixing issues due to the large number of sites it manages and the complexity of robust manual testing.

- axe Auditor and axe Monitor
- mortise
- Accessible Resource Center (ARC)
- Accessibility Management Platform (AMP)

3rd party code

OpenText websites and web applications utilize a long list of JavaScript frameworks, libraries and plug-ins. Very often, this 3rd party code introduces accessibility issues. Developers need to consider the accessibility support before including any 3rd party code as they will have to fix any major accessibility issues discovered.

Appendix

Below are websites that will be consolidated into the https://www.opentext.com/ domain with content remediated as part of the OpenText Modernization project (1975084962).

Domain	Name	Solution
https://opentext.com/	OpenText	Merge content into TeamSite
https://www.opentext.com/TrainingRegistry	Training Course Calendar	Merge content into TeamSite
https://businessnetwork.opentext.com/	Business Network (EN)	Merge content into TeamSite
https://partnerblogs.opentext.com/	OT Partner Blog	Retire
https://ottechsupport.wpengine.com/	OpenText | GXS Cloud Support Services	Redirect to new landing page

https://partners.opentext.com/	Partner Portal	Merge content into TeamSite
https://www.opentext.com/opentext-world/	OpenText World (agenda builder)	Merge content into TeamSite

Public-facing websites – partial inventory

Last updated 21 January 2020.

URL	Site Name	Solution
1) https://opentext.com/	OpenText	
2) https://investors.opentext.com/	OT Investor Relations	
3) https://mysupport.opentext.com/	My Support Ticket Portal	
4) https://knowledge.opentext.com/	My Support KC	
5) https://epay.opentext.com/	ePay	
6) https://forums.opentext.com/	OpenText Forums	
7) https://login.opentext.com/	OT Connect	
8) https://developer.opentext.com/	Developer Community	
9) https://security.opentext.com/	OpenText Security	
10) https://searchopentext.com/	OpenText Search	
11) https://carbonite.com/	Carbonite	
12) https://support.portal.covisint.com/	Covisint Support Portal	
13) *.Carbonite.com	Carbonite	

URL	Site Name	Solution
14) https://www.webroot.com/	Webroot	
15) *.Webroot.com	Webroot	
16) https://www.xmedius.com/	XMedius	
17) *.xmedius.com	XMedius	
18) https://blogs.opentext.de/	Blogs (DE)	
19) https://blogs.opentext.com/	Blogs (EN)	
20) https://blogs.opentext.jp/	Blogs (JP)	
21) https://www.corporatetobank.com/	Corporate-to-Bank	
22) https://www.edileitfaden.de/	EDI Basics (DE)	
23) https://www.edibasics.com/	EDI Basics (EN)	
24) https://www.edipourtous.fr/	EDI Basics (FR)	
25) https://www.edibasics.co.uk/	EDI Basics (UK)	
26) https://www.einvoicingbasics.co.uk/	e-Invoicing Basics	
27) http://www.faxbasics.com/	Fax Basics	
28) https://blog.hightail.com/	Hightail Blog	
29) https://www.infogovbasics.com/	Info Gov Basics	
30) https://mands.gxs.co.uk/	M&S Electronic Trading	
31) https://mips.health/	MIPS	

URL	Site Name	Solution
32) https://nestle.opentext.com/	Nestle	
33) http://www.riteaidediservices.com/	RiteAid EDI	

Please contact Rachele DiTullio with any accessibility questions:

rachele.ditullio@opentext.com

Message me on Teams

+1-210-213-2795