

Digital Accessibility Standard

Standards & Shared Practices

Accessibility Consulting, Tools & Training

Enterprise Applications🡪Website Design & Operations

April 2021
Yale University

This page is intentionally blank.

**Table of Contents**

[1. Document Control 4](#_Toc70412212)

[1.1. Document Information 4](#_Toc70412213)

[1.2. Document Edit History 4](#_Toc70412214)

[2. Introduction and Purpose 4](#_Toc70412215)

[2.1. Introduction 4](#_Toc70412216)

[2.2. Industry Standards and Reference Models 4](#_Toc70412217)

[2.3. External References 5](#_Toc70412218)

[2.4. Terms and Abbreviations 5](#_Toc70412219)

[3. Scope 5](#_Toc70412220)

[3.1. Scope 5](#_Toc70412221)

[3.2. Assumptions 6](#_Toc70412222)

[3.3. Constraints and Limitations 6](#_Toc70412223)

[3.4. Dependencies 6](#_Toc70412224)

[3.5. Risks 6](#_Toc70412225)

[4. Governance and Compliance 6](#_Toc70412226)

[4.1. Governance 6](#_Toc70412227)

[4.2. Compliance 7](#_Toc70412228)

[5. Guiding Principles 7](#_Toc70412229)

[5.1. Accessibility means usability for people with disabilities. 7](#_Toc70412230)

[5.2. If ICT isn't usable, it won't be accessible. 7](#_Toc70412231)

[5.3. Multiple testing methods are required. 7](#_Toc70412232)

[5.4. Extensive training is required to perform an expert review. 8](#_Toc70412233)

[6. The Standard 8](#_Toc70412234)

[7. Recommended Processes 8](#_Toc70412235)

[7.1. Train 8](#_Toc70412236)

[7.2. Plan 8](#_Toc70412237)

[7.3. Procure 9](#_Toc70412238)

[7.4. Design 9](#_Toc70412239)

[7.5. Develop 10](#_Toc70412240)

[7.6. Testing/QA 11](#_Toc70412241)

[7.7. Documentation 12](#_Toc70412242)

[7.8. Change Management 12](#_Toc70412243)

[8. Engagement and Support 12](#_Toc70412244)

[8.1. Available Support 12](#_Toc70412245)

[8.2. Roles and Responsibilities 13](#_Toc70412246)

[8.3. Helpful Links 13](#_Toc70412247)

# Document Control

## Document Information

|  |  |
| --- | --- |
| **Document Identifier** |  |
| **Document Name** | Digital Accessibility Standard |
| **Document URL** | tbd |
| **Service Name** | Accessibility Consulting, Tools & Training |
| **Document Author** | Michael Vaughn |
| **Document Keeper** | Michael Vaughn |
| **Document Version** | 1.2.2 |
| **Document Status** | In Revision |
| **Effective Date** | tbd |
| **Last Reviewed Date** | April 25, 2021 |
| **Next Review Date** | April 25, 2022 |

## Document Edit History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Changes | Done By |
| 1.0 | 11/2018 | Initial draft. | Michael Vaughn |
| 1.1 | 12/2018 | Addressed feedback from reviewers, no substantial changes. | Michael Vaughn |
| 1.2 | 3/2021 | Remove reference to WCAG 2.0 in favor of WCAG 2.1 (where possible—ICT guidance still references WCAG 2.0 as this documentation has not been revised, for example). Substantial revision to the “Process” section to include many more specific recommendations. | Michael Vaughn |
| 1.2.1 | 4/2021 | Incorporated feedback from reviewers. | Michael Vaughn |
| 1.2.2 | 4/2021 | Additional changes from Accessibility Steering Committee | Michael Vaughn |

# Introduction and Purpose

## Introduction

The Digital Accessibility Standard supports Yale’s commitment to making its digital campus accessible to people with disabilities. Websites, web applications, desktop software, mobile applications, and digital content are often integral to the academic and administrative work of the University. User interfaces and digital content that adhere to these standards will support participation by people with disabilities in the Yale community.

## Industry Standards and Reference Models

This standard provides guidance for acquiring and building accessible technology including websites covered by Yale’s Web Accessibility Policy and other technologies not explicitly covered by that policy. It references Section 508 of the Rehabilitation Act of 1973, which clarifies how Web Content Accessibility Guidelines can be applied to all information and communications technology (ICT, see definition below).

* [Web Content Accessibility Guidelines (WCAG) 2.1](https://www.w3.org/TR/WCAG21/)
* [Revised Section 508 Standards](https://www.access-board.gov/ict/)
* [Guidance on Applying WCAG 2.0 to Non-Web Information and Communications Technologies (WCAG2ICT)](https://www.w3.org/TR/wcag2ict/)
* [Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile](https://www.w3.org/TR/mobile-accessibility-mapping/)

## External References

Documents not specifically referenced in this standard but that may provide useful information include:

* [PDF Techniques for WCAG 2.0](https://www.w3.org/TR/WCAG-TECHS/pdf.html)
* [Visual identity standards for websites at Yale](https://yaleidentity.yale.edu/web)
* [Visual identity standards for mobile applications at Yale](https://yaleidentity.yale.edu/mobile)

## Terms and Abbreviations

* University business: Activities that are carried out under the auspices of Yale University.
* WCAG: Web Content Accessibility Guidelines: a reference standard for accessible websites. The WCAG principles are easily adapted to guide the accessibility of ICT beyond websites.
* VPAT®/ACR: Voluntary Product Accessibility Template/Accessibility Conformance Report: a template created by the Information Technology Industry Council to support uniform reporting by vendors of the accessibility status of their products. There are a variety of different versions of the template to support reporting against the most common accessibility standards, namely Section 508 (U.S. Government), EN 301 549 (EU), and WCAG. The template is used to create an Accessibility Conformance Report (ACR).
* User interface: The visual and interactive component of websites, web applications, desktop software, and mobile applications, both administrative and end-user facing.
* Digital content: Content which is available online via web pages and files.
* ICT: Information and communications technology, including websites, web applications, desktop software, mobile applications, digital content, and support documentation.
* Accessibility expert: An individual with the training and experience to understand how to make ICT compliant with accessibility standards, including an understanding of how people with disabilities interact with ICT.
* Usability: “Usability is the ease of use and learnability of a human-made object such as a tool or device. In software engineering, usability is the degree to which a software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.” (Wikipedia contributors. (2018, November 1). Usability. In Wikipedia, The Free Encyclopedia. Retrieved 14:41, November 19, 2018, from <https://en.wikipedia.org/w/index.php?title=Usability&oldid=866743307>)
* Accessibility: “Accessibility is the design of products, devices, services, or environments for people with disabilities.” (Wikipedia contributors. (2018, October 25). Accessibility. In Wikipedia, The Free Encyclopedia. Retrieved 14:42, November 19, 2018, from <https://en.wikipedia.org/w/index.php?title=Accessibility&oldid=865715054>)

# Scope

## Scope

These standards apply to all websites, web applications, desktop software, mobile applications, digital content, and support documentation that are used to conduct University business and are developed, configured, or purchased by ITS staff. These technologies are referred to collectively as ICT in this document.

## Assumptions

* Building accessible ICT requires training. All staff with responsibilities involving the development or configuration of ICT will be trained to follow accessibility standards while doing so.
* Purchasing accessible ICT requires an early consideration of accessibility. All staff with responsibilities involving the purchase of ICT will be trained to consider accessibility standards early in the selection process.
* New projects and initiatives need to account for accessibility and usability requirements. Adherence to these requirements will be reviewed during ITS gating and governance processes.
* It is possible to satisfy visual identity standards for websites and mobile applications at Yale while remaining in accordance with this standard.

## Constraints and Limitations

It may not be possible or advisable to apply this standard to a specific technology if doing so would fundamentally alter that technology’s essential function.

## Dependencies

Creating accessible ICT is much easier when the person responsible for the ICT has a thorough understanding of the tools used to create the ICT. For example, someone who knows how to use Microsoft Word well will have an easier time understanding how to apply accessibility requirements to Word than a novice Word user. Similarly, a front-end developer with expert understanding of HTML5, CSS, and JavaScript will have a much easier time understanding how to apply accessibility requirements to these languages than a novice front-end developer.

## Risks

* Selecting vended products which do not meet this standard can delay the purchasing process.
* Not being properly trained, both on tools used to create ICT and on accessibility standards, will create added risk of failing to adhere to accessibility standards.
* Failing to account for accessibility early in the software development, documentation creation, or procurement process can increase the complexity and cost to that project of achieving conformance with this standard.

# Governance and Compliance

## Governance

[1605 Web Accessibility Policy](https://your.yale.edu/policies-procedures/policies/1605-web-accessibility-policy) establishes requirements for websites and web applications to comply with accessibility standards. The policy, along with [1605 PR.01 Web Accessibility Procedure](https://your.yale.edu/policies-procedures/procedures/1605-pr01-web-accessibility-procedure), describes the exception process. Not all ICT is currently covered by these policies and procedures, however all ICT is covered in this standard.

The Accessibility Steering Committee governs compliance with [1605 Web Accessibility Policy](https://your.yale.edu/policies-procedures/policies/1605-web-accessibility-policy) at a University level. Contact the Accessibility Steering Committee by emailing accessibility@yale.edu. The service and project gating process is the primary means of governance for enforcing this standard within ITS. Compliance with this standard is assessed by the ITS Digital Accessibility team at these gates: Initiate, Design, and Build. When assessing compliance, the ITS Digital Accessibility team will investigate the tools and methods being used to create accessible ICT. Before moving to production, compliance should be documented via a [WCAG checklist](https://usability.yale.edu/web-accessibility/articles/wcag2-checklist), or a [VPAT](https://usability.yale.edu/web-accessibility/working-vendors/voluntary-product-accessibility-templates-vpats).

It is the responsibility of the service owner to provide for compliance with this standard when the ITS gating process does not apply.

## Compliance

Conformance with this standard will help ITS comply with Yale’s policies and federal laws and regulations.

[Americans with Disability Act (ADA), Title III](https://www.ada.gov/ada_title_III.htm), requires private organizations to provide equal access to places of public accommodation. Some courts have ruled that certain websites are covered by this requirement, particularly when they are integral to accessing physical spaces and programming that are open to the public.

[Section 504 of the Rehabilitation Act of 1973](https://en.wikipedia.org/wiki/Section_504_of_the_Rehabilitation_Act) prohibits universities receiving federal financial assistance from discriminating against students on the basis of their disabilities. Yale is required to provide reasonable accommodations to students who have disabilities upon request. Proactively addressing the accessibility of ICT greatly simplifies meeting those accommodation requests.

[Section 508 of the Rehabilitation Act of 1973](https://section508.gov/) and standards promulgated thereunder establish requirements for ensuring the accessibility of information technology developed, maintained, procured, or used by the Federal government. The Section 508 ICT Accessibility Standards reference WCAG 2.0 AA, and clarify that WCAG applies to all ICT, including documents, software, and mobile apps. While Yale is not directly covered by Section 508, Federally contracted work done by Yale may be required to meet Section 508 standards. Section 508 can be used to help clarify how to apply the WCAG 2.0 standard to non-web ICT.

IT solution architects are expected to comply with this standard unless there is a compelling reason, consistent with [IT Architecture Principles](https://yaleits.atlassian.net/wiki/spaces/STAN/pages/781156666/IT%2BArchitecture%2BPrinciples), to do otherwise. Solutions may request an exception to the standard through the Enterprise Architecture Team.

# Guiding Principles

## Accessibility means usability for people with disabilities.

*Rationale:* The purpose of designing accessible ICTs is to improve the likelihood that people with disabilities will be able to use them. The standards used to assess accessibility are important, but do not substitute for assessing the usability of the ICT to people with disabilities.

*Implications:* Testing should include navigating through the ICT and completing tasks via assistive technology, preferably by people with disabilities.

## If ICT isn't usable, it won't be accessible.

*Rationale*: If ICT is difficult to use, it will be particularly difficult for people with disabilities. Accessibility reference standards provide specific criteria which, when satisfied, increase the usability of ICT for people with disabilities. However, application architects should also consider usability best practices as a precondition to accessibility.

*Implications:* Best practices of user experience design should be applied when planning ICT. Much ICT is not usable, which can make implementing accessibility challenging.

## Multiple testing methods are required.

*Rationale:* Both usability and accessibility require multiple means of testing to identify challenges. No single testing modality can be relied on to find the majority of real-world challenges with ICT.

*Implications:* Accessibility testing will need to be included throughout the product development life cycle. Expertise in assessing accessibility is not widespread, so planning ahead for this requirement is important.

## Extensive training is required to perform an expert review.

*Rationale:* Expert review requires extensive training to fully understand the implications of this standard. Those with some familiarity with accessibility review can provide helpful support, but the ability to perform a complete review does require a significant amount of domain-specific knowledge.

*Implications:* There are many different roles typically involved in the creation or acquisition of ICT, and most all roles need to understand this standard. For example, for a product development project, project managers and business analysts should understand the need to include accessibility requirements as an important project deliverable in the project plan. Designers should deliver designs to developers that can be implemented in conformance with this standard. Developers should understand how to test and implement accessible code using automated and manual methods. Quality assurance analysts should be able to verify that builds conform to this standard. Technical writers should understand how to create accessible documentation to support the product.

# The Standard

Compliance with this standard can be assessed via WCAG 2.1 (see reference standards above). To meet the requirements of this standard, ICT must comply with WCAG 2.1 Level AA. To meet Level AA conformance, all Level A and Level AA Success Criteria within the standard must be satisfied.

Where WCAG refers to web interfaces, it is appropriate to substitute mobile or desktop interface language (refer to the WCAG2ICT, W3C/WAI Mobile Accessibility, and Section 508 guidance included in the reference standards). A variety of evaluation methods should be used to test the accessibility of ICT, including automated testing, expert evaluation, and user testing including the use of assistive technology.

# Recommended Processes

Accessibility should be incorporated into all steps of the software development lifecycle, as well as procurement, document creation, and other ICT acquisition and development processes. The exact steps to incorporate into a process will depend on the team and the technologies in use. The example process steps below should be modified as necessary. Some of the specific accessibility strategies and tools mentioned may not apply to the platforms in use throughout ITS, so service owners should seek out equivalent strategies and tools to support testing and creating accessible technology that is specific to your platform.

## Train

* Training is required for all roles involved in the acquisition or creation of ICT.
* The Yale ITS Digital Accessibility Team creates and maintains recommended training resources by role, which are available at <https://usability.yale.edu/accessibility-trainings>
* Project managers and supervisors can contact accessibility@yale.edu to receive confirmation that team members have completed recommended training.

## Plan

* Project managers and supervisors should complete the “project management” track of accessibility training offered by the ITS Digital Accessibility team and should ensure that project staff and contractors complete training based on their role.
* If Business Analysts are involved in a project, they should complete the “BA” track of the accessibility training offered by the ITS Digital Accessibility team.
* Provide that all selected technologies support accessibility or can be modified to do so.
* Provide that project requirements include accessibility requirements.
* Project budgets should consider costs for:
	+ Time to train team members in accessibility
	+ Use of third-party accessibility firms for QA if team members lack the required expertise. Redacted or Redacted are often used on ITS projects.
	+ Use of third-party accessibility firms to facilitate user testing with people with disabilities. Redacted has been used on ITS projects. A recent option that is growing in popularity is Redacted.
* Contact the Digital Accessibility team for cost estimates and contact information for the vendors recommended above.

## Procure

* Anyone making purchases of off-the-shelf software or hiring contractors to complete work should complete the “project management” track of accessibility training offered by the ITS Digital Accessibility team.
* Provide that all purchased technologies support accessibility or can be modified to do so.
* Follow the guidelines outlined at the [Working with Vendors](https://usability.yale.edu/web-accessibility/working-vendors) section of the digital accessibility site.
* Accessibility requirements should be made clear to vendors/contractors as early as possible in the selection process. Accessibility questions/language should be included in RFIs/RFPs (requests for information or purchase) and SOWs (statements of work). RFIs/RFPs should ask vendors for a recent (no older than 18 months) ACR (VPAT) for existing deliverables.
* A vendor’s response to RFI/RFP questions related to accessibility should be considered a very high priority when selecting vendors for further consideration. If you do not feel confident assessing RFI/RFP answers to accessibility questions, reach out to accessibility@yale.edu for help reviewing responses.
* All agreements related to the acquisition or creation of ICT, including Services Agreements, should include the Accessibility Addendum.

## Design

* Train designers to consider accessibility in their designs and to recognize potential barriers in existing designs. Designers should complete the “design” track of accessibility training offered by the ITS Digital Accessibility team.
* Refer to specific guidance on accessibility considerations for designers, including the [Asking Vendors About Visual Design](https://usability.yale.edu/web-accessibility/working-vendors/asking-vendors-about-visual-design) page.
* Designers should be expected to deliver designs to developers with clear information a developer will need to implement accessible builds. Some examples include:
	+ Choose colors that conform with color contrast requirements in WCAG. Use the [WebAIM Color Contrast Checker](https://webaim.org/resources/contrastchecker/), [Contrast Triangle](https://contrast-triangle.com/), or similar tools to verify color choices.
	+ Avoid use of color alone to convey meaning, for example to identify links.
	+ Include skip links in designs, to allow keyboard and screen reader users to skip navigation elements repeated on each page.
	+ Include controls in designs for content that starts automatically (for example, a video that lasts longer than 5 seconds must have an easy to find play/pause button).
	+ Provide designs for focus indicators and focus states for interactive controls (to support keyboard-only use) that conform to requirements in WCAG.
	+ Provide alternative text for images or indicate if an image is decorative.
	+ Specify ARIA landmarks, roles and labels for visual elements and regions of a page.
	+ Design a logical heading structure.
	+ Use [Semantic HTML](https://developer.mozilla.org/en-US/docs/Glossary/Semantics). For example, make it clear what is a button and what is a link.
	+ Use native HTML5 controls as much as possible. If custom widgets are designed in place of standard HTML controls (not recommended), provide details about the expected interaction for mouse, mobile and keyboard use (follow the [ARIA Authoring Practices](https://www.w3.org/TR/wai-aria-practices-1.1/)).
	+ Provide designs that include various viewport sizes, including for mobile and tablet use in portrait or landscape orientation. Designs should be responsive at various viewport sizes, custom font sizes, and page zoom levels.
	+ Include form feedback in designs, that does not rely solely on color (for example, relying on the color red to indicate an error), and provide clear information about form status.
* Design pages to include an “Accessibility at Yale” link in the footer pointing to <https://usability.yale.edu/web-accessibility/accessibility-yale>, as outlined in the [Accessibility Statement Page](https://usability.yale.edu/web-accessibility/accessibility-yale/accessibility-statement-page) and required by the [Web Accessibility Policy](https://your.yale.edu/policies-procedures/policies/1605-web-accessibility-policy).
* Designers should utilize the [Yale UI Component Library](https://yale-a11y.gitlab.io/ui-component-library/) to help design accessible front ends. Email ux@yale.edu to obtain an Adobe XD library containing components that match the UI Component Library assets to simplify wireframe creation.
* Additional recommend resources include:
	+ [Stark](https://www.getstark.co/) plug-ins for Sketch, Figma, Adobe XD
	+ [A11y - Focus-Orderer](https://www.figma.com/community/plugin/731310036968334777/A11y---Focus-Orderer) plugin for Figma
	+ [Accessibility Bluelines](https://dribbble.com/shots/6269661-Accessibility-Bluelines) template for accessibility markup for Sketch, Figma, Adobe XD and InVision Studio

## Develop

* Developers should complete the “developer” track of accessibility training, offered by the ITS Digital Accessibility team, to understand accessibility basics and specifics of HTML/CSS/JS accessibility. Additional training should be sought for application-specific accessibility considerations unique to platforms used by the development team.
* Provide that all front-end code satisfies the standards, whether written directly or created via a back-end system.
* Use [Semantic HTML](https://developer.mozilla.org/en-US/docs/Glossary/Semantics) (particularly important to consider if using a single page application framework such as React, which tends to rely too heaving on generic HTML elements, especially <div>).
* Use native HTML5 controls as much as possible. If you must create custom widgets follow the [ARIA Authoring Practices](https://www.w3.org/TR/wai-aria-practices-1.1/). Custom widgets should implement all of the expected actions for mouse, keyboard and screen reader use.
* Test frequently with automated and guided manual accessibility testing tools.
	+ Take advantage of free accessibility browser extensions to run automated tests and guided manual tests, including:
		- [Lighthouse](https://developers.google.com/web/tools/lighthouse), included in Chrome and Edge developer tools
		- [WAVE](https://wave.webaim.org/extension/) from WebAIM
		- [axe](https://www.deque.com/axe/browser-extensions/) from Deque
		- [Siteimprove Accessibility Checker](https://siteimprove.com/en-us/core-platform/integrations/browser-extensions/)
		- [ARC Toolkit](https://www.tpgi.com/arc-platform/arc-toolkit/) from TPGi
		- [Accessibility Insights](https://accessibilityinsights.io/) from Microsoft
	+ Use the [webhint linter](https://webhint.io/) (VS Code plug-in available) to detect accessibility and other web best-practices while coding.
	+ Incorporate automated tests for accessibility into your overall testing strategy, including unit, functional and integration tests as appropriate.
	+ Prevent code commits if automated tests produce errors, either by using pipelines or actions configured in your repository, or by implementing developer environment standards (for example, using git hooks to run [axe-core/cli](https://github.com/dequelabs/axe-core-npm/tree/develop/packages/cli) installed locally).
	+ Integrate [axe-core from Deque](https://github.com/dequelabs/axe-core), or a similar solution, into your tests/pipelines.
* If applicable, integrate accessibility testing into the continuous improvement/continuous delivery (CI/CD) process, and fail builds that do not pass accessibility tests.
* Utilize the [Yale UI Component Library](https://yale-a11y.gitlab.io/ui-component-library/) to help build accessible front ends:
	+ Pull in pre-built accessible assets to include in your project, if applicable.
	+ If it is not possible to include the assets in your project, at least refer to the component patterns as guidance for developing your own interfaces.
	+ Refer to the component library documentation for implementation details. Components that are built to be accessible can still be implemented in a way that is not accessible if the guidance is not followed.
	+ Implement integration tests to verify accessible components maintain their accessibility when implemented on a page.
* For native mobile application development, refer to specific guidance based on the platform (listed below). It is best to develop using the native platform SDKs (software development kits) rather than frameworks like React Native or Flutter. If a platform like one of these must be used, React Native is recommended. Redacted comment about Flutter limitations.
	+ [Accessibility on iOS](https://developer.apple.com/accessibility/ios/)
	+ [Android accessibility](https://developer.android.com/guide/topics/ui/accessibility)
	+ [React Native accessibility](https://reactnative.dev/docs/accessibility)
	+ [Flutter accessibility](https://flutter.dev/docs/development/accessibility-and-localization/accessibility)
* If a project does not have dedicated QA (including usability testing), developers should conduct QA testing of their builds. Refer to the **Testing/QA** section below.

## Testing/QA

* Expert review is the minimum required technique for validating adherence to this standard.
* Quality assurance analysts should complete the “QA” track of accessibility training offered by the ITS Digital Accessibility team.
* QA analysts should use a combination of tools and techniques to validate the accessibility of builds, including:
	+ Browser extensions to conduct automated and guided manual tests, as outlined in the **Develop** section above.
	+ Keyboard-only testing, including confirmation that pages minimally:
		- Include a visible “skip to content” link if there are more than 4 main navigation items
		- All interactive controls can be operated by the keyboard, include a focus ring that conforms with WCAG requirements, and focus is not lost when moving from item to item
		- Focus order follows the visible order of items on the page
	+ Testing with the following assistive technology and browser combinations, in order of importance:
		- NVDA + Firefox or VoiceOver + Safari (macOS)
		- JAWS + Chrome
		- VoiceOver + Safari (iOS) and TalkBack + Chrome (Android)
		- It is particularly important to use assistive technology to test custom widgets. Other testing techniques should uncover problems that will impact most assistive technology (for example, proper text labels will prevent issues with navigation using speech recognition software).
	+ Manual inspection of code.
* All pages should include an “Accessibility at Yale” link in the footer pointing to <https://usability.yale.edu/web-accessibility/accessibility-yale>, as outlined in the [Accessibility Statement Page](https://usability.yale.edu/web-accessibility/accessibility-yale/accessibility-statement-page) and required by the [Web Accessibility Policy](https://your.yale.edu/policies-procedures/policies/1605-web-accessibility-policy).
* Public websites (anonymously accessible) should be added to [Yale’s instance of Siteimprove](https://my2.siteimprove.com/Auth/Saml2/66356571), and QA analysts should verify sites obtain an accessibility score of 85% or greater.
* In addition to expert review, it is preferred that user acceptance testing be performed by people with disabilities. Third-party accessibility firms are best suited for this type of testing. Redacted has been used in the past on ITS projects. A recent option that is growing in popularity is Redacted. Contact accessibility@yale.edu for additional suggestions.

## Documentation

* Anyone responsible for creating documentation (content on websites, Microsoft Office documents, or PDF documents) should complete the “content editors” track of accessibility training offered by the ITS Digital Accessibility team.
* All Microsoft Office document authors should use the “Check Accessibility” tool integrated into most Office products, and correct any errors identified.
* Tools used to create PDF documents should export PDF/UA-conformant PDF files containing proper tags. Refer to the [Tagged PDF Best Practice Guide: Syntax](https://www.pdfa.org/wp-content/uploads/2019/06/TaggedPDFBestPracticeGuideSyntax.pdf) for guidance on proper tag structure.
* Refer to the [PDFs & Documents article](https://usability.yale.edu/web-accessibility/articles/pdfs-documents) for further information.

## Change Management

* Professionals supporting change management for projects should complete the “project managers” track of accessibility training offered by the ITS Digital Accessibility team, along with the “content editors” track if they are involved in creating digital communications for a project.
* All digital content and communications created to support change management for projects should conform to the Standard. Examples include:
	+ Technical documents should be accessible.
	+ Electronic training platforms and materials should be accessible.
	+ Email communications should be accessible.
	+ Videos should be captioned.
* Social media posts should be accessible (alt text for images, captions for videos).

# Engagement and Support

## Available Support

Support is available through the ITS Digital Accessibility team and on the [Usability & Web Accessibility website](https://usability.yale.edu/web-accessibility). The site offers guidance on achieving this standard from multiple perspectives, including developers, designers, and content creators. It also recommends specific scanning tools and checklists to assist with accessibility review. The team can also assess the expertise of those being hired to configure, build, and review ICT for accessibility.

## Roles and Responsibilities

While service and application owners hold responsibility for the accessibility of their ICT, the ITS Digital Accessibility team will provide hands-on support and consultation to assist in this process. Request support from the team at accessibility@yale.edu.

## Helpful Links

* [On-Demand Accessibility Trainings](https://usability.yale.edu/web-accessibility/training)
* [Articles on best practices and testing methods](https://usability.yale.edu/web-accessibility/articles)
* [An interactive outline of the WCAG guidelines](https://usability.yale.edu/web-accessibility/articles/wcag2-checklist)
* [Resources for Working with Vendors](https://usability.yale.edu/web-accessibility/working-vendors)
* [Yale UI Component Library](https://yale-a11y.gitlab.io/ui-component-library/)