



THE 508 XPRESS IS ROLLING INTO YOUR STATION NOW. HOP ABOARD!

Meet a Member of Our Staff



Starting with this issue, the VHA Section 508 Office will be introducing you to a member of our staff. Our first interview is with Ellen Crowe, the Director of VHA Section 508.

Q: When did you join VA?

I began my career in March 1990 as a Wang System Administrator (SA) for the Regional Office and Insurance Center in Philadelphia right out of college. From there, I was detailed to the Benefits Delivery Center, and ultimately took a position as a National Help Desk Analyst providing support for Wang SAs across the country. A few years later, I moved to VHA and out to the Chicago area where I took a position as the supervisor and trainer of the Blind Rehabilitation MUMPS training program at the Hines Blind Rehabilitation Center. When that program lost its funding, I was detailed to the Hines Blind Center as a Computer Access Training Specialist, where I taught blinded Veterans to use assistive technology in conjunction with Microsoft Office applications for personal and professional use.

Q: What college degrees do you hold?

I have an Associate's in computer science from Peirce College in Philadelphia and a Bachelor's in liberal studies from West Chester University in Pennsylvania.

Q: How did the VHA Section 508 Office come to be?

After a few years of working at the Hines Blind Rehabilitation Center, I accepted a job with the Hines Information Resources Management (IRM) shop, where I put my MUMPS and VA Fileman training to use as a programmer supporting various entities of the Hines Medical Center with various VistA applications. While in this position, I provided some support as a collateral duty to the Computerized Patient Record System (CPRS) development team, offering insight and guidance to how the CPRS graphical user interface could be designed and made accessible. Section 508 was a relatively new law at the time, so when a position was announced for a 508 analyst I applied for it and was accepted. It quickly became evident that we needed a team of talented folks doing this type of work. Having just been aligned under Chief Information Officer (CIO) Craig Luigart in a reorganization, I appealed to

him for resources to build a team and the VHA Section 508 Office was born in 2006.

Q: What are the main objectives and authority for the VHA Section 508 Office?

In December, 2008, the office was granted authority for ensuring Section 508 compliance of VHA eLearning content and VHA Web Communications. This means we conduct accessibility testing of electronic content, provide guidance and resources for remediating Section 508 violations and grant certification once the content is free of violations. Certified content is then provided on the Talent Management System (TMS) or distributed through other electronic means.

In addition, the office takes a proactive approach by creating specialized training to assist developers in creating compliant products. Training resources are intended to help all staff consider 508 and accessibility of eLearning and web content early in its development.

Some other functions of our office include:

- Providing ad-hoc support to the field on specific Section 508 issues and requirements
- Maintaining Internet and intranet sites to better support VHA staff and contractors
- Assisting VHA with accessibility issues and solutions
- Supporting management requests to ensure products are 508 compliant
- Speaking at conferences both inside and outside the VA, sharing our expertise via presentations and staffing exhibit booths

(continued on page 2)

Inside This Issue	
Meet a Member of Our Staff	1
Thanks & Farewell	2
Resource Spotlight: Creating Accessible Content for Mobile Devices	2
Creative and Accessible	3
Quick Tip: Cautions on Using the Screen Reader Flag	3
Get Onboard!	3
Avoiding Common Violations	4
Designing and Developing with Accessibility in Mind	5
Captivate, Presenter, Camtasia and Accessibility	6
Feature Comparison Table	7

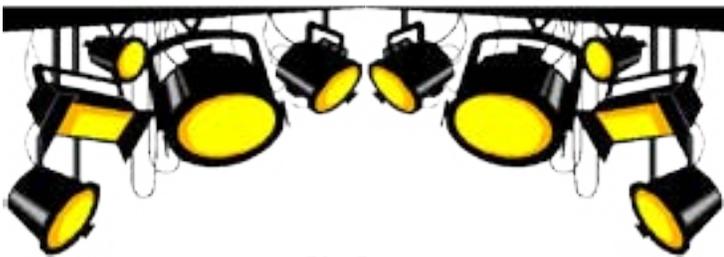
(continued from page 1)

Q: What specific training resources are available through the VHA Section 508 Office?

We have prepared a great deal of material to help our customers. We offer courses on creating accessible Flash and HTML content. We also offer tutorials on making accessible PDFs and Microsoft Word documents. Our website offers resources for developers and information about our testing procedures and contains the checklists we use in testing.

Q: Is there anything else you would like people to know about the office?

Yes. Section 508 is a federal regulation. It has a set of clearly defined accessibility requirements. Our office does not and cannot arbitrarily create accessibility standards. The requirements contain language that some may find confusing. We work to ensure that all requirements are met. We want to let electronic content developers know that we are here to help them navigate and interpret the language. Our philosophy is that we are partners in the process. Please reach out to us with any questions. The earlier we are consulted, the more likely testing and certification will occur in an efficient, cost-effective manner.



**Resource Spotlight:
Creating Accessible Content for
Mobile Devices**

Section 508 applies to mobile technologies. Below are some resources for developing accessible mobile apps and web sites:

APPLE IOS ACCESSIBILITY

- [The Mobile Accessibility Landscape](#)
- [Make Your iOS App Accessible with VoiceOver](#)
- [Accessibility for iPhone and iPad apps - Matt Gemmell](#)
- [A Video demonstration of iOS accessibility features](#)
- [Apple's Guide to accessibility](#)

GOOGLE ANDROID ACCESSIBILITY

- [Accessibility for Android Developers](#)
- [Google IO Making Android Accessible](#)

GENERAL

- [Web Content Accessibility and Mobile Web](#)

Thanks & Farewell

In June 2012, the VHA Section 508 Office welcomed its first intern when Garrett Kearns joined our office as a part-time employee. Garrett, who is visually impaired and began using a screen reader just a couple of years ago, was an asset to our office this summer. Due to his interest in writing, he became involved in editing tutorials being created by our office on making Microsoft Word documents accessible. He worked with our team to provide edits to make content clearer to those not as familiar with accessibility and Section 508.

He also spent time auditing our websites to improve his own knowledge and awareness of the breadth of Section 508 and accessibility best practices.

Garrett says accessibility is important to him personally because he now has to rely on a screen reading program to study educational material, research information on the Internet and maintain his connections on social media. He has experienced first-hand difficulties in using untagged PDFs in high school, which created stress because he was delayed in accessing required reading material.

Garrett says he now realizes that accessibility is not as complicated and daunting as people may think. He learned that if accessibility is considered early, everyone can benefit. Plus, there are many useful resources available to help people understand how to make their content accessible.

“People should care about accessibility because so many rely on electronic information. People who are visually impaired, hearing impaired, motor impaired, etc., all have different ways of using assistive technology, but they all have the same needs for accessing the content as everyone else.”
---Garrett Kearns

The rate of unemployment or underemployment for working-age persons with visual disabilities is much higher than for the population at large—over 70 percent by some estimates. Garrett is grateful to VA and the VHA Section 508 Office for giving him his first job experience.

In his spare time, Garrett plays classic rock on the electric guitar, reads science fiction and keeps up with his friends on a variety of social media sites.

We wish Garrett the best of luck and greatest success as he begins his studies in English at the University of Mary Washington in Fredericksburg, VA.

Creative and Accessible

As the rollout of Windows 7 with Office 2010 rapidly moves forward, the VHA Section 508 Office decided to examine the Office 2010 training on the VA TMS. The first course we looked at is a SkillSoft course, called "Getting Started with Word 2010." The course begins with a skill check in which the learner takes a quiz to determine how much he or she already knows about Word 2010. The first question is a matching exercise - a challenging type of quiz in terms of accessibility.

At the beginning of the exercise, the learner is asked to match options to targets. With the next press of the Tab key, the learner lands on the first option and is given precise instructions for completing the exercise using the keyboard, as follows:

Option A: Move the cursor to the beginning of the previous paragraph, not matched.

Drag and drop matching page, use CTRL-N to pick up, CTRL-D to drop.



When the keystroke to pick up an option is pressed, the learner then receives further instruction. The name of the option is spoken and the learner is told that the option has been picked up. The learner is further told to drop it on a target, Tab to the target and press CTRL-D.

The instructions are clear and concise. Support is provided at each step during the quiz. The learner using assistive technology can easily follow the instructions and complete the matching exercise.

Congratulations to SkillSoft for producing such a unique and accessible approach in the drag-and-drop matching exercise!



Get Onboard!

It is now possible to be alerted when a new edition of the 508 XPress becomes available. Just visit <http://www.ehealth.va.gov/508/newsletter/> and activate the link to Subscribe to Our List.

Visit the VHA Section 508 Web site to review Section 508 checklists; training materials for developing accessible content in Flash, HTML, Word, PDF and PowerPoint; and to locate additional resources.

Internet: www.ehealth.va.gov/508/

Intranet: vaww.vista.med.va.gov/508workgroup/*

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



Quick Tip: Cautions on Using the Screen Reader Flag

Web browsers and applications may have a toggle that can be set to detect whether a screen reader is running. This is commonly referred to as the "screen reader flag", and was designed to help adjust the behavior of a page to make it easier to use for screen reader users. One potential use for this would be to provide specific textual clarifying information that might be redundant or unimportant to other users.

However, there are occasions where the screen reader flag gets used inappropriately, often as a shortcut for an "accessibility mode." Accessibility and Section 508 compliance is about more than just screen readers, and in some cases, a feature that is enabled with the screen reader flag can be unusable to others who need it.. One example of this seen by the VHA Section 508 Office is the Plateau Quiz Editor tool called PQE. The screen reader flag is used to detect the presence of screen reading software and then enable keyboard accessibility for its quizzes. Although someone using a screen reader can access the quizzes, someone else who does not use a mouse, but also does not use a screen reader, cannot complete these same quizzes at all.

Best practice says it is better to design for overall accessibility, and only use something like the screen reader flag for special cases where it really is the best or only way to address an issue specific to screen reader access.

Avoiding Common Violations

LEARN HOW TO AVOID THESE ISSUES SO YOUR CONTENT WILL PASS ACCESSIBILITY TESTING

During the third quarter of fiscal year (FY) 2012, the most common violations our office helped remediate fell into one of two categories: ensure proper reading order and ensure list items are structured properly in PDFs.

ENSURE PROPER READING ORDER

Particular care must be taken to ensure that assistive technology conveys information to the user in the intended reading order. Here are considerations for providing content in Flash, HTML and PowerPoint:

FLASH

In Flash, reading order is the order in which elements within a Flash movie are presented by assistive technology. The reading order is defined based on the `tabIndex` attribute of an element. Thus, the `tabIndex` attribute of elements serves two purposes: 1) defines the keyboard tab order if the element is in the tab order, and 2) defines the reading order of elements that are not in the tab order. Assistive technologies, such as screen readers, provide access to these elements through a virtual representation of the page similar to a word processor. When an element has an action listener but does not have a `tabIndex`, the reading order is determined automatically in Flash, based on the X and Y coordinates of the registration point of each object. In Flash 8 and above, Flash will properly integrate elements with and without a `tabIndex` in a reasonable reading order. However, in Flash 7 when some elements contain `tabIndex` and others do not, Flash will automatically put all elements in an x/y reading order and ignore the `tabIndex` attribute. Flash 7.0's automatic reading order, however, does not always result in a reading experience similar to the expected reading order. Thus, it is important that all elements provide a unique `tabIndex` property.

The `tabIndex` property for all the elements can be set using ActionScript and updated dynamically as the tab order of the application changes. All objects that are not part of the reading order must have the silent field of their `_accProps` property set to "true." This is true of objects that are onstage, offstage or at any point in their life cycle are present in the movie. When rollover content appears after a user action, the content should appear later in the reading order (higher `tabIndex`) than the element that generated the rollover.

When pop-up layers are used to display dialogs and other content, inactive content should be silenced to ensure it is not seen by assistive technology. For example, all grayed-out, inactive content should have the silent property set to

"true" and movie clips that have accessible children should have the `forceSimple` property set to "true."

More information can be found in the [Controlling Reading Order and Tab Order](#) section of our Creating Accessible Flash course.

HTML

Reading order in web pages is similar to reading order in documents; it's about making sure that a screen reader user can read through the page in an order that makes sense. For example, the reader will not encounter buttons in the middle of an article when those buttons actually appear after the text, or in a place where someone looking at the page would think of them as after the text. Reading order is related to tab order but is not the same thing. Screen reader users don't use the tab to review text; they tend to use the "Tab" key to navigate through elements that are interactive. For many screen readers, reading order is determined by the order the text appears in the source. So, if you have a block of text that is styled to appear in the middle of a page, but is coded at the end, a screen reader will read that section at the end of the page, out of the intended reading order. The way to fix this is to move things around so that the flow in the source matches where it ought to be read, and then style it to fit the look you want.

POWERPOINT

The order in which objects are presented on PowerPoint slides is important because screen readers convey slide content based on various factors, not necessarily the way the objects appear visually. For example, the intended reading order may be the slide title, a list of bullet points, followed by a chart of data and a picture in the lower right corner. However, depending on the reading order, the screen reader may indicate the chart of data first, followed by bullet points with a picture containing alt text in their midst and the slide title indicated last.

The reading order is determined by the order in which the objects are layered on the slide, which is also called the Z-order. The Z-order will dictate the reading order of the objects to certain screen readers.

You can check the Z-order of the objects on your slides by tabbing through them while in Design View.

The tab order of the objects is the same as their Z-order from lowest to highest. If the tab does not follow your intended

(continued on page 5)

Designing and Developing with Accessibility in Mind



Software designers and developers who leave Section 508 accessibility considerations until the end of a project and then rely on testing to tell them what needs to be fixed often find themselves rushing to make the changes needed to meet the requirements. It doesn't have to be that way. If you start thinking about accessibility up front, while you're planning out your storyboards and

deciding on the look, feel and functionality of your product, there will be much less work to do at the end.

Think about this scenario: if you were an architect designing a building and knew that there were accessibility requirements for your structure, you would want to incorporate wheelchair ramps, elevators and doorways of the correct width into your design from the beginning. Putting them in after the building is finished would be time-consuming, expensive and difficult, and might even change the look and function of the building you originally created.

So, how can you start planning accessibility into your project from the beginning? Here are some concepts to think about:

- When planning the behavior of an element that responds to the mouse, either by being clicked on or hovered over, remember that it needs to respond to keyboard events as well – either a key press or getting keyboard focus when a user tabs to it.
- When considering the visual response of items on the screen – such as dynamically changing content, pop ups or dialog boxes – remember that someone may be using your product with screen reading or screen enlargement software. You need to make sure that the same dynamic information is also available to those users.
- When planning the look of pages, use “semantic markup” to designate lists and headings and other structures, and then apply visual styles to those items. That way, the purpose will be clear to everyone.
- When planning your color scheme, take contrast requirements into account and use a contrast checker to make sure that everything falls within good contrast ratios for accessibility.
- Wherever you use images, make sure you've provided alternate text for meaningful images and empty (“”) attributes for decorative images like space holders and borders.
- If your project has form elements make sure they are accessible by using standard controls whenever possible, labeling them properly and creating custom controls that are accessible.

Ask for help or advice! If you're not sure whether your idea can be made Section 508 compliant or how best to do it, just contact us. Some tools are better than others in producing accessible content. We can give you advice before you've committed yourself to one that might not meet your needs. We're here to help you plan effective approaches that may help enhance everyone's user experience while assuring compliance. The earlier you ask, the more possibilities may be available.

We invite you to explore our Resources page for more information on the topics mentioned in this article.

http://www.ehealth.va.gov/508/resources_508.html

(continued from page 4)

reading order, you can change the Z-order to make sure that the JAWS screen reader will read the slide correctly.

To change the Z-order choose the Selection Pane from the Arrange section of the Format Tab on the ribbon and use the Re-order buttons to set the reading order from bottom (read first) to top (read last).

For more detailed instructions on fixing the Z-order and making your slides Section 508 compliant, refer to the article “The Importance of Setting PowerPoint Z-order” in [Volume 03 of the 508 XPress newsletter](#) or by visiting the [Microsoft Office Products section on our website](#).*

ENSURE LIST ITEMS ARE STRUCTURED PROPERLY IN PDFS

Lists are used to group and organize related items. Lists can be ordered or unordered based on their content. Utilizing explicit list markup allows users who are visually impaired to rapidly browse and access list content and have access to the same visual structure and relationship of list items that are displayed visually using space for hierarchy structure.

Proper list container markup should be utilized using the list element (L), the list item element (LI), the label element (LBL), and the list body element (LBody). The label element should include the bullet or number of the list item and the (LBody) element should include the text of the individual list item. It is important to ensure that nested lists are structured properly in order for the nested structure to be indicated to users of assistive technology. A sub-list element should be included before the closing list item (LI) tag of the list item it is located under. Sub-lists should use the same markup as lists for their structure.

When a table of contents is used to provide a list of sections in large documents, the table of contents should be properly marked up as a table of contents list using the table of contents (TOC) tag, the table of contents item (TOCI) tag, the reference (Reference) tag for the text and page number, and the (NonStruct) tag for the “...” leader.

[Tutorials on making accessible PDFs](#) are available on our VHA Section 508 public website.

Captivate, Presenter, Camtasia and Accessibility

The VHA Section 508 Office has recently received a great deal of eLearning content produced with Captivate, Presenter and Camtasia. Some of these tools cause certain types of content to fail Section 508 accessibility testing. To help you plan with accessibility in mind and pass 508 testing more quickly and with less remediation, we provide known limitations and benefits to using Adobe Captivate 6, Adobe Presenter 8 and TechSmith Camtasia 8.

Several features of these screen capture, rapid development eLearning and video editing tools are discussed as they relate to creating content that is accessible to persons with disabilities using assistive technologies.

What are these tools used for?

Each of these three products allows for the creation of videos, along with tools to annotate the experience. Furthermore, they provide options to generate learning assessments.

All three programs primarily render the content in Adobe Flash and accessibility is exposed through Microsoft Active Accessibility (MSAA). HTML5 support is also available in two of the products to support mobile devices. Each product has a distinct primary purpose, as described below.

Captivate is used to:

- Rapidly create eLearning content and assessments that are either slide or video based
- Import content from PowerPoint
- Support SCORM and other standard compliant methods for sharing eLearning content
- Capture the screen either as full motion video or in demonstration mode

Presenter is used to:

- Generate presentations from PowerPoint
- Create engaging presentations that can be used in the context of other environments, such as Adobe Connect, learning management systems, etc.

Camtasia is used to:

- Produce screen recording
- Edit video

CAPTIVATE

Adobe Captivate is a standalone, feature-rich tool for creating eLearning content, including presentations. PowerPoint presentations can be imported as a source for the project, but are not required. Captivate includes built-in support for video and audio recording as well as screen capture. Basic audio and video editing support is provided. Closed-captions and audio can easily be added to content. Several quiz question types, such as multiple-choice, can be provided in an accessible format.

Captivate 6 provides support for rendering content in HTML5 with some support for the Accessible Rich Internet Applications (ARIA) specification. Note that the HTML5 support provides a modest amount of accessibility, but does not provide as much accessibility at this time as a Flash version.

CAPTIVATE TIP

- Authors should provide accessible names for all non-decorative content and ensure that "Enable for Accessibility" is checked in Project Settings before publishing.

PRESENTER

Adobe Presenter is a plug-in for PowerPoint and appears on the ribbon. It is not a standalone application. Presenter primarily supplements objects already available within PowerPoint and exposes the accessibility of items in PowerPoint, such as slide text and alt text of images. Presenter 8 provides video capture features allowing for full-motion video and screencasts. Presenter is not video editing software however, and the video capturing components are still maturing. Flash-based Interactions were added in version 8, but accessibility support for these features could not be verified.

The support for accessible quiz questions is very similar to that provided by Captivate. Several types of questions are exposed in an accessible fashion.

PRESENTER TIPS

- The options to "Pause After Each Animation" should be checked under Presenter Settings before publication to ensure that animation can be stepped through and read by users of screen readers. By default, slides auto-advance

which must be disabled on each slide (through the Slide Manager > Advance by User) unless narration is provided with the exact information on what is displayed on the slides.

- Presenter does not support closed-captioning. It is recommended that a transcript of the narration is provided in the Notes pane. However, a transcript may not be sufficient to meet Section 508 compliance requirements for synchronized videos.

CAMTASIA

TechSmith's Camtasia Studio is a feature-rich, screen-recording and video-editing software. The software's primary output is mp4 video format. A media player is provided to allow the content to be viewed on various devices. Quiz questions, as well as actionable callouts and table of contents markers, are provided through Flash or HTML5 and are needed to appear in mp4 video. By default, the Adobe Flash Player is loaded when viewing content on a computer, while the HTML5 player is loaded when viewing content on a mobile device, such as an iPad. Closed-captions can be added to the video. Tools are provided to assist in the process of creating captions including one that generates text captions by listening to the audio track.

Note: The Media Playback Bar and "Play" button do not expose accessibility information, although they are keyboard accessible.

Because Camtasia is for screen recording and video playback, there are few interactive elements. The author can add callouts to the video; however, they do not expose accessible text. Callouts with actions associated with them, as well as quiz items, are the primary interactive content available within Camtasia. The quiz questions do provide a basic level of accessibility, but there are limitations in the Flash and HTML5 players. Specifically, there are problems moving keyboard focus into the quiz area and the state of radio buttons, such as checked or unchecked, is not conveyed to assistive technology.

A table comparing some features is below. A [full feature comparison is available on our website.](#)

Captivate, Presenter, Camtasia Feature Comparison Table

ELEMENT	CAPTIVATE	PRESENTER	CAMTASIA
Slide Text imported from PowerPoint	<ul style="list-style-type: none"> › As text with accessible name › Bullets appear as characters; there is no list structure; certain bullet types are announced as odd symbols › Heading text is not indicated as heading but as first text 	<ul style="list-style-type: none"> › As text with accessible name › Bullets appear as characters; there is no list structure; certain bullet types are announced as odd symbols › Heading text is not indicated as heading but as first text 	<ul style="list-style-type: none"> › Content is not imported from PowerPoint; the screen, mouse, and audio, can be recorded from PowerPoint
Slide Graphics imported from PowerPoint	<ul style="list-style-type: none"> › Alternative text does not appear 	<ul style="list-style-type: none"> › Appears as an image with an accessible name of the alt text 	<ul style="list-style-type: none"> › Content is not imported from PowerPoint; the screen, mouse, and audio, can be recorded from PowerPoint
Slide Reading Order from PowerPoint	<ul style="list-style-type: none"> › Order follows the order of items in the selection pane of PowerPoint 	<ul style="list-style-type: none"> › Order follows the order of items in the selection pane of PowerPoint 	<ul style="list-style-type: none"> › Content is not imported from PowerPoint; the screen, mouse, and audio, can be recorded from PowerPoint
Notes from PowerPoint	<ul style="list-style-type: none"> › Can be added to slide accessibility text or used a captioning text- are not displayed 	<ul style="list-style-type: none"> › Displayed in the notes pane as accessible text 	<ul style="list-style-type: none"> › Can be imported as captions. Notes are also displayed on the slide in a callout - but not as accessible text
Slide Title	<ul style="list-style-type: none"> › As a graphic with accessible name › Defaults to PowerPoint heading if PowerPoint used 	<ul style="list-style-type: none"> › Appears as button in outline view › Not in tab order 	<ul style="list-style-type: none"> › Does not have slide titles other than quizzes
Slide Level Accessibility Text	<ul style="list-style-type: none"> › As a graphic with accessible name 	<ul style="list-style-type: none"> › No specific accessibility text can be added to the slide; text is taken from the PowerPoint slide 	<ul style="list-style-type: none"> › Does not have concept of slides other than quiz screens where the question does appear as accessible text
Slide Reading Order	<ul style="list-style-type: none"> › No way to specify › Based on screen location 	<ul style="list-style-type: none"> › Taken from the selection pane order in PowerPoint 	<ul style="list-style-type: none"> › Reading order for simple quiz questions, such as multiple choice, is correct and cannot be changed
Click Box Button	<ul style="list-style-type: none"> › Accessible button › Default name of "click box" (can be changed) 	<ul style="list-style-type: none"> › Play button provided as part of media playback › Button is keyboard accessible 	<ul style="list-style-type: none"> › Next button exists of quiz questions and has accessible name, role of button, and is keyboard accessible › Call out action buttons used for interactions are missing accessible names
Text Caption	<ul style="list-style-type: none"> › A graphic with accessible name of text caption 	<ul style="list-style-type: none"> › Use standard PowerPoint textbox to expose accessible text 	<ul style="list-style-type: none"> › Text is not exposed for accessibility
Rollover Caption, Rollover Image, Rollover Slidelet	<ul style="list-style-type: none"> › Generally keyboard accessible although sometimes not in the keyboard order for some reason (appears as button) › Expose accessible name for the hit area and caption › No keyboard shortcut available › Caption that appears is not announced automatically and user must look for it › Recommend use of audio in addition indicate content › Click boxes can be used in the same manner but do allow shortcut keystrokes and audio - use the hint feature for the roll over and success for the keyboard accessible trigger 	<ul style="list-style-type: none"> › PowerPoint does not directly support Rollover functionality. PowerPoint "Actions" can move to different slides, end the presentation, go to URI, etc. › Presenter Interactions are could not be tested to determine their level of accessibility 	<ul style="list-style-type: none"> › Callouts can be used to create rollovers › Callouts set to click to continue are in the keyboard tab order › No accessible name is provided for callouts
Inserted Images and Smart Shapes	<ul style="list-style-type: none"> › Graphic with accessible name 	<ul style="list-style-type: none"> › Alt text of PowerPoint images are provided as the accessible name 	<ul style="list-style-type: none"> › N/A
Highlight Box and Mouse Pointer/ Trail	<ul style="list-style-type: none"> › Not accessible 	<ul style="list-style-type: none"> › N/A 	<ul style="list-style-type: none"> › Not accessible