VA SECTION 508

508 XPress

Hop aboard!

The 508 XPress is rolling into your station now.

VOLUME 17 | SUMMER 2016

Meet A Member of the 508 Team: Dan Diddle

"I care about people. I'm here to help them. If I don't have the answer, I'll find it." Dan began an interview for this article with those words, which sum up his commitment to his work in the Section 508 Office. Dan works with project teams to make their applications conformant with the accessibility provisions of Section 508.

"Section 508 is an awesome law. I want others to know about it." Dan didn't know anything about Section 508 when he first came on board. I give Rick Melton the credit for getting me started. He told me to read the 508 law and provisions until they made sense to me. At first they were really dry. But then I started sitting down with project teams, hearing their questions and helping them find answers. Now I've worked on applications that start with only twenty percent conformance, and they've worked their way up to one hundred percent."

He has had the opportunity to see firsthand how important Section 508 and accessibility can be. "I met a woman who is blind on the train one day. I struck up a conversation and asked her if she used a screen reader. She said she knew about them but didn't have one at work. When I found out she worked at the VA, I helped her get a copy of the JAWS screen reader. Now she is not only able to do her job, but she also lets our team know when she's having trouble using an application."

Dan's path to the Section 508 Office is an interesting one. He joined the Air Force in 1983. "I thought they were sending me to the State Department, but that's because I was a hillbilly from Ohio," he joked. "They put me into a career field dealing with languages. I loved learning about cultures and people. Most of my work was in Intelligence. I learned Mandarin. I had already done mission work for two years in Brazil, so I had learned Portuguese. And I had taken Spanish in high school. I became an Airborne Cryptologic Linguist."

Dan says one of his proudest achievements was receiving the 2005 Jerome F. O'Malley Award which recognizes the best USAF Reconnaissance Crew. On 8 June 2004, a Senior Scout crew departed Karshi-Khanaba Air Base, Uzbekistan, in route to provinces located in south-central Afghanistan. When the crew of which he was a member discovered the 22nd Marine Expeditionary Unit surrounded by 120 insurgents, Scout operators quickly and efficiently passed critical intelligence to the marines. "Within a short time of being made aware of the danger, (Senior Scout) provided targets for tactical strikes. Five Marines sustained injuries, but because of the quick, efficient work of the Senior Scout crew, they all went home. When all come home alive, we have done our job!"

INSIDE THIS ISSUE

After serving for thirty-two years, first on active duty, then in the Air Force Reserves and in the National Guard, Dan was happy to take a job that would allow him to be at home in Utah with his wife and six kids. "My family is the most important thing in my life," he says.

"My programming background is the result of my work with the United States Olympic Ski Team. They needed a database manager. They wanted to collect data to figure out what would make the perfect Olympian athlete. I developed a database to measure everything from how far athletes walked to how much they ate and drank, to how much they exercised. The measurements could get about ninety per cent of the information that was needed. But they couldn't measure the internal drive that makes someone a perfect athlete."

If he worked for the Olympic Ski Team, he must love skiing, right?

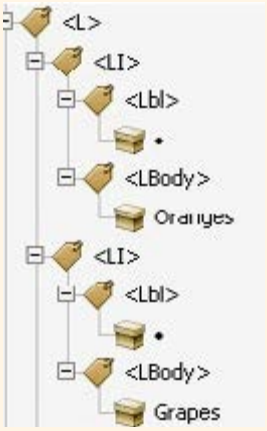
"Well, no. They told me one day they needed me to score a race at the top of a mountain. That's when I told them I didn't know how to ski. They gave me a couple of lessons, just enough to teach me to snowplow. They took me to the top of the mountain on a snow mobile and dropped me off. But after the race, nobody came back to pick me up. I had to snowplow all the way down the mountain – and I've loved skiing ever since."

His programming background gives him the ability to help developers with their 508-related questions. "I want people to understand that they can do a lot of their 508 testing themselves. They can start with a keyboard and make sure they can use it to do everything in their application. And they can use a screen reader to find out if everything on the screen is being read. Those tests won't catch everything, but they will catch a lot of the 508 issues that need to be fixed."

He offered one last piece of advice. "Don't be afraid to ask. If you have a question, send it to section508@va.gov. Talk to us. We really can help."

Quick Tip: Proper PDF List Structure is All in the Family

It is important that related list items be structured as a list to allow AT users to efficiently navigate the content. List structures help assistive technology users to identify and understand the relationship items have to each other.



Think of the structure of lists and list items as a family tree. There are the parents, which are the main list items. Then there are the children, or sub-lists, under one parent item. All list items contained at the same level can be thought of as siblings. In Adobe Acrobat, each list structure must consist of a parent <L> tag and subsequent list item <LI> tags for each item in the list. If there are a total of three fruits listed, then there needs to be three <LI> tags in the structure. Each list item <LI> tag must contain a label <Lbl> tag that includes the number or bullet and a list item body <LBody> tag that contains the text of the item.

Heed lists that span across multiple pages. Even though a list continues on to another page, it is important that in the tag structure those list items are part of the original list and not a list of their own. Attention to detail of these items enables users to associate related content and know the total number of related items.

Get Onboard!

It is now possible to be alerted when a new edition of the 508 XPress becomes available. Just visit www.section508.va.gov/support/newsletter and activate the link to subscribe to our list.

Visit the VA Section 508 website to review Section 508 checklists; training materials for developing accessible content in Flash, HTML, Word, PDF and PowerPoint; register for courses and to locate additional resources.

Internet: www.section508.va.gov || Intranet: vaww.section508.va.gov*

Links designated with an asterisk are available to VA internal users only.

Plain Language Provisions: Text Equivalents

This is the second in a series of articles called *Plain Language Provisions*, where we will explain some of the commonly misunderstood Section 508 requirements. The requirements are part of the Section 508 standards, which are available on the United States Access Board website. The requirements are listed in lettered paragraphs under Subpart B – Technical Standards, Subpart C – Functional Performance Criteria, and Subpart D – Information, Documentation and Support. They are called provisions because, legally speaking, they represent an agreement between a federal agency and the people or organizations responsible for the Electronic and Information Technology (EIT) that it develops, procures, maintains, or uses.

Provision §1194.22(a) is one of the most important in making EIT accessible, and it states: **A text equivalent for every non-text element shall be provided**.

Non-text elements are any part of a screen that is not text. This includes, but is not limited to:

- controls (like links, buttons, drop-down menus, and checkboxes)
- images, pictures, and graphics
- images used as links or other controls
- maps
- charts, diagrams, and decision trees
- videos
- podcasts
- multimedia

A text equivalent means that all information represented visually must also be available in text. That text can be visible on screen for all users, or it can be hidden. Whether visible or hidden, the text representation of visuals must be either programmatically associated with or very near the thing(s) it is describing.

Plainly, this provision states: **Whenever information or meaning is communicated visually, also communicate that information or meaning in text.**

SUGGESTED TECHNIQUES FOR PROVIDING TEXT EQUIVALENTS

Using a label is a great way to provide a text equivalent for controls, as discussed in the article on user interface elements in the Spring edition of the 508 XPress.

- For podcasts and other audio-only portions of a screen, a transcript is the best text equivalent.
- For a video or animation with no sound, write a description of what is happening in the content. Write out all of the important information displayed in the content.
- Video, animation, or other multimedia must use captions as the text equivalent. A transcript alone will not suffice. This is reinforced by Section 508 provisions §1194.22(b) and §1194.24(c), which might be the subjects of future articles in this series.
- For still images, consider using alt text. For still images that cannot be sufficiently described in a sentence or two, treat it like a complex image.
- Maps, charts, diagrams, and decision trees should be treated like a complex image.

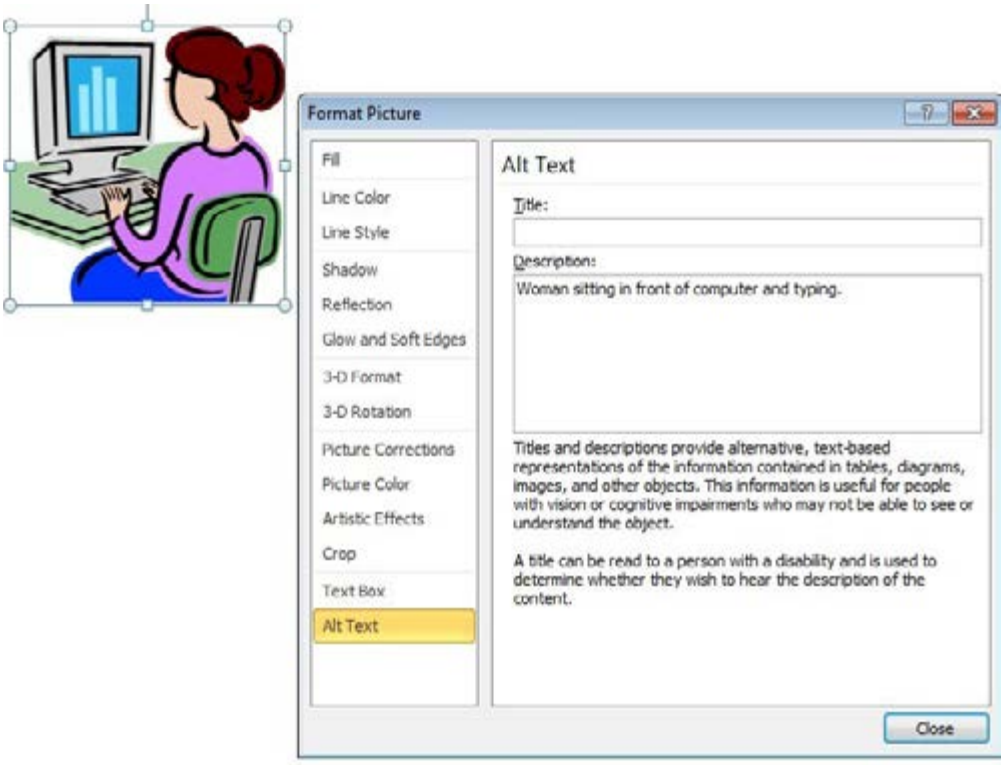
WHAT IS ALT TEXT?

As discussed in the previous article, screen readers can read two things about most user interface elements: structure, and text. For applications capable of producing accessible documents (like Microsoft Word and Adobe Acrobat), structure is included by using the functions built into the application. For instance, a screen reader would know the following example below is an image because it was inserted using the Insert > Picture command. (Different screen readers might announce “image” or “graphic” or “picture,” but the technical structure is the same). In HTML, structure is determined by the role of the element. For instance, a screen reader might know that something on a webpage is a graphic because that graphic uses the <img> tag.

Although a screen reader can announce the structure of the image, it cannot describe what the image is about because there is no text associated with it. That is where alt text comes in. Alt text, short for alternate or alternative text, is text that is used to describe it and programmatically associated with it.

The example below comes from the VA tutorial on Creating Accessible Word Documents. A woman is sitting at the computer. By adding the alt text of “woman sitting in front of a computer and typing,” a screen reader will

*Links designated with an asterisk are available to VA internal users only.



announce "Graphic. Woman sitting in front of the computer and typing."

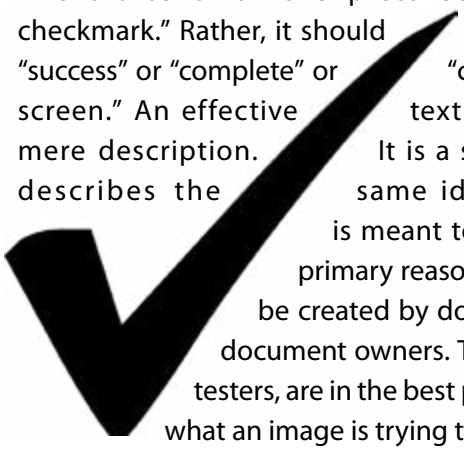
Notice that when writing alt text, it is rarely advisable to include phrases like "image of …" or "picture of …". Since a screen reader automatically announces the structure of "Graphic," including the words, "image of" is redundant.

Techniques for adding alt text

- In Microsoft Office 2010, you can add alt text to any picture by bringing up the Context Menu (right click the image, or press shift+F10 on your Windows keyboard) and selecting Format Picture. In the Format Picture dialog box, select alt text. Whatever you write in the Description box will be automatically read out loud by assistive technology such as screen readers and screen magnifiers with speech output.
- Do not use the Title field in Microsoft Office 2010. It is not well supported (meaning, users of assistive technology may not be able to retrieve the information you put in). Furthermore, there are some known defects. One popular screen reader will sometimes read "Slash" when an image in Office has a Title, regardless of what information that you enter.
- In HTML, you add alt text with the alt attribute. For example, <img alt="woman sitting in front of computer and typing."> Did you know that valid HTML requires the alt attribute on every image?
- In Adobe Acrobat XI, you can add alt text by selecting View > Tools > Accessibility > Set Alternate Text.
- People generally like alt text because it is easy and well supported. Perhaps just as important, it does not interfere with your visual design. Alt text is not printed on the page; it is typically only available to people using assistive technology.
- Alt text must be short. Try for no more than a few words, or a sentence or two at most. Some user agents (web browsers, document viewers, assistive technology, etc.) will stop presenting the alt text after a sentence or two. Perhaps more importantly, alt text is announced immediately and automatically after reading an element's structure. Unlike text written on a page, alt text can't be stopped, started, and reviewed easily by a user.

What It Does or What It Means is More Important Than What It Is

Alt text should not necessarily describe what something literally is. Rather, the alt text should describe the meaning, purpose, or function of an image. Consider the following checkmark.



The alt text for the picture should not be "big checkmark." Rather, it should be "correct!" or "success" or "complete" or "continue to the next screen." An effective text equivalent is not a mere description. It is a series of words that describes the same idea that the image is meant to convey. This is the primary reason that alt text should be created by document authors and document owners. They, not Section 508 testers, are in the best position to determine what an image is trying to say.

TOO BIG FOR ALT TEXT?

Alt text is a great technique for providing text equivalents of simple, static images, but what if the meaning of the image you want to describe is too detailed for a sentence or two? In that case, the text equivalent must be printed in the document itself, or available through an easy-to-find link. Consider this example from the World Wide Web Consortium (W3C). The bar chart has an alt text of "Bar chart showing monthly and total visitors for 2014 for sites 1 to 3." That is a decent alt text insofar as it describes what the chart is, but it neglects the most important meaning conveyed by the chart: the information and data inside of it! That information much too big for alt text, so the author provided a link right next to the image. Following that link presents users with a table. Inside the table is the same data that is in the graph.

If you know that your readers will have Internet access, links are a well-supported way to provide text equivalents. If you are unsure whether your readers will be online while reading your graphic, include the text description in the document itself – using a separate heading or an appendix, for example. Last but not least, you could use elements that are programmatically associated (that is to say, automatically connected) to their elements – like figure or table captions, which are available in HTML, PDF, and Microsoft Office.

More Resources

Providing text equivalents is one of the most important (and most thoughtful) tasks in all of accessibility. Like most 508 compliance tasks, it is a mix of art and science. Your job when providing text equivalents is to describe the essence of something, and to make sure that the description is available. You can test alt text by using any of the tools available from VA Section 508, or by looking in your document itself -- either the source code of a website, or by using the same functions within your application that enable you to write alt text in the first place. For additional information, consider: the VA Section 508 training on creating accessible documents; this excellent tutorial from WebAIM; or the guide from the W3C.

Exploring A Possibility Of Sharing 508 Conformant Mobile Components

Over the past 3 years the VA Section 508 Office has been fortunate to play a part in the VA's mobile development process. Various development teams have embraced the necessity for 508 conformance as it develops mobile websites and apps to best serve its Veterans and employees who use mobile tablets and smartphones. As a result of the 508 Office's interactions with a number of different development teams, it has been able to further tweak and refine its agile test process to address many accessibility considerations when testing mobile web content, native iOS/Android apps, and Hybrid apps. The Office has learned many lessons, and has played a role in providing a great deal of guidance to a number of developers.

One of the frequent trends that occurs is the constant influx of new developers that are assigned to work on various mobile projects. These developers often are assigned to new mobile development efforts as well as to sustain or update mobile content that has been previously certified. In either case, these developers are often new to Section 508, and sometimes, the 508 regulations and their impact on mobile environments are not at the forefront of these developers thinking. To that end, the 508 Office is presented with the challenge of remediating a number of violations that repeatedly show up when new mobile content is audited.

We've put a great deal of thought into learning from past lessons while accelerating our ability to test and certify content. The most valuable lesson that has been learned is that many development teams are faced with similar violations across mobile projects when developing content for the afore-mentioned platforms. We've discovered that when code is shared with our subject matter experts, we're able to suggest alterations, reach out to the manufacturers of these mobile operating systems, and overcome the challenges presented by operating systems that are updated along with their respective browsers.

The Section 508 Office is exploring ways to engage developers to share their code in order that all VA stakeholders may benefit from the strategies and methods by which mobile Section 508 conformance is achieved. To reiterate, many development teams tend to make similar mistakes when coding as well as run into similar obstacles presented by the constant evolution of mobile assistive technologies. The intent of this article is to introduce its readers to the concept of a "component library" which is an industry standard vehicle for sharing code amongst all parties who share the same mission of a given agency or organization. A component is an independent piece of code that can be reused in other development efforts. For instance, a piece of code created to resolve a focus issue in a particular environment can be reused in another effort that mirrors the previous environment from which the component was used. The creation of a component library for sharing content would provide a means for all VA developers to assist one another as they strive to bring their mobile content into conformance in a timely, cost-effective manner.

There are several reasons why sharing mobile components with the Section 508 office and making them available to other developers is important. Most obviously, if a known violation is remediated for one project, this remediation can easily be applied across various development efforts. Why recreate the wheel to find a solution to a problem that's already been solved? And why make the same mistakes more than once?

Secondly, the utilization of shared components amongst mobile development efforts funded by the same stakeholder can give a more common look, feel, and usability to the different apps developed within a given development project. Assume that a suite of mobile apps are being developed to best serve our Veterans, and that this effort is being managed by a single department? Sharing components across these apps will not only more efficiently remediate 508 violations, but may also give to the entire project a uniform accessibility theme providing a seamless and intuitive experience for users who rely on assistive technology.

Thirdly, sharing components amongst development teams can serve as the brainchild for future development efforts and cement ongoing partnerships. If products are being developed quickly and efficiently, and are not spending lots of time being remediated. More projects can be started and completed with more people being served.

Lastly, the components being developed for VA stakeholders belong to the VA, and their development is being funded by VA dollars. It only makes sense that if all development teams and all compliance bodies can leverage off these shared components for current and future development efforts, time will be saved, dollars will be maximized, and more Veterans and employees will have access to more content in a conformant manner. While it's comfortable to "not" share, and to operate in isolation, progress will accelerate and the VA's rate at which 508 conformance is achieved will only be expedited when development teams and compliance testers work closely together to create a means by which code is readily shared, and compliance is more efficiently achieved.

In summary, reusable components, component-based development, provides:

1. Reuse: components are reuseable across many platforms, "build once, use many"
2. Central Management: components are standardized on agency requirements, branding and code standards that are shared across the development platforms. Components and code can be updated and added as needed, immediately available to developers.
3. Dynamic Construction: developers, UI developers program managers can pick from components that are unique to their application.
4. Standardization: components have a standard look, behaviors, accessibility, metadata, etc.

If you have questions or ideas as to how to best share or benefit from the ideas expressed in this article, please email the mobile testing mailgroup at Section508mobile@va.gov.

**Links designated with an asterisk are available to VA internal users only.*