Designing Accessible E-Learning Using Articulate Storyline

An incomplete reference for meeting WCAG 2.0 Level AA standards

Last updated 01/25/16

Designing Accessible E-Learning Using Articulate Storyline

When we talk about accessible e-learning design we're talking about making your online courses useful to those with complete or incomplete sight impairments, those with hearing impairments, and those who won't be using a mouse to click through your course.

The two main set of standards governing web accessibility (WCAG 2.0 – an international standard, and the less stringent Section 508 – an American standard) also apply to online courses. As of the release of Storyline 2 update five, Articulate claims that Storyline is compliant with both standards, but much of that compliance is left to author control.

This guide is intended to be an incomplete reference for the main aspects of developing accessible e-learning to WCAG 2.0 Level AA standards, using Articulate Storyline as an authoring tool.

Caveats from the author of this guide, Fiona Macelli:

- I am not employed by or affiliated with the Articulate company in any way, nor do I profit from the distribution of this guide.
- This guide does attempt to be a comprehensive guide to achieving the WCAG 2.0 standards. As such, I do not assume any liabilities related to the compliance of courses you create.
- I do not identify myself as a person living with a disability, nor am I an accessibility expert.
- I personally feel that Articulate Storyline is at least as good a tool in the accessibility arena as its competitors, but that there is room for improvement. Where product deficiencies are noted, these are based on my experience and interpretation and may not be wholly accurate. The Articulate company has not endorsed this guide.
- I am happy to correct any mis-information or mis-representation contained in this guide. Please contact me via my Articulate Community profile (preferred), or my LinkedIn account.

Contents

Accessible Design – Quick Guide

Because e-learning is such a visual format for training, and because few sighted e-learning authors have access to screen reading tools for testing and experimentation, I have found that the most challenging aspect of creating accessible e-learning is designing with sight impairments in mind. Once you get yourself into the mindset of a user with sight limitations, the rest of your design work will go relatively easily.

Design issues related to accommodating sight impairments

1. Text may need to be read by a screen reader. You'll need to:

   - Indicate the order that each text box is to be read (tab order) and which text should not be read at all. Learn how to indicate the order that each text box is to be read. Learn how to remove objects' visibility to accessibility tools.
   - Minimize on-screen navigation options so that someone using a screen reader doesn't need to "go through" them on each screen. Learn how to minimize persistent on-screen navigation options.
   - Type the numbers into a list rather than using the numbered list feature in Storyline, since the numbers in a Storyline numbered list are not read by the screen reader. Avoid parenthesis to indicate a list – e.g. 1) or A). Indicate sub-lists (second level lists) with a numbering system such as 1.1, 1.2 ; or avoid sub-lists if possible. Ensure that the preceding text indicates that a list will follow. Learn how to create recognizable lists.
   - Avoid text that is visible only on mouse-over (a.k.a "hover" or "rollover") or use alt-text to indicate what will happen when the object is activated using a mouse-over command (the JAWS® screen reader command to activate a mouse-over is INSERT+CTRL+ENTER). Creating a similar effect using a button that activates a layer is preferable. Learn how to avoid interactivity that requires a mouse. Learn how to add alt-text.
   - Understand that text that is within an image file won't be read by a screen reader. As a best practice, place text within the e-learning tool and not within the image itself. This is helpful for accessibility but it's also helpful for making updates to the course in the future and for translating the course (if that's ever needed).
   - Avoid, or provide an alternative for, animated or timed text boxes (text that is not available for the full length of the slide, or appears/disappears without activation from the user). Screen readers only read text boxes when they are in-focus, so they'll be missed if they aren't tabbed to during the time they appear on screen.
   - Link to URLs using a button (can be clear so that it looks like normal hyperlinked text) instead of just inserting a hyperlink within text. This identifies the URL as a button by the screen reader and enables the link to be tabbed to and accessed via the keyboard. Learn how to create keyboard-accessible hyperlinks.
   - Plan to publish as Flash (not HTML 5 or mobile player) in order to have control over tab order (and screen reading order). Also note that Storyline only supports screen reader functionality on Internet Explorer 11 and later and with JAWS 16 and later. Learn how to publish in an accessible format.

   You don't need to:

   - Label interactive elements (such as buttons) – the screen reader will do that for you.
   - Worry about how the objects (buttons, text boxes, shapes, etc.) are named within the Storyline timeline.

2. Visuals and visual organizers may not be seen at all. You'll need to:

   - Label (add alt-text to) visuals that are important to the learning, so they are described by the screen reader. Keep alt-text concise and describe the contribution of the image to the course rather than the image itself. Don't add "picture of.." – screen reader will add "graphic" to the end of your description automatically. Learn how to add alt-text.
   - Remove the "visibility to accessibility tools" of graphics that are only decorative so that they aren't included in the tab sequence or read as "graphic" by the screen reader. Learn how to control objects' visibility to accessibility tools.

- Reconsider tables, charts or other visual organizers that rely on the visual organization of the text to be perceived correctly. Use the alt-text function to describe the visual organizer or ensure the surrounding page text describes the visual organizer well-enough that seeing it is optional.
- Provide equivalent text to support synchronized media (e.g. videos, software demonstrations, animations) that are unnarrated or essential information is not narrated. Equivalent text includes a description of visual events, and is not just a transcript of the narration. Learn how to create a full-text equivalent transcript.
- Either use buttons with text that indicates what the button is for (e.g. Next), or apply alt-text to label the function of buttons in a way that can be read by the screen reader. Learn how to add alt-text.
- Rephrase visual-based instructions like "click on the link in the left hand corner" or "click the red button", etc.

3. Text and visuals may just be difficult to see (but the user is not equipped with a screen reader). You'll need to plan for:
    - High contrast text and visuals with a contrast ratio of at least 4.5 to 1, unless the text size is large enough to compensate for lessened contrast (to a minimum of 3:1 at a font size of at least 14 pt bold or 18 pt regular). Learn how to check for adequate color contrast.
    - Ensure that the use of color to differentiate items in a set (for example, bars on a chart) are additionally differentiated by pattern, line-type or another non-color reference.
    - Player settings that enable the user to resize content using their browser, and a story size that will accommodate the majority of monitors. Learn how to enable users to resize text.

Design issues related to accommodating hearing impairments:

1. Provide synchronized closed captions (timed on-screen text that can be turned on and off) to accommodate users who can't hear your audio. Learn how to create closed captions.
2. Avoid audio-based instructions and cues (such as narrated instructions or quiz feedback that is only provided via a sound)
3. Provide a full text-equivalent of synchronized media to accommodate users who have both sight and hearing limitations. Learn how to create a full-text equivalent transcript.

Design issues related to accommodating fine motor impairments (users who will use a keyboard or other device to interact with the course instead of the mouse):

Note that those with significant visual impairments will also not be using a mouse.

1. Plan for a good tabbing experience. You'll need to:
    - Ensure that the tab order makes sense visually as well as conceptually (tab order controls both the access to on-screen objects via the keyboard and the order in which they are read by the screen reader). Learn how to adjust tab order / reading order.
    - Make non-interactive objects "invisible" to accessibility tools to remove them from the tab sequence. By default they are set to "visible" and will be tabbed through). Learn how to remove objects' visibility to accessibility tools.
2. Avoid interactivity that requires the mouse (drag and drop questions, rollovers). Learn how to avoid interactivity that requires a mouse.
3. Include instructions for keyboard users:
    - Users can press the Tab key to advance through slide elements, and press Shift-Tab to move backward in the sequence.
    - In general, use the space bar to activate items such as buttons, or to respond to quiz questions.
    - Some items (such as drop-down lists and the player menu) require use of the arrow keys.

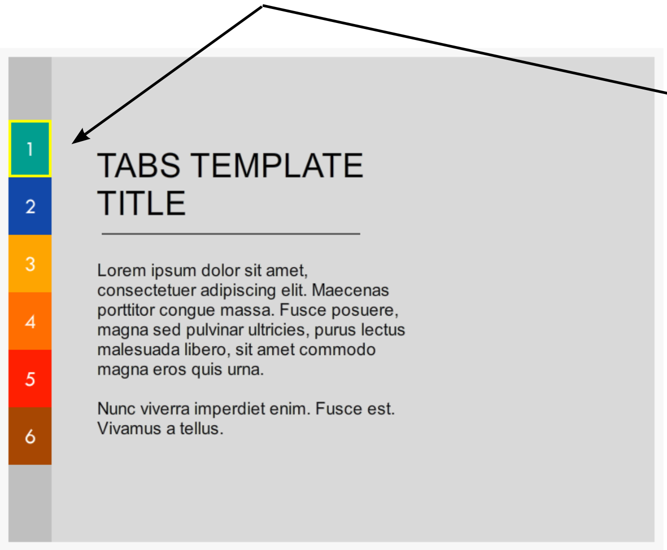
Indicating the Order That Each Text Box Is To Be Read

If you press the Tab key on a keyboard while viewing a previewed or published Storyline course, a yellow highlight appears around an object to indicate that it is "in-focus". If the in-focus object is one that can be interacted with (such as a button), you can interact with it by pressing the space bar. Continuing to press the Tab key cycles you through all the elements of the screen that are "visible" to accessibility tools.

A screen reader reads the text on a slide in the same sequence. This sequence is called Tab Order by Storyline. WCAG 2.0 uses a comparable term: Focus Order.

By default, the tab order (and therefore also the screen reading order) is from top-left to bottom-right on any visible layer of the slide, and then through the player itself.

The yellow highlight shows the part of the screen that is "in-focus" and can be interacted with by pressing the space bar.



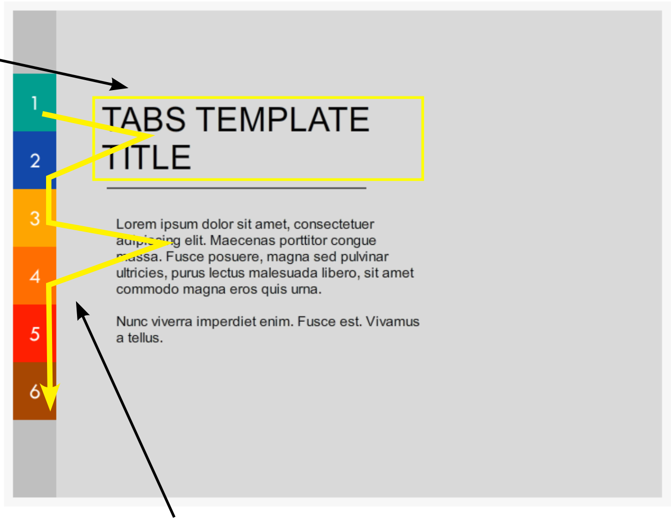
1
2
3
4
5
6

TABS TEMPLATE TITLE

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

Color Block Tabs Template is a free download distributed by Storyline.



1
2
3
4
5
6

TABS TEMPLATE TITLE

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

By default, a screen reader will read from top left to bottom right, following the default tab order. This isn't always desirable. In this example, the screen would be read "one", "Tabs template title", "two", "three", "Lorem ipsum...", "four", "five", "six".

Storyline's article Customizing the Tab Order of Slide Objects provides instructions for how to manipulate the tab order (and therefore the reading order).

This article also explains how to **remove objects from the tab order so they are not read** by the screen reader. This is also an important design consideration. As you build your course, consider how each slide will be perceived through a screen reader and remove or reorganize objects accordingly within the tab order. Learn more about changing objects' visibility to accessibility tools.

Note: Even with changes to the reading order, there are several problems with the Tabs Template example layout from an accessibility perspective.

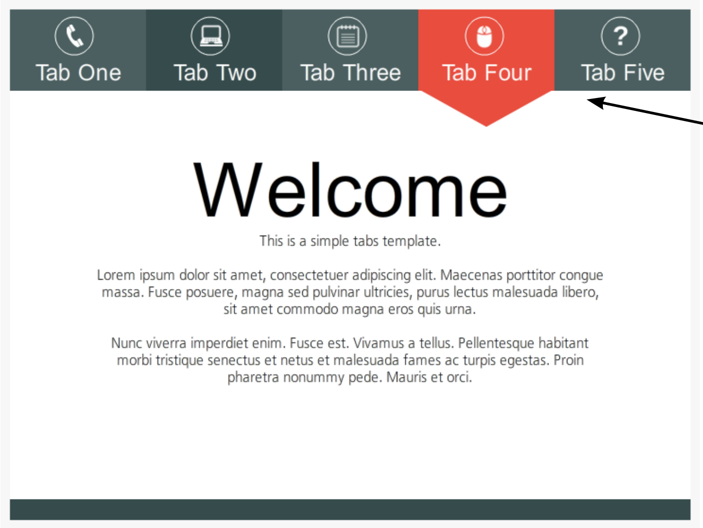
Consider:

- Tabs as an information organizer are only relevant from a sighted user's perspective.
- Even when read in the correct order by a screen reader, the above example wouldn't make much sense to a blind user. One, two, three.. what? Are they sections? Some kind of a list?
- Some of the tab colors don't provide enough contrast to the foreground text color (e.g. tab #3 – white on yellow) and would be difficult for someone with sight limitations, but no screen reader, to perceive.

What changes could you make to this layout to make it more accessible without reducing its functionality and attractiveness to your sighted audience?

Minimizing Persistent On-Screen Navigation Options

A user who relies on a screen reader is only able to interact with the parts of the course that the screen reader reads to him/her (objects that are set to be "visible to accessibility tools"). While a sighted user might automatically scan for and ignore elements such as navigation that are present on each slide, a screen reader must read that text on every slide.



Welcome

This is a simple tabs template.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

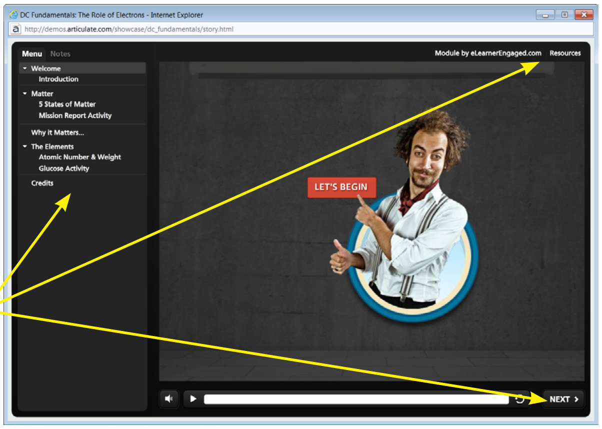
If these navigation buttons are available to the sight-limited user on every screen, they will be read by the screen reader on every screen.

Chunky Tab Template is a free download distributed by Storyline.

Although a screen reader reads quite fast, you can imagine that it would be annoying to listen to "*Tab one, tab two, tab three, tab four, tab five*" on every slide of the course, and that it would disrupt the learning experience in a way that it may not for a sighted user. A course designed this way will not be compliant with accessibility standards such as WCAG 2.0. A criterion of WCAG section 2.4.1 Bypass Blocks is that the user be able to skip past persistent elements, such as navigation, that are repeated on multiple slides.

Solutions:

1. As of Articulate Storyline 2 Update 5 (a software update that introduced a number of accessibility features), navigation contained within the Storyline **player** can be skipped to access the screen content immediately. Therefore, keeping persistent navigation options off the screen and within the player will meet the WCAG section 2.4.1 Bypass Blocks criterion.



As of Storyline 2 Update 5, navigation text within the player can be skipped by a user with a screen reader.

Example course credit: eLearnerEngaged
From the Storyline Showcase

2. If the player is providing other navigation options (besides Next and Previous) for getting around the course, then you can have on-screen navigation elements available to sighted users only, by making that text invisible to accessibility tools. Turning off the visibility will mean that this text will not be read by a screen reader, but it also means that it will not be keyboard-accessible.

   Note that WCAG section 2.4.5 Multiple Ways requires that the user be provided with more than one way to access slides via accessibility tools, so you'll need a keyboard-accessible method other than the Next and Previous button to get around in order to be compliant.

3. Whether navigation text is contained within the player or not, keep it concise. "Main Menu" is better than "Return to the main menu", especially when you're hearing it often.

Creating Recognizable Lists

Storyline doesn't programmatically indicate to the screen reader that a text box belongs to a list (a requirement of WCAG 2.0), and sub-lists are not indicated at all. The numbers made using the numbered list feature in Storyline are also not read by the JAWS screen reader (as of Storyline 2 Update 5).

For example:

List example	Read by the JAWS screen reader as	Effectiveness
Pack the following items for your trip: • Shoes ◦ Casual ◦ Formal • Computer • Notebook	"Pack the following items for your trip colon shoes casual formal computer notebook."	Inadequate
Where the numbering is made using Storyline's numbered list features: Pack the following items for your trip: 1. Shoes a) Casual b) Formal 2. Computer 3. Notebook	"Pack the following items for your trip colon shoes casual formal computer notebook."	Inadequate
(Where the numbering is entered manually) Pack the following items for your trip: 1. Shoes a) Casual b) Formal 2. Computer 3. Notebook	"Pack the following items for your trip colon One shoes a closed parenthesis casual b-closed parenthesis formal two computer three notebook."	Better. Parenthesis are read. Letters interpreted as sounds rather than list indicators.
Where the numbering is entered manually: Pack the following items for your trip: 1. Shoes 1.1 Casual 1.2 Formal 2. Computer 3. Notebook	"Pack the following items for your trip colon One shoes one point one casual one point two formal two computer three notebook."	Good
Where the numbering is entered manually: Pack the following three items for your trip: 1. Shoes, both casual and formal 2. Computer 3. Notebook	"Pack the following three items for your trip colon One shoes both casual and formal two computer three notebook."	Best

Adding Alt-Text (Alternative Text)

Use alt-text to add descriptive text to an image, chart, graphical button, etc. The screen reader will read the alt-text when the user tabs to the object instead of only reading the default: "graphic".

The Storyline article Adding Alternate Text for Screen Readers describes how to add or change alt-text.

Many resources give advice about writing appropriate and effective alt-text because it is a common concept in web-design. For this topic and others related to web accessibility, WebAIM is a good place to start your research.

http://webaim.org/techniques/alttext/

Keep in mind:

- Only add alt-text to functional images (not decorative ones). Decorative images, lines, shapes, etc should be removed from the tab order (made invisible to accessibility tools).
- If you add alt-text to an object that already has text (such as a text box or a button with text), the screen reader will read the alt-text in place of the object's text. This means that alt-text is not a good way to add additional meaning to text boxes, for example: to indicate that a text box contains a list.
  An exception to this rule is Storyline's "marker". You can add alt-text to the marker's button to instruct the visually impaired user about what to expect if they activate the marker (by pressing space bar), and still add text to the marker's text box. The screen reader will read the marker's text box when it is tabbed to.
- In Storyline, characters include default descriptive text that describes the character's pose. This description is read by the screen reader and, to my knowledge, cannot be edited. Adding alt-text to the character image does not replace this default text. If the user does not need to interact with the character object, and its function is mostly decorative, the character can be removed from the tab order (made invisible to accessibility tools), in which case nothing will be read by the screen reader.

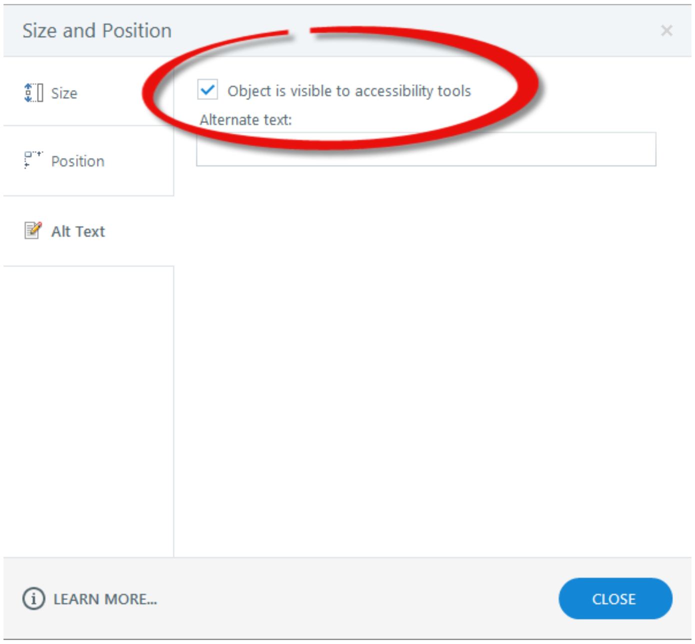
Changing Objects' Visibility to Accessibility Tools

By default, all Storyline objects are set to be "visible" to accessibility tools. This is not always desirable, for example in the case of decorative elements. Many of Storyline's default settings and master slides (for example, feedback messages) have all of the decorative elements "visible", when they should be invisible.

Why does it matter?

- The person who is using your course with the keyboard instead of the mouse will have to tab to every element that is visible to accessibility tools. This includes background shapes, images, lines, and things they neither need to access nor read. This takes the keyboard user longer to access what they need, causing a poor user experience.
- The person who is using your course with a screen reader hears "tab" every time they press the Tab key to move between elements and will also hear "graphic" every time they tab to a graphic, line, shape, etc. Since they are also trying to learn from the text that you intended for them to hear, hearing all those "empty" tabs makes it very difficult to interpret a logical sequence and then learn from your content.

There are two ways to change objects' visibility to accessibility tools: through the Tab Order window and through the Size and Position window. Since these are both also places to adjust the object's alt-text, they are described by Storyline's Adding Alternate Text for Screen Readers article.


Size and Position

☑ Object is visible to accessibility tools

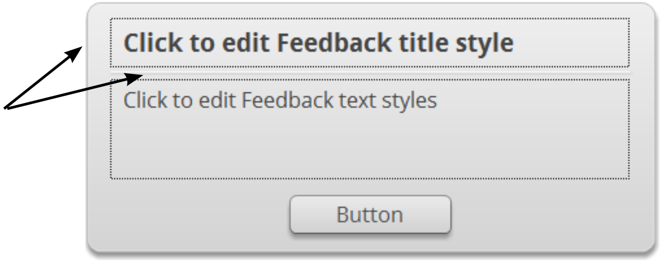
Alternate text:

Changing object visibility on master slides and feedback masters

As of the time of writing this guide, Storyline's default feedback messages (for correct, incorrect and incomplete answers to quiz questions) have decorative elements that are visible to accessibility tools. Also, if you download templates created by other users, it is likely that there will be decorative master slide elements that need to be hidden.

Storyline's article: Using Feedback Masters explains how to open the Feedback Master view and make changes to the default objects. Storyline's Using Slide Masters article explains how to edit objects that are only editable from the Master layout view (custom backgrounds, typically).

Turn off the visibility of the rectangle shape and the line shape on the feedback master layouts so that you don't have to turn them off on each quiz question that you create.


Click to edit Feedback title style

Click to edit Feedback text styles

Since there is no Tab Order window in either of the Master views, the only way to adjust object visibility is through the Size and Position window, which is still located under Format > Size.

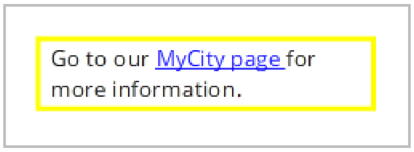
Note: Objects from the feedback master are copied onto Correct and Incorrect slide layers when you create a quiz question slide. Therefore, you can hide or reorder feedback objects using the Tab Order window if you are in Slide view. However, hiding these elements on the slide doesn't impact the master, so you'll have to hide them on every question you create. It is more efficient to hide them on the feedback master before you create any quiz questions.

Creating Keyboard-Accessible Hyperlinks

Unfortunately, hyperlinks inserted within a line of text – such as this link to Storyline's article [Adding Hyperlinks] – are not recognized as a link by screen readers, nor can they be activated using a keyboard.

Instead, you can either:

1. Replace the in-text hyperlink with a button placed elsewhere on the page.
2. Use Storyline's hyperlink feature so that the text characters are formatted to look like a typical hyperlink, and then place an invisible button over top to provide the functionality of the hyperlink in a way that is keyboard and screen reader accessible.



Go to our MyCity page for
more information.

Focus does not move to the hyperlink; only the text box as a whole can be in-focus. Therefore, a keyboard can't be used to activate the hyperlink.

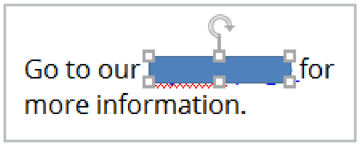
Using either method, make sure to clearly indicate (through alt-text, usually) what users can expect to have happen when they activate the link.

Adding an invisible button

Note that in this context, "invisible" means not visible to sighted users as a separate button. It does not refer to the "Visible to Accessibility Tools" check box, which must remain selected.

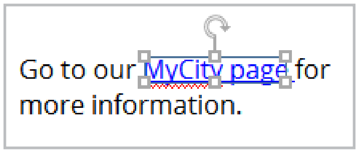
To add an invisible button in Storyline use the following steps:

1. Use **Insert > Shape** to create a rectangle. Size the rectangle to cover the text you want to trigger the hyperlink.



Go to our for
more information.

2. Use **Format > Shape Fill > No Fill** and **Format > Shape Outline > No Outline** to remove the color from both the rectangle shape itself and the outline.

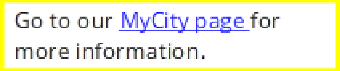


Go to our MyCity page for
more information.

3. Use the Triggers panel to add the action you want (such as jump to a URL or jump to a particular slide) to the colourless rectangle. This gives the rectangle the functionality of a button.
4. Use the alt-text option to describe the button's function (for example, "Access the MyCity intranet page"). You don't need to add the words "click this button" to the alt-text. A screen reader will recognize it as a button and add the word "button" after your alt-text description. Learn how to add alt-text.
5. Consider when, in the sequence of objects on the slide, you want users to be able to access the hyperlink. Use the Tab Order window to adjust the tab sequence as necessary. Learn how to adjust the tab order.

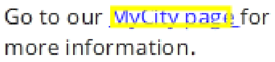
To a sighted user, the result will look and function identically to a hyperlink inserted with the Hyperlink command, but keyboard users will now be able to tab to the button to activate it.

Note that a screen reader will read all the words in the text box when the focus is on the text box. The screen reader will read the alt-text you enter, and enable the user to activate the hyperlink, when the focus is on the button.



Go to our MyCity page for
more information.

A screen reader will read all the text in the text box when the text box is in-focus. E.g., "Go to our MyCity page for more information".



Go to our MyCity page for
more information.

A screen reader will read the alt-text you've entered, followed by the word "button" when the button is in-focus. E.g., "Access the MyCity intranet page, button"

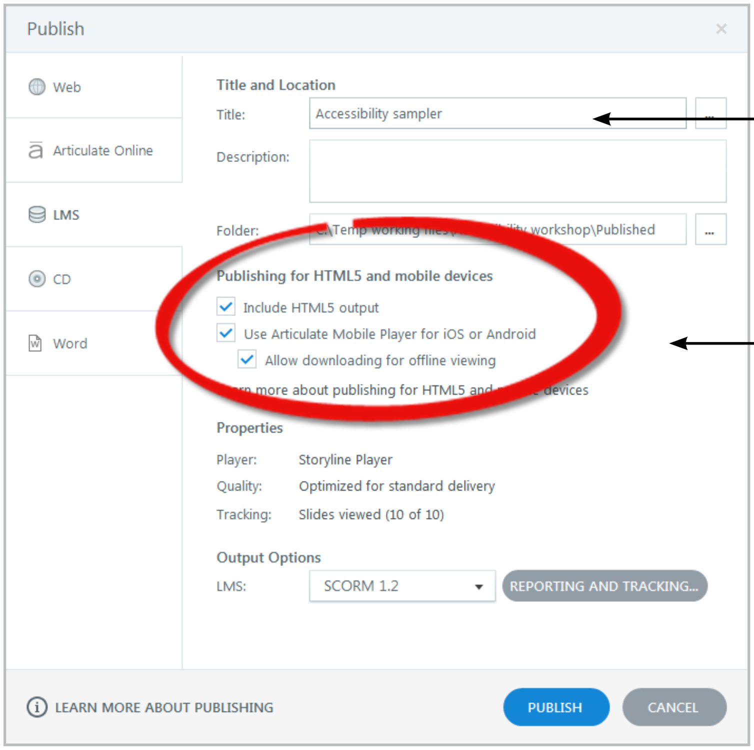
Publishing in an Accessible Format

Storyline only claims to be accessibility compliant when the output is published in its standard Flash format.

Therefore, at this time, HTML 5 output (important for those deploying their course on mobile devices) and output for Articulate Mobile Player (important for iOS – Macintosh – systems that don't recognize Flash output) may not meet WCAG 2.0 or Section 508 accessibility standards because the author will not have control over the tab order, player font size and other features.

The following Storyline articles will be helpful for understanding where to adjust these options and what the lack of accessibility for mobile options will mean for your user:

1. Publishing a Course for LMS
2. Publishing a Course for Mobile Devices



Publish

Web
Articulate Online
LMS
CD
Word

Title and Location

Title: Accessibility sampler

Description:

Folder: C:\Temp working files\Accessibility workshop\Published

Publishing for HTML5 and mobile devices

☑ Include HTML5 output

☑ Use Articulate Mobile Player for iOS or Android

☑ Allow downloading for offline viewing

Learn more about publishing for HTML5 and mobile devices

Properties

Player: Storyline Player

Quality: Optimized for standard delivery

Tracking: Slides viewed (10 of 10)

Output Options

LMS: SCORM 1.2

REPORTING AND TRACKING...

LEARN MORE ABOUT PUBLISHING

PUBLISH

CANCEL

The Title field is read by screen readers (and also appears for sighted users in the browser's title bar). Ensure that you set it to something meaningful before publishing.

The Publishing for HTML and Mobile Devices options, selected by default for most publishing options, can be deselected if you only want to publish accessible content.

Titling Courses and Screens

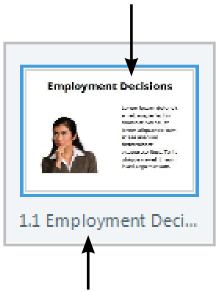
Both WCAG section 1.3.1 Information and Relationships and WCAG section 2.4.2 Page Titled refer to the need to make titles accessible by a screen reader in order to communicate sequence and structure.

In Storyline, the title of the course can be set in the Publish window. The text that is included by the author to identify the slide (this appears under the slide thumbnail in Storyline) will be read by a screen reader only when a menu is used as part of the course player. Text boxes containing a slide title, even those entered in Storyline using a Title placeholder, will not be identified as a title to the screen reader.

Ensuring that each screen which represents a notable change in topic includes a text box with a title, and that this title appears first in the tab order, is the best authors can currently do to meet accessibility requirements. Consider using the wording of the text box contents to indicate sub-topics and show structure.

Do not use alt-text to identify this text box as a title, because that alt-text will be read **instead** of the text box's contents. If your player will include a menu, don't include a number in the title text box contents because it will be duplicated by Storyline's numbering system (scene.slide) in the menu.

A screen reader won't identify the slide title as a title.



The name and numbering under the slide thumbnail is read if you include a menu in the course player.

Checking For Adequate Color Contrast

WCAG section 1.4.3 provides a minimum contrast ratio that represents the acceptable level of contrast between the color of text and the background. This provides for users with moderately low vision who don't use contrast-enhancing technology or a screen reader, but it is a good practice to follow for all users.

The minimum acceptable contrast ratio is 4.5 to 1, except in any of the following three situations:

1. The text is large enough to compensate for the lower contrast (at least 14 pt bold or 18 pt regular). Even large text must have a contrast ratio of at least 3:1.
2. The text is incidental – i.e. decorative or part of an inactive user interface component. Incidental text has no contrast requirements.
3. Logotypes – Text that is part of a logo or brand name has no minimum contrast requirement.

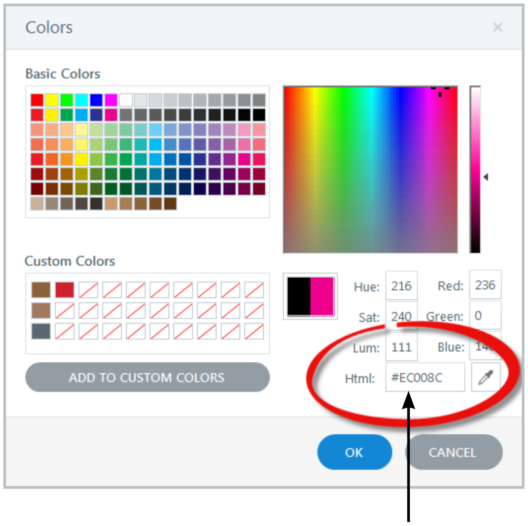
For comparison with the standard, black text on a white background has a contrast ratio of 21:1, which far exceeds the minimum. White text on a black background has the same ratio. The typical "hyperlink blue" color on a white background has a ratio of 9.4 to 1 (also more than adequate), but the same hyperlink color on a black background has a ratio of only 2.2 to 1, and so does not meet minimum contrast requirements. You would either need to lighten the foreground text or lighten the background color in order to increase the contrast.

12 pt text	12 pt text	12 pt text	12 pt text
Contrast 21:1 – Pass	*Contrast 21:1 – Pass*	*Contrast 9.4:1 – Pass*	*Contrast 2.2:1 – Fail*
12 pt text	18 pt text		
Contrast 3.6:1 – Fail	*Contrast 3.6:1 – Pass*		

Checking the contrast ratio is not a feature of Storyline, but there are a number of free applications and online tools that can be used to check contrast:

- The Paciello Group provides a free downloadable contrast checking tool. This tool is particularly handy because it includes a color picker (eyedropper) tool so that you don't need to know the color's hexadecimal code. Another nice feature of this tool is that you have the ability to preview the color combination as it would appear to a user with each of a variety of color vision limitations. Download the Paciello Group's Colour Contrast Analyser.
- If you want a fully online tool with nothing to download, WebAIM has a good one. You can either enter the color's hexadecimal code or pick one from their color palette to get an approximate contrast value. Access WebAIM's Color Contrast Checker.
- MSF&W has a web page that allows you to compare the contrast ratio of various foreground and background combinations, which is useful for understanding how the color contrast ratio calculation works. Access MSF&W's Color Contrast Ratio Calculator.

To find out a colour's hexadecimal code in Storyline, select More Colors from any menu where you'd assign a color (for example, to assign the color of text or a shape). Choose a color from the color palette or use the Pick Color tool (shaped like an eyedropper) to select a color from the slide. The Html box shows the colour's hexadecimal code which you can then enter into a contrast checking tool.



Hue: 216
Sat: 240
Lum: 111
Red: 236
Green: 0
Blue: 14
Html: #EC008C

Use the More Colors window in Storyline to find out the hexadecimal code for a particular color. You can also enter a color code into this box in order to use it.

Enabling Users to Resize Text

The intent of WCAG section 1.4.4 Resize Text is actually that the *user* is able to resize text up to 200% without loss of content or functionality, not the author. Storyline provides the course author with the ability to set the size of slide text, as well as (new to Storyline 2 Update 5) player text. This is helpful, but doesn't really meet the WCAG 2.0 criteria.

The Storyline Support article How to Change the Player Font Size explains how to set the text size of the player.

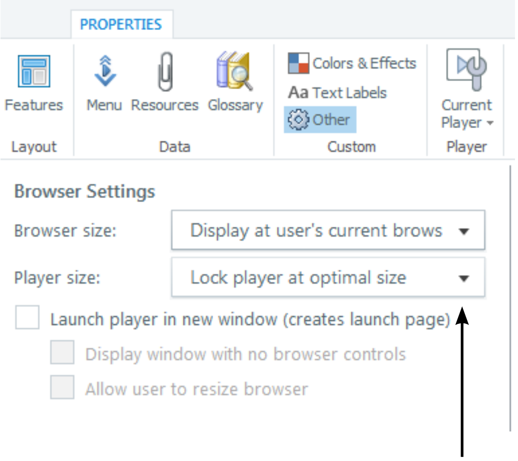
Adjusting the player options so that users can use their browsers to zoom

It is actually not particularly important to change the player font size, unless you feel it is too small for most users. To meet the intent of Section 1.4.4, and enable user control over the magnification, you need to adjust how the player is set up to interact with the user's browser.

Most combinations of Storyline's browser and player size settings prevent users from being able to use their browser to zoom. These settings are intended to give the author control over how the published course appears to the user, regardless of the user's individual settings. This can result in a more predictable published experience, without scroll bars showing up when you don't want them to. However, it also limits the user's control over their own experience, and individual control is important for accessible content.

The Storyline tutorial Changing the Browser Settings and Player Size explains how to adjust these settings. To give the user the ability to magnify the course, you must use the following settings:

- Set the browser size to Display at User's Current Browser Size
- Set the player size to Lock Player at Optimal Size



Browser Settings

Browser size: Display at user's current brows

Player size: Lock player at optimal size

□ Launch player in new window (creates launch page)

□ Display window with no browser controls

□ Allow user to resize browser

In Player > Options, adjust the browser settings to Display at User's Current Browser Size and Lock Player at Optimal Size. This will give the user the ability to magnify course content using their browser's Zoom settings.

Adjusting the story size so there is room to magnify the course before scroll bars appear

With these browser settings, the size of your story (which is the size that the player "locks to" using the Lock Player at Optimal Size setting), becomes a more critical decision. You may want to choose a story size (aka project size) that is small enough that it can be seen without scroll bars on ancient, low resolution monitors (typically 800 x 600), and the small screens of laptops (my 13" laptop maxes out at a resolution of 1366 x 768). Remember to account for the additional height and width added by the player and by the browser's tabs and tool bars. The smaller your story size is, the more opportunity you will give users to magnify the course (enlarging text and images), before scroll bars appear.

Storyline's Choosing a Story Size article explains how to change the size of your project. It's best to figure out your story size before creating any content, because changes will affect your layout and how much you can fit on a slide.

A story size of 770 wide and 395 high is a good size if you're not using a player menu sidebar. If you are, 450 wide and 395 high is about as big as you can get and still have the course fit onto an 800 x 600 monitor without scrolling. At these sizes, a good sized monitor can magnify the course up to 175% before scroll bars appear. In effect, this magnifies a 12 pt font to a 20 pt font and a 10 pt font to a 15 pt font.

Note that you need to publish the course (not just preview it) to accurately see how it behaves in a browser.

Creating Closed Captions for the Hearing Impaired

Synchronized captions provide a way for users with hearing impairments to watch synchronized media presentations that include sound, such as videos or narrated animations. Generally, there is something happening on the screen when audio plays in e-learning, and the timing of the visuals with the audio adds meaning that would be lost in an unsynchronized transcript.

Therefore, it is important that captions:

- Are synchronized and appear at approximately the same time that the audio is delivered
- Are equal in content to the audio (as in a transcript)
- Include non-speech information that is important for understanding – for example, meaningful sound effects and identification of the speaker.

The difference between closed captions and open captions is that open captions are always displayed (whenever there is audio) and closed captions are turned on and off by the user. Both methods of captioning are acceptable for meeting WCAG section 1.2.2 Captions (Prerecorded). However, you will be doing your non-impaired users a disservice by forcing them to watch a word-for-word transcript of audio that they are also listening to. Using open captions violates instructional design best practices by forcing too much cognitive load on the user and reducing their ability to attend to either the voice or the caption.

Most of the work involved in captioning e-learning in Storyline is in synchronizing the captions to the audio, so adding a button to create closed rather than open captions is a minor additional step.

The Storyline Support video How to Create Closed Captioning in Articulate Storyline demonstrates three methods for creating captions (not all of which are closed captions, and not all of which are synchronized): using slide notes, using layers, and using animated text boxes.

Using slide notes

Using slide notes is certainly the easiest method of captioning in Storyline. You place the text caption for the full slide into the Notes area of Storyline, and enable the player to show Notes as a button. You can even change the name of the button in the player to say Transcript or CC instead of Notes. However, although this method of captioning is closed (because it can be turned on and off), it's not synchronized to the visuals appearing on that slide.

It's not clear to me from the WCAG section 1.2.2 standard whether synchronization of captions with visuals is a *requirement* or a best practice. In cases where the timing of the visuals with the narration is not important for understanding, I would feel fairly comfortable with using the slide notes method of captioning. However, for many applications, synchronization will be important and this method will not be valid.

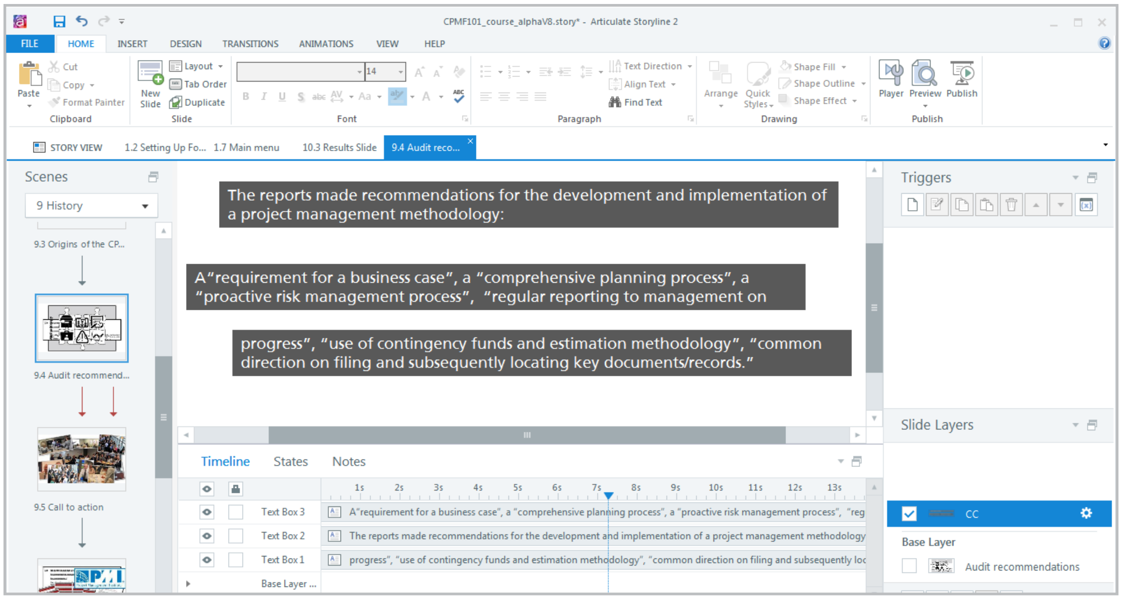
Creating synchronized closed captions

To create synchronized closed captions (the best practice if perhaps not the requirement) requires a combination of Storyline's second and third methods: using layers and using animated text boxes. To make the closed caption layer pervasive (stays on, when opened, on multiple slides; or off, when closed, on multiple slides) also requires the use of a variable. Pervasive closed captioning is nicely shown in the second video on the How to Create Closed Captioning in Articulate Storyline page.

Essentially the process is as follows. It is not difficult, but it is definitely time-consuming:

1. Ensure your audio is final and that the timing of the audio on the slide is final. Changes to audio tracks will be even more of a pain after you add closed captions.
2. Produce a typed transcript of the narration, adding important non-speech audible information such as important sound effects and indications of who is speaking (especially necessary for off-screen voices, etc.).
3. On a new layer (called Closed Captioning for your reference), paste the portion of the script that will be heard on that slide into a text box. Turn off visibility of the text box to accessibility tools so that screen readers don't read it (a visually impaired person will be listening to the recorded audio). Adjust the contrast and font, etc., as appropriate for easy reading. Apply a high contrast fill to the text box so that the text remains easy to read when superimposed over other screen content. Make the width of the text box an appropriate dimension, perhaps 90% of the width of the screen for captions that will appear at the bottom of the screen.

4. Working on your Closed Captioning layer, duplicate the text box several times so that the formatting and accessibility settings are retained. Cut the text into portions that can easily be read in the time that they are on the screen, are synchronized with any important visual actions, and where the remaining text boxes are of the size that the height doesn't block too much of your visuals. You may choose to limit these text boxes to one line of text, or two, high depending on the total length of the script, the number of visually important synchronization points, the size of the caption font, and the layout of your slide.



The reports made recommendations for the development and implementation of a project management methodology:

A"requirement for a business case", a "comprehensive planning process", a "proactive risk management process", "regular reporting to management on

progress", "use of contingency funds and estimation methodology", "common direction on filing and subsequently locating key documents/records."

Text Box 3 A"requirement for a business case", a "comprehensive planning process", a "proactive risk management process", "reg

Text Box 2 The reports made recommendations for the development and implementation of a project management methodology

Text Box 1 progress", "use of contingency funds and estimation methodology", "common direction on filing and subsequently loc

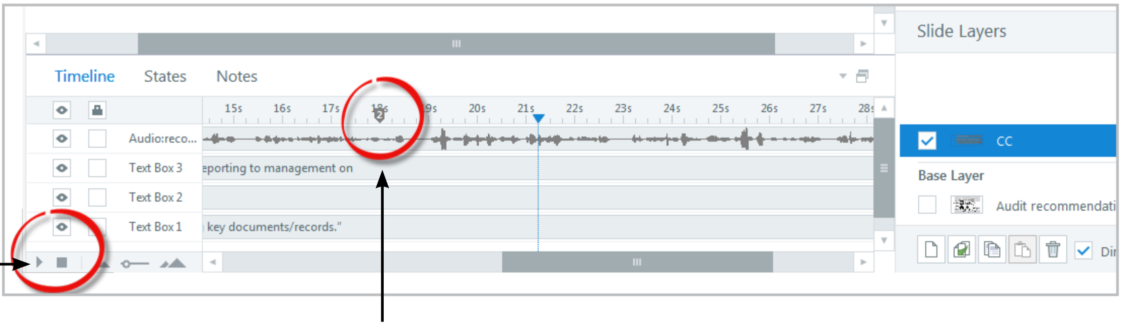
Base Layer ...

Create a layer for the captions (which will simply be animated text boxes). Split the caption for the whole slide into manageably sized text boxes that can be synchronized to appear in time with the audio.

5. Things get a bit sticky when it comes to timing the text boxes to the audio. The best method is to create cue points in the timeline to mark the points in the audio's timeline when the first caption needs to change to the second caption, etc. The slide audio belongs on the base layer, not the Closed Captioning layer, otherwise the audio will not play unless the Closed Captioning layer has been activated. However, because the cue points do not 'travel' between the layers in Storyline, I recommend temporarily copying the audio from the base layer timeline to the Close Captioning layer's timeline, adding cue points, and then removing the audio from this layer.

6. Using the play button at the bottom of the timeline (not the Preview Audio button), play the timeline. Because you've copied the audio onto this layer, you should hear the audio. Press the "C" key on your keyboard to add a cue point to the timeline at the points in the audio when you'd need to display each new caption.

Use the play button at the bottom of the timeline to preview the slide, including the audio.



Audio:reco...

Text Box 3 eporting to management on

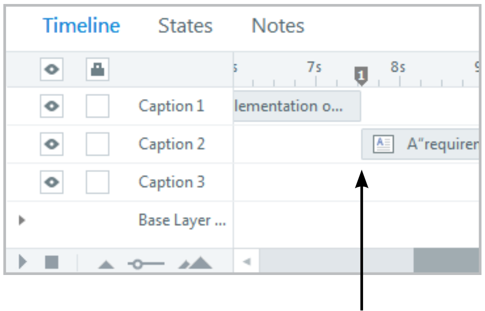
Text Box 2

Text Box 1 key documents/records."

As the timeline plays, press the "C" key on the keyboard to add cue points, marking when to transition from one caption to the next.

7. Delete the audio from the Closed Captioning layer, ensuring it remains on the base layer.

8. Using the cue points as a reference, adjust the timing of each caption on the timeline so that the first caption disappears when it is time for the second caption to appear. Each cue point represents the time in the audio when one caption disappears and the next appears. The Storyline article Using the Timeline explains the basics of using the timeline, including how to change the timing of objects on your slide.
9. Align the text box captions horizontally and vertically so that they are centred on top of each other at the bottom of the slide. You now have synchronized *open* captions for that slide.
10. To make these into closed captions, add a button (typically labelled CC) that shows and hides the Closed Captions layer. The Storyline Support video Creating Interactivity with Triggered Layers explains how to assign a trigger to a button that shows or hides a layer. The second video on the How to Create Closed Captioning in Articulate Storyline page shows how to use variables to make the Closed Caption layer stay showing or hidden throughout your slides depending on a user's choice on any slide.



Adjust the timing of each caption on the timeline . The cue point represents the point in the audio where one caption should transition to the next.

Alternatives

If you're re-thinking your decision to use narration at all when it's so much work to create synchronized closed captions in Storyline, you're not alone. E-learning narration has always been a pain to create and maintain, and updating it almost requires fully starting over at the recording stage. To also meet accessibility standards may be enough to push the return on investment over the edge to "not worthwhile."

- Consider whether learners get enough value from your course narration to make it worth the investment of creating and maintaining it.
- Consider whether captioning is required for your use of the media, or if designing differently would negate the need for captioning. According to WCAG 2.0, captions are not required if the synchronized media is provided as a *supplement* to text, and is **labeled as such**. This is the case if, for example, text is used to describe a procedure, and a narrated video or screencast (labeled as an alternate format for the same information) supports that text.

Creating Full-Text Equivalent Transcripts

Providing equivalent text for synchronized media (videos, screencasts, animations, etc.) is required in addition to providing synchronized captions. Equivalent text helps deaf/blind users access the information through refreshable Braille and other devices. It is also a way of describing unnarrated events that are important to the learning but that rely on sight – for example, the actions and important expressions of a character in a video.

In Storyline, a good place to store the full-text equivalent transcript for a slide is in the Notes panel. Ensure you enable the Notes button on the Storyline player before publishing.

A full-text equivalent transcript should include more than the script that the narrator uses, and it's not necessary that it be exactly the same, just equivalent. Include the following elements also:

- Non-speech audio that is important for understanding – for example, meaningful sound effects and identification of the speaker.
- Descriptions of visual elements that contribute to understanding – for example, meaningful expressions and character actions, setting or context, important physical relationships between objects, etc.

Other uses for equivalent text

You can apply the equivalent text idea to other complex visuals such as charts, tables or graphs, an interaction that uses sliders to reveal information, or a software demonstration.

You can design the equivalent text to be:

- Visible to all users (part of the slide layout)
- Hidden to sighted users but accessible to screen readers (ordered behind another object)
- Accessed as an attachment (from a button). Make sure the attachment is in an accessible format also.
- Accessed from the Notes panel

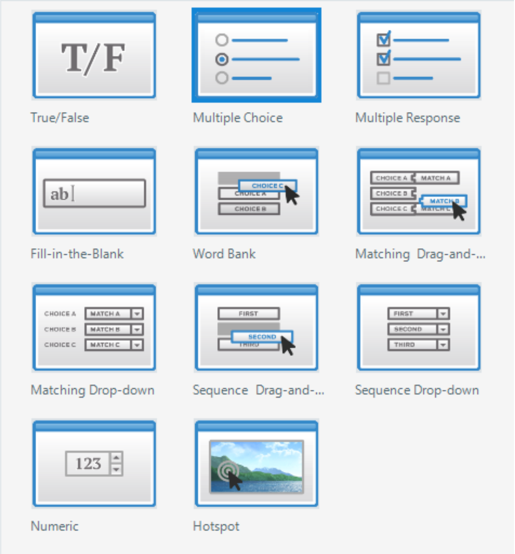
Avoiding Interactivity That Requires a Mouse

Some of Storyline's interactive features will be out of reach of designers who are aiming for an accessible course experience, but there are still many options for quizzing and interactivity that are keyboard-accessible.

Keyboard-accessible interactivity

1. Buttons can be accessed using the keyboard. The user can tab to the button, the screen reader will read the button's text, or the alt-text you've assigned to it, and will identify the object as a button. The user can press the space bar to trigger the action you've assigned to it.
2. The following question types:
    - Multiple Choice (and Pick One)
    - Multiple Response (and Pick Many)
    - True/False
    - Fill-in-the-Blank
    - Numeric
    - Short Answer
    - Essay
    - *Matching Drop-Down
    - *Sequence Drop-Down (and Ranking Drop-Down)

* Use the Drop-Down question types with caution. Choices are not eliminated from the list as they are selected, which changes the experience. Keyboard use is awkward; provide instructions.



Question types that require a mouse are marked with a black arrow to symbolize a cursor.

3. Slide lightboxes (only keyboard-accessible when published with Storyline 2 Update 5, and later. In this update, a bug that prevented the Close Lightbox button from being keyboard-accessible was fixed.)
4. Markers are accessible by keyboard and can be given alt-text to explain their function. The text you type within the marker can be read by a screen reader. The accessibility of markers may depend on how you've used them. For example, if you are "marking" places on an image, this will not have much meaning to a sight-impaired user.

Keyboard inaccessible interactivity

1. Drag-and-drop interactions are inherently inaccessible because they can't be activated by a keyboard, require fine-motor control, and vision.
2. Hotspot interactions (those where you're asked to select the correct area of an image) are also inherently inaccessible because they require sight and can't be activated by a keyboard

Not quite accessible interactivity

1. Sliders *can* be manipulated with the keyboard (as of Storyline 2 Update 5), but this is an instance where you have to be critical of the feature's use instead of purely its function. Sliders are generally used to trigger events on the page, for example, to show a different image for each slider position. Although a user with sight limitations can tab around each object as it's exposed and have the screen reader read any text or alt-text associated with those objects, there is nothing to associate the slider's stop position with any text below it that indicates the meaning of the stop, or anything to indicate that the screen has changed as a result of manipulating the slider. Remember, the screen reader only reads what 's in focus. It *may be* possible to use sliders in ways that are accessible to a blind user, but think your design through carefully or provide equivalent text as a replacement for the experience.
2. Mouse-over (a.k.a "hover" or "rollover") interactions *can* be accessed by keyboard, if you know the command (the JAWS command to activate a mouse-over is INSERT+CTRL+ENTER), but many keyboard users may not be using a screen reader. Creating a similar effect using a button that activates a layer is preferable.

Designer / Supervisor Checklist

You can use this checklist as a guide for testing the accessibility of your work or the work you are supervising. Note that meeting the following requirements does not guarantee your course will pass a WCAG 2.0 level AA or Section 508 compliance audit.

1. **Use a keyboard to navigate through the published or previewed course.** The Tab key advances through the accessible slide elements, Shift-Tab moves backward in the sequence. In general, use the space bar to activate items. If an item that is in focus (has a yellow highlight around it when you tab to it) is a text box or contains alt-text, you can assume it will be read by a screen reader.

   - ☐ Did the designer include instructions for navigating using a keyboard?
   - ☐ If you read the slide elements in the order that you can tab to them, does the slide make sense?
   - ☐ Can you tab to every object that needs to be read? (If not, they won't be read by the screen reader.)
   - ☐ Is the title the first thing you can Tab to, and any persistent navigation buttons the last?
   - ☐ Can all hyperlinks be accessed and activated using the keyboard?
   - ☐ Is all the information that is valuable to the learner (including instructional graphics, videos, or other visuals) available in a textual format that can be tabbed to? (Note that as a reviewer you won't be able to tell if alt-text has been added to an image or button unless you have a screen reader installed).
   - ☐ Have all the objects that only serve a decorative function been removed from the tab sequence?
   - ☐ Is there more than one keyboard-accessible method to access most screens or non-linear sections?
   - ☐ Do quiz questions, question feedback and quiz reviews behave as expected using a keyboard? Can you answer the questions? Can you tab to the feedback to "read" it?
   - ☐ Do you have full control over video and/or audio playback using the keyboard?

2. **Test the course without sound.**

   - ☐ Can all the information that is valuable to the learner (including videos, feedback, or other audio) be perceived without sound?
   - ☐ Can you turn the audio on and off and change the volume using the course player?

3. **Test a published version of the course in any browser you expect learners to use (hopefully this is Internet Explorer 11 or higher, since it is the only browser that Storyline supports for compliance with the JAWS 16 screen reader).**

   - ☐ Does the title appear correctly in the browser's title bar?
   - ☐ Does the course appear on your monitor without horizontal or vertical scroll bars at 100% (normal) magnification? If you expect users to use the course on a smaller screen, or lower resolution (old monitors), change the display settings of your laptop to reflect the minimum likely resolution.
   - ☐ Can you use the browser's zoom controls to increase the magnification of the course?
   - ☐ Is the course still useful when increased to 200% or the size where the text appears as large as 18 pt font would at a normal magnification? Is any text missing? Is scrolling especially arduous? (Some scrolling at this magnification is expected and acceptable).
   - ☐ Does the course text avoid acronyms that would make the course difficult to understand (especially via a screen reader)?
   - ☐ Does the course text avoid, or provide alternatives for, references to color, screen placement, sounds or other cues that would only be meaningful to an individual with uncompromised vision and hearing?
   - ☐ Does the course use a numbered format for lists and sub-lists, or (preferably) avoid sub-lists completely? Does the preceding text indicate that a list will follow? (Note that as a reviewer you won't be able to tell if a screen reader will read the list numbers.)
   - ☐ Do bulleted lists make sense if you imagine there are no bullets there to separate each point?
   - ☐ Is all text set against a sufficiently contrasting background?

Glossary

Alt-text – Alternative text (alt-text) is a word or phrase applied to a screen object to tell users the nature or contents of the object (usually an image). Alt-text is created by the course author and read by a screen reader, if the object is set to be "visible to accessibility tools".

Closed captions – Sometimes called subtitles, captions are on-screen text that represent speech and non-speech elements in a text format so that they can be read by the hearing-impaired as an alternative to listening. Captions can be either closed or open. Closed captions are those that can be turned on or off by the user.

In-focus – The in-focus object is the active, or selected, object that can be interacted with using the keyboard, or that the screen reader can read. In Storyline, the in-focus object appears surrounded by a yellow rectangle when you publish or preview a file and then press the Tab key to bring the focus to the first item in the slide's tab order.

Screen reader – Screen readers are software programs that allow blind or visually impaired computer users to perceive the text on their computers with a speech synthesizer or braille display. JAWS® is a popular screen reader produced by Freedom Scientific.

Synchronous media – Media that has time-based interactive components and/or uses a combination of both audio and video. For example, a narrated video, animation or screencast.

Tabbing – Pressing the Tab key on the keyboard to cycle through each in-focus object in the tab order.

Tab order – The order in which screen elements that are "visible to accessibility tools" are become in-focus when the user presses the Tab key on their keyboard.

WCAG – Web Content Accessibility Guidelines. An international standard that defines how to make Web content more accessible to people with disabilities. These guidelines also make Web content more usable by older individuals with changing abilities due to aging and often improve usability for users in general.

Other Resources

- Articulate provides a free e-book called 6 Best Practices for Designing Accessible E-Learning. Download it here.
- I'm sure there are many excellent training providers for this topic, but I will personally recommend the Eliquo course that I have taken called Creating Accessible Articulate Storyline eLearning Projects.
- Freedom Scientific allows you to download a demonstration version of the JAWS screen reader from their website. The demonstration version allows you to use a fully functioning speech synthesizer for 40 minutes from the time that you activate it. If you want to continue using the product to test your courses after 40 minutes, you'll need to restart your computer.