

Updated eCourse Accessibility Checklist

This document provides a more up-to-date version of the eCourse accessibility checklist and Storyline guidance presented in [Creating Accessible eCourses \(pdf\)](#), which was developed before Articulate released Storyline 360 and Storyline’s “Accessible Player.”

All the information within the Creating Accessible eCourses PDF that remains applicable has been ported over into this document, so you need only reference that PDF if you’re working in a pre-Accessible Player version of Storyline 3.

Due to differences between Storyline versions and the frequency of updates, neither this document nor the Creating Accessible eCourses PDF may perfectly match your experiences developing eCourses in Storyline and accessibility testing them.

This document was last updated: 9/1/22.

General Accessibility

General accessibility standards pertinent to most digital environments, including eCourses.

Standards	Details
Semantically structure content	<p>If possible in your authoring tool, give content the proper semantic structure. Program headings and use an appropriate heading hierarchy¹. Program table structure. Program lists as lists, links as links, etc.</p> <p>Note: Laying out text in Storyline.</p> <p>Note: Links in Storyline.</p>

¹ If you’re using Storyline and have the course title displayed in the [Storyline course player](#), the title will automatically be programmed as an H1, so H2 is the appropriate heading level for the first in-slide heading (e.g., for the slide title).

Standards	Details
Sufficient foreground/background contrast	Ensure there is sufficient contrast between foreground and background elements. Follow the WCAG 2.0 AA guidelines. Use a contrast checker tool, like the WebAIM Contrast Checker , to assess contrast ² .
Do not use color as the sole means of conveying information	If you want to use color to convey information, use it in combination with something else that conveys the same information (e.g., text, symbols, etc.).
Use unique and descriptive link text	<p>Unique: no two links should use the exact same link text unless they direct to the same destination.</p> <p>Descriptive: users should be able to understand a link's destination from the link text alone; no context clues should be necessary.</p> <p>If a link directs to a file, include the file type in the link text: e.g., User Guide (pdf).</p> <p>Do not use "Click here," "learn more" or similarly generic phrases as link text.</p>
Proper reading order	<p>Content should be sequenced in the way that best enables assistive technology users to understand and engage with the content.</p> <p>Note: Reading order in Storyline.</p>

² Chrome recommended; in Chrome, an Eye Dropper (color sampling) tool is available if you click either of the two color swatches.

eCourse General Accessibility

General accessibility standards for eCourses.

Standards	Details
Accessible by default	<p>An eCourse's default state should be its most accessible state. Users should have to actively choose settings that are less accessible.</p> <p>For example:</p> <ul style="list-style-type: none">• Captions should start on; users can choose to turn them off (Turn captions on by default in Storyline)• If an eCourse has "manual play" and "autoplay" modes, it should start in "manual play" mode and require users to actively choose "autoplay" mode within the eCourse• All activities are fully accessible; inaccessible activities are avoided and never required for course completion
Essential course content is available to assistive technologies	<p>Screen readers and other assistive technologies must be able to access/read all essential content (e.g., all text, all controls).</p> <p>Users must be able to engage with and complete all aspects of an eCourse using only their keyboard. Use of a mouse/trackpad should not be required in order to engage with and complete an eCourse.</p>
Hide decorative ³ or duplicative content from assistive technologies: aka, "Accessibility Tools" ⁴	<p>Assistive technology users should not have to encounter content that is decorative, duplicative or, in other ways, does not contribute to their learning experience in an eCourse.</p> <p>Note: Accessibility settings for individual elements in Storyline.</p>

³ "Decorative content" is content that does not communicate information (or at least, information that is not communicated elsewhere) and/or is only intended to enhance visual design.

⁴ The term Storyline uses to describe "assistive technologies."

Standards	Details
Provide easy-to-access accessibility instructions	<p>Accessibility instructions should be provided within an eCourse and should be easy to access from anywhere within the eCourse. For example, you could have an Accessibility Instructions slide that can be accessed via: a link in the course player, a button that is available in all slides, and/or a keyboard shortcut.</p> <p>Accessibility instructions should also be provided before users open an eCourse. For instance, if an eCourse is hosted in an LMS, the eCourse’s LMS-based course description⁵ should include the accessibility instructions.</p>
Prepare an alternate version	<p>Prepare a fully accessible alternate version of the eCourse that can be deployed if an accommodation request is made. HTML, PDF, Docx, and PPT are all acceptable formats for the alternate version so long as it is fully accessible.</p>
Internal consistency	<p>Consistency within an eCourse helps assistive technology users know what to expect. Examples of helpful consistency:</p> <ul style="list-style-type: none"> • Consistent reading order, tab order or Focus Order (or tab/Focus order pattern⁶) in each slide • Consistent word/verb choice when describing actions in buttons, alt text and instructions. For example: <ul style="list-style-type: none"> ○ “Explore ____” for any button that opens a pop-up layer ○ “Jump to ____” for any button that directs users to another slide • Using hyperlinks for all external links and buttons for all internal “links” (i.e., actions/triggers)

⁵ In Storyline, this is accomplished through the Description field in the Publish panel. Use HTML to format Storyline course descriptions, but do not include breaks between tags.

⁶ I.e., even if the reading order/tab order/Focus Order are not exactly the same in all slides, because there are different elements in each slide, the same pattern or logic should govern the order in all slides.

eCourse Interactivity and Navigation

Standards pertaining to interactivity and navigation within eCourses.

Standards	Details
Full use of eCourse controls	Assistive technology users must be able to engage with all interactive controls within an eCourse.
Avoid time limits	<p>Do not impose time limits on activities. If a time limit enhances the experience for certain users, provide it as a secondary option.</p> <p>If a time limit is essential to an activity, you're encouraged to provide an untimed version of the activity as a secondary option.</p> <p>Consider also providing a version that has a time limit but a far more accommodating one, like twice or three-times the default limit.</p>
When a pop-up (layer) opens...	<p>Screen reader Focus should go to the first readable element in the pop-up⁷; users should not have to continue reading through base layer content in order to reach the pop-up content.</p> <p>Content outside a pop-up (e.g., content in the base layer or in other layers) should be hidden from assistive technologies while the pop-up is open⁸, unless the content is necessary for a specific purpose.</p>
Easy restart of slides and slide pop-ups	<p>Users should be able to easily restart slides and slide pop-ups that contain audio or video from their beginning.</p> <p>Note: Restarting slides and pop-up layers in Storyline.</p>

⁷ In Storyline: customize the slide's Focus Order so that the pop-up layer's content immediately follows the button used to open the pop-up layer; and/or, use the "[Prevent the user from clicking on the base layer](#)" slide layer property.

⁸ In Storyline: use the "Prevent the user from clicking on the base layer" slide layer property to hide base layer content from assistive technologies.

Standards	Details
Provide keyboard shortcuts for common in-course actions	<p>For example: a keyboard shortcut to jump to the next slide; a keyboard shortcut to jump to the previous slide; a keyboard shortcut to access the slide transcript; keyboard shortcuts to pause/play slides; etc.</p> <p>Describe these shortcuts in the course/accessibility instructions.</p> <p>Note: Keyboard shortcuts in Storyline.</p>
Do not use interactivity types that require sight and/or use of a mouse	<p>For example, do not use:</p> <ul style="list-style-type: none"> • Drag-and-drop interactions • Hotspot interactions that lack pre-defined selection areas that are accessible to assistive technologies • Hover states to reveal important information or facilitate essential interactions
Describe interactive functionality	<p>Use button text, alt text or other means to describe interactive element functionality⁹ (if it is not communicated by default).</p> <p>If there is text associated with an interactive element — e.g., answer choice text associated with a radio button — consider if it should be incorporated into the interactive element’s alt text and otherwise hidden from accessibility tools, so assistive technology users only have to read and engage with one element instead of two.</p> <p>If an interactive element has multiple states relating to meaningful engagement — e.g., selected/unselected, visited — each state’s alt text should somehow convey which state is presently active.</p> <p>Describe activity-specific functionality in the activity instructions and global functionality in the eCourse instructions.</p>

⁹ E.g., “Currently unselected, True, press enter to select” for a button’s unselected state; “Re-visit pop-up” for a visited state; assuming those states aren’t communicated to assistive technologies by default.

eCourse Multimedia Integration

Standards for integrating multimedia (graphics, audio, video and animated content) in eCourses.

Standards	Details
Timed content should always be available to assistive technologies	<p>Content that is programmed to appear/disappear at certain times during a slide should be available at all times to assistive technologies¹⁰.</p> <p>For example, if you have a text box that is visible when the slide starts and then disappears after 30 seconds: A) there should be a version of that text box that is always available to assistive technologies (a common strategy is to hide this version behind the slide's background); and B) the version that disappears after 30 seconds should be hidden from assistive technologies so they don't encounter two versions of the same content.</p>
Avoid "strobe effects"	<p>Minimize content that appears/disappears in rapid succession. Strobing, flickering or flashing effects can trigger seizures. If such content is unavoidable, provide a warning in advance.</p>

¹⁰ Because you cannot anticipate when assistive technology users will choose to explore a slide's contents.

Standards	Details
<p>Avoid autoplayed media (i.e., non-consensual media: media that plays without express user consent)</p>	<p>Audio and/or video should not play in a slide without the user’s consent, such as is exercised by clicking a play button or engaging in such a way¹¹ that can reasonably result in media being played. That is to say, users must choose/consent to play all media.</p> <p>Choosing to go to the next slide does not sufficiently constitute consent to automatically play audio or video on the next slide.</p> <p>Having to manually start each slide’s media might not be the ideal experience for many users, so it’s recommended you implement an “autoplay” option¹² that users can consent to use once they’re in the eCourse.</p>
<p>Full use of media controls</p>	<p>Assistive technology users must be able to use all media controls within an eCourse, including controls for embedded media.</p> <p>All users should be able to pause/stop and resume media.</p> <p>Note: Media controls in Storyline.</p>

¹¹ For example, if you submit a quiz question answer choice for evaluation, you can reasonably expect to hear feedback for that answer choice, so it would be okay for the feedback audio to begin playing automatically after a user clicks a Submit button, without the user needing to click a separate “play feedback audio” button.

Similarly, if a user chooses to open a pop-up, they can reasonably expect to hear the pop-up’s audio, so again, a separate “play pop-up audio” button would not be necessary.

¹² That is, an option that causes slide audio or video to automatically play once a user arrives in the slide. See [Creating Accessible eCourses \(pdf\)](#) for further details on such an option.

Standards	Details
Transcripts are required for...	<ol style="list-style-type: none"> 1. Slides with audio and completely static visuals¹³ 2. Slides with audio and no time-based visuals¹⁴ that are essential to the learning experience¹⁵ <p>Transcripts must reflect, verbatim, what is said in the audio.</p> <p>If the slide text (or pop-up text) captures, verbatim, what is said in the audio, then a separate transcript is not necessary.</p>
Allow early and easy access to transcripts	<p>Minimize the number of keystrokes required for assistive technology users to access a slide's transcript.</p> <p>Transcripts in Storyline.</p>
Captions are required (and transcripts are recommended) for...	<p>Videos, and slides with audio and <i>any</i> time-based visuals¹² that are essential to the learning experience¹³.</p> <p>Audio descriptions must be captioned as well.</p> <p>Turn captions on by default in Storyline</p>

¹³ That is, the visual components never change during the slide: nothing enters/exits, moves, is highlighted or otherwise emphasized, etc.

¹⁴ That is, visuals that change at specific times. For example: a text box that moves 30 seconds into a slide; a button that appears in the final 15 seconds of a slide; an image that appears 20 seconds into a slide and disappears 20 seconds later; etc.

¹⁵ A visual component is essential to the learning experience when users will miss information by not seeing the visual. For example: if a slide's audio says, "this UC campus is closest to the state capitol," while at the same time, the UC Davis logo is shown, then that UC Davis logo visual is essential to the learning experience, because users wouldn't know which UC campus the audio was referencing without seeing the logo.

A visual component is not essential to the learning experience when it merely repeats or emphasizes information that is provided through audio; that is to say, users will not miss information if they do not see the visual. For example: if a slide's audio says, "UC Davis is the UC campus closest to the state capitol," while at the same time, the UC Davis logo is shown, then that UC Davis logo visual is not essential to the learning experience, because seeing it is not necessary to understanding the information being transmitted through the audio.

Standards	Details
Provide descriptions of non-decorative visual elements	<p>Simple images/graphics: assign succinct, descriptive alt text.</p> <p>Graphs, tables, diagrams and other complex graphics: assign succinct, descriptive alt text <i>and</i> try to relay all of the graphic’s information through another means, such as through semantic structure or a separate, robust audio or text description.</p> <p>If a slide has audio, it is recommended that the audio describe important visuals and/or their key takeaways so that users who cannot see those visuals can better follow along.</p>
Provide audio descriptions for videos	<p>Visual information in videos must be relayed to users <i>while</i> the video plays. It is not sufficient to provide an alternate media description (example pdf) that is consumed separate from the video, though some users may nonetheless appreciate being offered one.</p> <p>Often, the simplest way to achieve this audio description standard is to have the video’s audio describe visual information. But, if that is not done, a separate audio description needs to be provided.</p> <p>An audio description is a secondary audio track that plays in sync with the video and describes important visual information during moments that lack important video audio¹⁶¹⁷.</p> <p>Audio descriptions can be provided as a separate audio file that users can choose to have played in sync with the video or they can be provided in the form of a second version of the video, which users can choose to access in the eCourse, that includes the audio description audio: example video with audio description included.</p>

¹⁶ E.g., in between dialogue, voice-over audio, sound effects that are important to comprehension, etc.

¹⁷ If there is insufficient time between video audio to relay important visual information, an extended audio description is required. An extended audio description pauses the video as necessary so enough time can be provided to describe important visual information in the audio description.



Noted Storyline Features That Impact Accessibility

This is not an exhaustive list of accessibility-related Storyline features but rather a collection of less-publicized features that impact accessibility and pertain to guidance offered in these checklists.

Bugs, updates and testing

Storyline's accessibility features can be buggy — explore [Focus unpredictability when hiding a layer](#) — and accessibility-related behavior sometimes changes with Storyline updates¹⁸, so it's imperative that you test what you build (early and often!) and adjust your development approach as necessary.

At minimum, test all aspects of a build with keyboard navigation, but for the best accessibility testing results, connect with a screen reader user or (better yet!) learn to use a screen reader yourself¹⁹.

Re-test after installing an update, and consider not installing updates mid-build.

Copy/paste

Storyline offers robust copy/paste capabilities, including the ability to copy/paste triggers, layers and objects along with their trigger(s). Utilizing these copy/paste features can help you streamline many aspects of developing accessible content.


Bypassing the built-in manual start course feature

If the first slide of your course features media, the course will open in many browsers displaying just the course title and a play button. This play button may not be fully accessible to all assistive technologies, so it may be best to bypass this built-in manual start feature.

To do so, create a new slide and make it the first slide encountered when the course opens. Give the slide a jump to next slide when timeline on this slide starts trigger. You may also want to give the slide the same color as the course player so it goes (even more) unnoticed.

¹⁸ Again, keep in mind that, due to these factors, the Storyline behavior described in this document and other resources may not perfectly match your experiences using and accessibility testing Storyline.

¹⁹ The [NVDA screen reader](#) is free and comparatively easy to learn/use.



Laying out text

Be aware: if you use Enter or Shift + Enter within a Storyline text box, it will create a break; screen readers will stop reading text at the break and will have to be commanded to continue reading.

It is normal for breaks to occur between some content — e.g., paragraphs, list items — but it can be a problem when breaks occur mid-heading or mid-sentence; it can disrupt the semantic structure, and confuse or frustrate screen reader users.

So, do not use breaks to achieve a certain layout for headings and individual sentences²⁰. Instead, use text box width or expanded spaces²¹ to achieve the desired layout.

Links

For links to be semantically recognized in Storyline eCourses, they must be created in a particular way:

1. You must use link text, so create/use a text box and type in text that will host the link
2. Select the intended link text
3. Open the Insert ribbon and select the Hyperlink button (or use the Ctrl + K keyboard shortcut)

Assistive technologies will not be able to semantically recognize links that have been created by assigning “Open URL” triggers to elements. Storyline elements that have “when user clicks” triggers assigned to them tend to be semantically recognized as buttons.

Converting project from SL3 to SL360: Upgrade Project Text

If you are converting a project from SL3 to SL360, you’ll need to apply the Upgrade Project Text feature so that semantic programming can be recognized within the project. This feature is found at the bottom of the Fonts drop menu available in either the Design ribbon or Slide Master view. Applying this feature may cause slight changes in font size throughout the project²².

²⁰ E.g., Do not use breaks to move certain parts of a heading or sentence to another line of text, which is what was done to this sentence.

²¹ Highlight the space between two words, open the Font settings, set the Spacing to Expanded and assign a value.

²² It’s recommended you apply this feature as early as possible in a conversion process, so you have more opportunities to notice and correct unwanted font changes.

Reading/Focus order

Assistive technology reading/Focus Order in Storyline is controlled through the Focus Order panel. Order elements, in that panel, in the order you want them encountered by users.

Be aware of how the “[Prevent the user from clicking on the base layer](#)” slide layer property affects how assistive technologies navigate a slide’s reading/Focus Order.

If you experience an issue with focus not landing at the start of a text box or alt text description when a new layer opens²³, try this technique: [Addressing focus not landing at start](#).

Focus unpredictability when hiding a layer

Storyline still occasionally exhibits a bug where focus will not land in the appropriate spot in a slide’s Focus Order when a layer is hidden. At present, no known solution for this bug has been found, but it may help to mention the bug in your course’s screen reader instructions, to mitigate user confusion.

Prevent the user from clicking on the base layer

This is a slide layer property. If it is checked for a layer:

- A. When the layer opens, screen reader focus should go to the element in the layer that is highest in the slide’s Focus Order
- B. Assistive technology users will not be able to read or interact with base layer elements

Dialog layers

With the May 2022 versions, Storyline 360 introduced a “Present as: Dialog” slide layer property. If assigned to a layer, users will only be able to interact with that layer’s contents and will not be able to interact with anything else while that layer is open.

If a label and description are assigned, screen readers will read them automatically when the layer first opens²⁴.

²³ This issue has been reported by NVDA users and replicated via NVDA testing but has not been reported or replicated (where it’s known to exist for NVDA users) by JAWS users.

²⁴ Initial testing revealed additional, NVDA-specific behavior: NVDA automatically read the label and description, then automatically read them a second time while automatically moving focus to match what was being read. This behavior was not found with JAWS.



Accessibility settings for individual elements

In Storyline, the accessibility tool visibility²⁵ and alt text of individual elements are controlled through the Accessibility tab in the Size and Position panel²⁶.

There are a few quirks about these accessibility settings worth knowing; that is, if the accessibility settings you believe you've established don't match what you find when you test with assistive technologies, check these factors:

1. Elements can have different accessibility settings in different States
 - a. Elements can even have different accessibility settings for their Normal state (when viewed through the Edit States mode) than they have outside of Edit States mode
2. Elements within another element's States can have their own accessibility settings
3. Groups have their own accessibility settings, separate from the accessibility settings of each individual element within the group
 - a. Being in a group does not negate an element's accessibility settings

Keyboard shortcuts


Storyline now comes with a collection of [built-in keyboard shortcuts](#) covering most common course actions. These shortcuts do not require screen reader users to toggle their screen reader commands off²⁷ in order to be used.

You can create additional, custom keyboard shortcuts through “When users presses a key” triggers. If you use Ctrl + Alt in the key press combination — e.g., Ctrl + Alt + A, Ctrl + Alt + L, etc. — screen reader users will not need to toggle their screen reader commands off in order to use the shortcut.

²⁵ That is, whether an element is visible to accessibility tools or not; assistive technologies will not be able to detect, read or interact with elements that are not visible to accessibility tools.

²⁶ To open this panel, right-click on an element and choose the Size and Position menu option, or select an element and use the Shift + Ctrl + Enter keyboard shortcut.

²⁷ Different screen readers have different terminology for this. In JAWS, you toggle screen reader commands off by toggling off the Virtual PC Cursor (Insert + Z). In NVDA, you toggle screen reader commands off by toggling from Browse mode to Focus mode (NVDA key + Space).



Restarting a slide or pop-up layer

Storyline has two built-in methods that can, if set up properly, replay a slide or pop-up layer:

1. The Replay button that is available in the player controls when the Seekbar is enabled
2. The built-in “Replay the slide” keyboard short: Ctrl + Alt + R

Both methods trigger “When revisiting” functionality; if the “When revisiting” property is set to “Reset to initial state,” the slide/layer will restart from its beginning, but if the “When revisiting” property is set to “Resume saved state,” the Replay button and shortcut won’t have an effect. It is generally recommended that you not use the “Automatically decide” setting for this property, since it’s behavior can be unpredictable.

Media controls

Test the controls for embedded media to ensure they can be used by all assistive technology users (e.g., keyboard navigators and screen reader users)²⁸.

The seekbar and play/pause course player controls can be used to control media *if* the media is synched to the slide timeline and always remains so²⁹.

Transcripts

It is recommended that you not use the Storyline Notes feature to present slide transcripts because the Notes pane requires so many keystrokes to access.

Instead, it’s recommended that you present transcripts through pop-up layers that can be accessed early in each slide’s Focus Order³⁰ and that can also be accessed through a custom keyboard shortcut³¹.

²⁸ As of the May 2022 versions, controls for embedded video were found to be fully accessible.

²⁹ One of the challenges in this regard is ensuring users cannot click on videos — either with their mouse cursor or through keyboard commands — since clicking on a video pauses it without also pausing the slide’s seekbar. To prevent this, it’s recommended that you: 1) hide videos from accessibility tools, so assistive technology users cannot click on them via keyboard commands; 2) place a rectangle with transparent fill over each video, so mouse users cannot click on them; 3) allow those rectangles to be visible to accessibility tools and give them the alt text you would otherwise give the video.

³⁰ For instance, having an “open slide transcript” button ordered immediately after the slide title.

³¹ That is, through a “Show transcript layer when user presses ___ key(s)” trigger you create.

Storyline Techniques

Addressing focus not landing at start

As mentioned elsewhere, if you show a layer that uses the “[Prevent the user from clicking on the base layer](#)” layer property, focus will go to the element in that layer highest within the slide’s Focus Order; however, NVDA users³² may find that focus lands not at the start of the element but somewhere within it³³.

While the root cause of this issue is unknown, it seems to primarily occur with text frames and alt text descriptions that exceed the screen reader’s characters per line limit.

A simple solution is to ensure that the element in the layer that is highest within the slide’s Focus Order does not exceed the NVDA’s default characters-per-line limit of 100. For example, if the highest element within the layer is a text frame containing the layer’s title and a few paragraphs of text, put the layer title in its own text frame and make that text frame the element within that layer positioned highest in the slide’s Focus Order.

Avoiding autoplayed media

There are different ways you can achieve this standard within Storyline; however, achieving this standard typically isn’t as simple as just requiring users to manually start media. There are often other factors you have to account for, such as:

- Keeping the media tied to the slide’s timeline/seekbar so that they start and end together, and so users can reliably scrub forwards and backwards through the media using the seekbar in the player controls
- Accounting for users starting a slide using the Player’s play button or the built-in, play keyboard shortcut
- Having the media continue, pause or stop when other layers open
- Having the media behave as desired when users revisit and/or replay the slide
- Allowing users to choose an “autoplay” option

So whatever method you choose to use, make sure you thoroughly test it.

³² This issue has only been reported and replicated by NVDA users but not JAWS users.

³³ E.g., not at the start of a text frame but rather a few sentences into the text frame.

Manual play layer

An effective method for avoiding autoplayed media and accounting for all the factors noted in the prior section is to use a “manual play” layer.

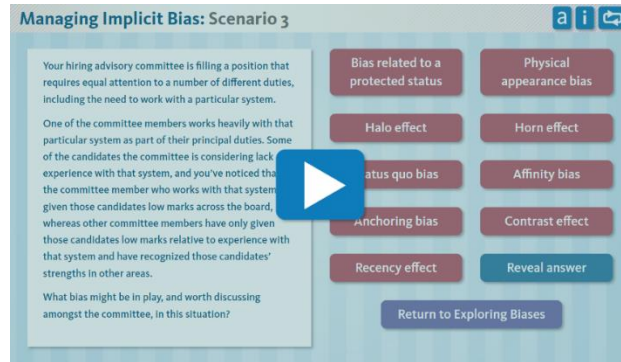
This layer will be triggered to show when a slide’s timeline starts; this trigger may include an “autoplay” True/False condition, as described in the [Allow for an “autoplay” option](#) section.

Show layer Manual Play
When the timeline starts on this slide

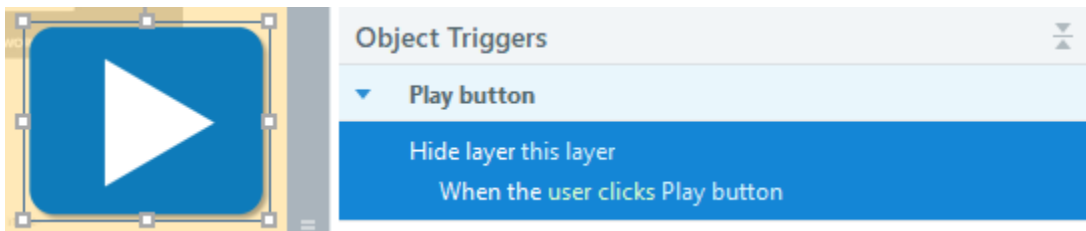
Show layer Manual Play
When the timeline starts on this slide
If autoplay_ON = value False

1. Give this layer the following layer properties:
 - a. Hide other slide layers: optional
 - b. Hide objects on base layer:
 - i. If checked, no users will be able to view or read base layer content
 - ii. If unchecked:
 1. Users will be able to see base layer content
 2. Screen reader users may be able to read base layer content depending on the “Prevent the user from clicking on the base layer” setting
 - c. Hide slide layer when timeline finishes: **unchecked**
 - d. Allow seeking: **no**
 - e. Prevent the user from clicking on the base layer/other layers:
 - i. If checked, no users will be able to interact with base layer content
 - ii. If unchecked:
 1. Screen reader users will be able to read and interact with base layer/other layer content
 - a. However, if screen reader users are able to open a layer that has the “Hide other slide layers” layer property assigned, you may need to incorporate additional conditions and triggers in order to do so, as described in the Account for Hide Other Slide Layers section

2. Non-screen reader users will be able to interact with base layer content so long as nothing in this layer blocks base layer content
 - a. Blocking base layer content may be recommended to ensure users do not skip the manually played media
 - i. E.g. Having a semi-transparent shape cover the entire manual play layer, blocking mouse users from being able to click on any base layer content



- f. Pause timeline of base layer: **checked**
 - g. When revisiting: **Reset to initial state**
2. Include in this layer a button that users will click in order to play the slide's media
 3. Give that button a hide layer, this layer, when user clicks this button trigger



- a. When the layer closes, its "pause timeline of base layer" property will no longer be in effect, causing the base layer's timeline to start, allowing media in the base layer to play
4. Finally, in the slide's base layer, add a trigger that will hide the manual play layer when the slide timeline reaches time 0.01 seconds
 - a. This trigger is necessary to account for users choosing to play the slide by using the play button in the player controls or by using the built-in, play keyboard shortcut

- b. Make sure this trigger is ordered beneath the trigger that shows the manual play layer when the slide's timeline starts³⁴

Show layer Manual Play When the timeline starts on this slide If autoplay_ON = value False
Hide layer Manual Play When the timeline reaches time 0.01s

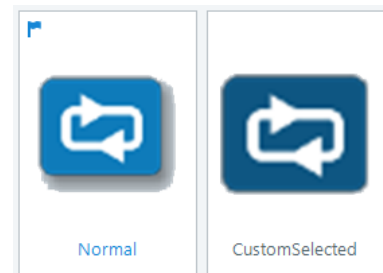


Allow for an “autoplay” option

While the “manual play” setting should be a course’s default setting, because by default, courses should be as accessible as possible, some users may not like having to manually play each slide’s media and may prefer an “autoplay” option, which would allow slides and their media to play automatically as soon as one navigates to a slide.

To enable an autoplay option:

1. Create a True/False variable, with a default value of False, that you’ll use to track if a user has chosen the autoplay option or not; for the sake of these instructions, let’s call that variable “autoplay_on”
2. Edit the trigger that shows the manual play layer when the slide timeline starts and add to it the condition: if autoplay_on = value False
3. Next, create a button that has a normal state and a custom state³⁵; the two states should be visibly different so users can tell which state the button is in
4. Give the button’s Normal state alt text like, “Toggle autoplay on”
5. Give the button’s custom state alt text like, “Toggle autoplay off”



³⁴ Storyline 360 should ensure this order by default and should not allow any “when timeline reaches” triggers to be ordered above “when timeline starts” triggers.

³⁵ Alternatively, you could use the built-in Selected state instead of a custom state, in which case you would not need triggers 6c & 6d, listed on the next page; however, there has been enough unpredictability with using the Selected state for this purpose, both historically and recently, that it may not be a superior, or even advisable, option.

6. Give the button the following conditional triggers:

- a. Change state of this button to Normal when timeline starts on this button if `autoplay_on = value False`
- b. Change state of this button to custom when timeline starts on this button if `autoplay_on = value True`
- c. Change state of this button to Normal when user clicks on this button if `autoplay_on = value True`
- d. Change state of this button to custom when user clicks on this button if `autoplay_on = value False`
- e. (Adjust variable) Toggle `autoplay_on` when users clicks on this button

Change state of Autoplay Button to Normal When the timeline starts on Autoplay Button If <code>autoplay_ON = value False</code>
Change state of Autoplay Button to CustomSelected When the timeline starts on Autoplay Button If <code>autoplay_ON = value True</code>
Change state of Autoplay Button to Normal When the user clicks Autoplay Button If <code>autoplay_ON = value True</code>
Change state of Autoplay Button to CustomSelected When the user clicks Autoplay Button If <code>autoplay_ON = value False</code>
Toggle <code>autoplay_ON</code> When the user clicks Autoplay Button

Account for “Hide Other Slide Layers”

If you allow screen reader users to read and interact with the base layer while the manual play layer is open *AND* users are able to open another layer from the base layer that has the “Hide other slide layers” layer property, you will need to include additional triggers and conditions to ensure media on the base layer does not begin playing when users open that other layer (since opening a layer with the “Hide other slide layers” property would hide the manual play layer, thus unpausing the base layer’s timeline).

In such cases:

1. Create a True/False variable, with a default value of False, that you’ll use to track if the manual play layer is open; for the sake of these instructions, let’s call that variable “`manual_play_open`”
2. On the slide’s base layer, create an adjust variable trigger that will set `manual_play_open` to value False when the timeline starts on the slide; if you’ve allowed for an autoplay option, include an if `autoplay_on = value True` condition
3. Order this trigger above the trigger the shows the manual play layer when the slide timeline starts

4. Create another adjust variable base layer trigger that will set manual_play_open to value False when the timeline reaches time 0.01 seconds
5. Order this second trigger above the trigger that hides the manual play layer when the timeline reaches time 0.01 seconds

▼ Slide - 1.14 Maintaining Legality in the Hiring Proc...

Set Manual_Play_Open to value False
When the timeline starts on this slide
If autoplay_ON = value True

Show layer Manual Play
When the timeline starts on this slide
If autoplay_ON = value False

Set Manual_Play_Open to value False
When the timeline reaches time 0.01s

Hide layer Manual Play
When the timeline reaches time 0.01s



6. Add two triggers within the slide's manual play layer:

- a. (Adjust variable) Set manual_play_open to value True when timeline starts on this layer
- b. This second trigger will be associated with the layer's play button: (Adjust variable) Set manual_play_open to value False when the user clicks play button

Slide Triggers

▼ Layer - Manual Play

Set manual_play_open to value True
When the timeline starts on this layer

Object Triggers

▼ Play button

Hide layer this layer
When the user clicks Play button

Set manual_play_open to value False
When the user clicks Play button

7. Then, on the layer with the "Hide other slide layers" property:

- a. If you want users to be able to open the layer while the base layer's media continues to play, add the following trigger: pause timeline on slide's title (which is how the base layer will be labeled) when the timeline starts on this layer, and if you've allowed for an autoplay option, include an if autoplay_on = value False condition, or...
- b. If you want the base layer's media to be paused when users open the layer, assign the layer the "Pause timeline of base layer" layer property

Base Layer

- Prevent the user from clicking on the base layer
- Pause timeline of base layer

OR

Pause timeline on Effective/Ineffective Hiring
When the timeline starts on this layer
If Manual_Play_Open = value True



8. If there are any objects in this layer that have a hide this layer when user clicks trigger, add to them an additional trigger that will show the manual play layer, when user clicks on this object, if manual_play_open = value True
9. If there is a key press trigger in this layer that hides this layer, create an additional trigger that will show the manual play layer, when users presses the same keys, manual_play_open = value True

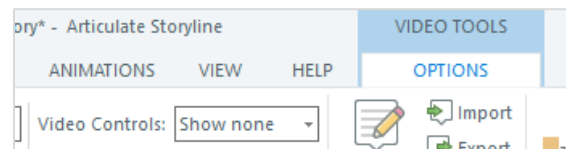
The screenshot shows the Triggers panel in Articulate Storyline, divided into two sections: Key Press Triggers and Object Triggers. In the Key Press Triggers section, there are two triggers: 'Hide layer this layer' (When the user presses Ctrl + Alt + X after clicking on this layer) and 'Show layer Manual Play' (When the user presses Ctrl + Alt + X after clicking on this layer, If Manual_Play_Open = value True). In the Object Triggers section, there are also two triggers: 'Hide layer this layer' (When the user clicks Close pop-up button) and 'Show layer Manual Play' (When the user clicks Close pop-up button, If Manual_Play_Open = value True). Two red arrows point to the 'Show layer Manual Play' triggers in both sections.

Creating custom captions

If you want even more control over the appearance of captions in your slides — or if you need captions to be readable by screen readers³⁶, as would be necessary if captions were serving to subtitle foreign language audio — you can create your own custom captions.

For the following method to work, you need to keep the media’s timeline connected to the slide’s timeline, which means:

- The media needs to play automatically
- The media cannot be paused independent of the slide timeline, so for video:
 - You need to set the “Video Controls” (in the Video Tools Options ribbon³⁷) to “Show none”
 - You need to hide the video from accessibility tools so assistive technology users cannot keyboard navigate to the video and press Enter/Space while focus is on it, thus pausing the video while the slide timeline continues to run...



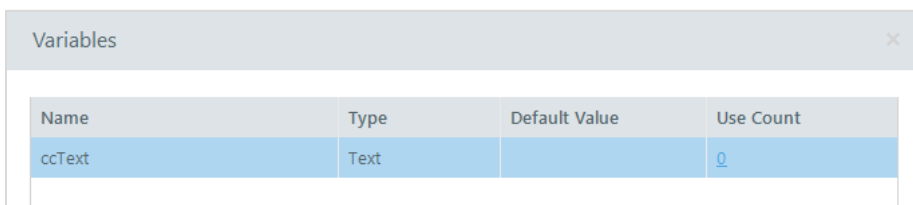
³⁶ Storyline’s built-in captions cannot, at present, be read by screen readers.

³⁷ Available only when a video is currently selected.

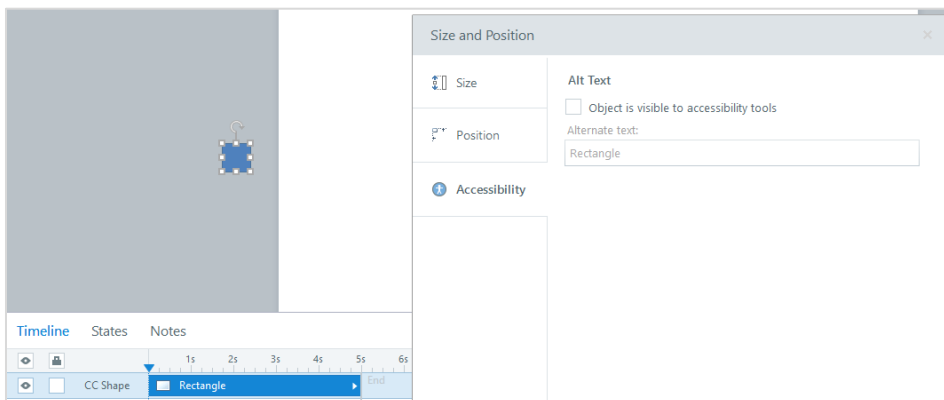
- And you need to place a rectangle with solid fill and 100% Transparency over the entire video (i.e., give the rectangle the same size and position as the video) so that mouse users cannot click on the video, thus pausing it while the slide timeline continues to run
 - It's recommended that you allow this rectangle to be available to accessibility tools and give it whatever alt text you would give to the video

Then, to create the custom captions:

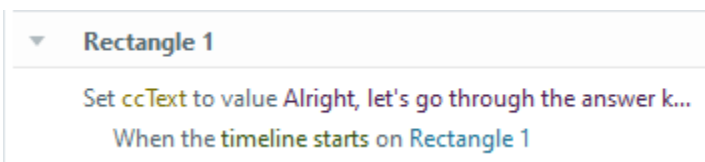
1. Create a Text variable with no default value; this will be the “caption variable”; for the sake of these instructions, let's call that variable “ccText”



2. Create a shape, position it outside the slide's visible area and hide it from accessibility tools

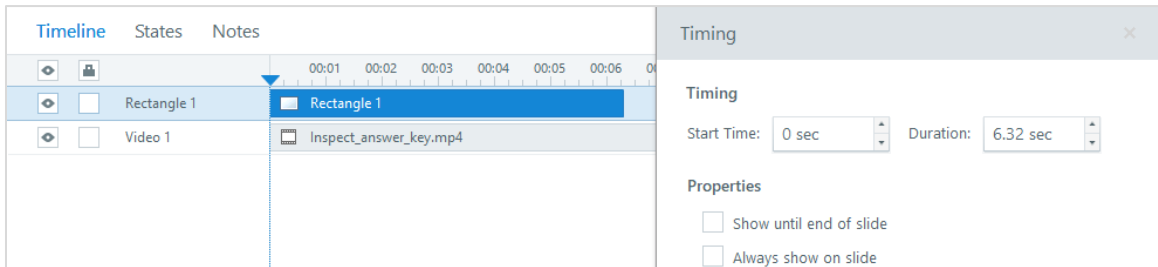


3. Create a trigger associated with the shape that will adjust the ccText variable's value when the timeline of the shape starts, with the newly assigned variable value being the first segment of caption text³⁸

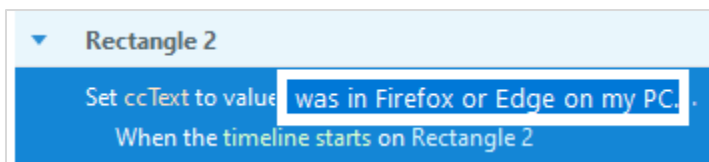


³⁸ Beware: the variable Value field will not accept paragraph breaks. If you copy a text segment that contains a paragraph break and attempt to paste it into the Value field, only the text before the paragraph break will be pasted in the field.

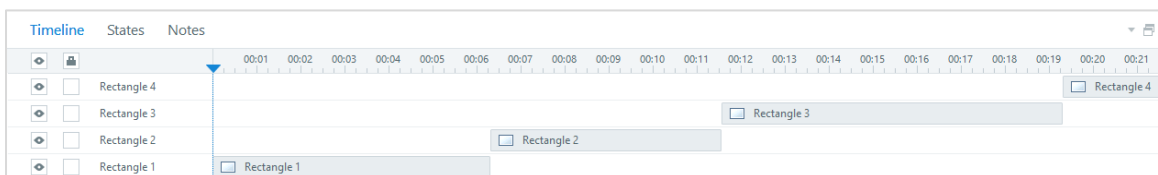
4. Position the shape and adjust its duration within the slide's timeline so that it is present during/synched with the stretch of audio that corresponds with the caption text³⁹



5. Next, copy the shape you created and paste another instance of it outside of the slide's visible area
6. In the Triggers pane, adjust the variable value for the copied instance's trigger so that the value captures the next segment of caption text



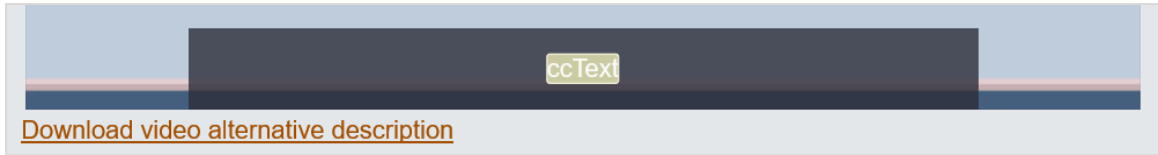
7. Position the new shape and adjust its duration within the slide's timeline so that it is present during/synched with the stretch of audio that corresponds with the caption text
8. Repeat this process as many times as needed so that all audio is captioned



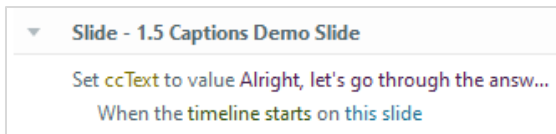
9. Create a text box — let's call it "CC Box" — and include in it a Reference to the caption text variable you created
 - a. To add a variable reference into a text box, place your cursor within the text box and use the Insert > Reference feature or type the variable's name within the text box with a "%" symbol immediately before and after it: e.g., %ccText%

³⁹ It's possible to do this by manipulating the shape's presence within the timeline, but for the most accurate synchronization, it's recommended that you use the Timing panel, accessed by right-clicking on the shape within the Timeline and selecting "Timing...".

10. Give this text box the size, position, shape fill, internal margins and text properties you desire for the caption text



11. Lastly, if the slide's "When revisiting" property is set to "Reset to initial state," create a trigger that will adjust the ccText variable's value to the first caption segment when the timeline starts on the slide, and make sure that trigger is positioned high enough in the slide's trigger order that it won't be impeded by another trigger⁴⁰



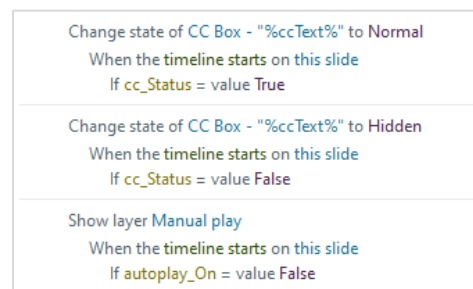
If these are intended to be open captions⁴¹ your work is done.

Turning these into closed captions

Closed captions are captions that can be turned off and on. Since we want our eCourses to be as accessible as possible by default, closed captions should start on, requiring users to turn them off if captions are not wanted.

1. Create a new True/False variable with a default value of True; for the sake of these instructions, let's call this variable "cc_Status"
2. Create two new conditional slide triggers that will control CC Box's visibility at slide start:

- a. Change state of CC Box to Normal when timeline starts on this slide if cc_Status = value True
- b. Change state of CC Box to Hidden when timeline starts on this slide if cc_Status = value False



- c. If the slide includes a [manual play layer](#), order these two triggers above the trigger that shows the manual play layer when the slide timeline starts

⁴⁰ Such as the show manual play layer trigger when timeline starts if autoplay_on = value True trigger, which could pause the base layer before other triggers have a chance to be activated.

⁴¹ Captions that are always visible and cannot be hidden or turned off.

3. Create a button that will be used to turn the closed captions off and on
4. Give the button a custom state visibly different than its Normal state that will represent the button's "captions on" status
 - a. Give the button's Normal state alt text like, "Toggle closed captions on"
 - b. Give the button's custom state alt text like, "Toggle closed captions off"

5. Give the button the following conditional triggers:

- a. Change state of this button to Normal when timeline starts on this button if `cc_Status = value False`
- b. Change state of this button to custom when timeline starts on this button if `cc_Status = value True`
- c. Change state of CC Box to Normal when user clicks on this button if `cc_Status = value False`
- d. Change state of CC Box to Hidden when user clicks on this button if `cc_Status = value True`
- e. Change state of this button to Normal when user clicks on this button if `cc_Status = value True`
- f. Change state of this button to custom when user clicks on this button if `cc_Status = value False`
- g. (Adjust variable) Toggle `cc_Status` when users clicks on this button

▼ Captions button
Change state of Captions button to Normal When the timeline starts on Captions button If <code>cc_Status = value False</code>
Change state of Captions button to customSelected When the timeline starts on Captions button If <code>cc_Status = value True</code>
Change state of CC Box - "%ccText%" to Normal When the user clicks Captions button If <code>cc_Status = value False</code>
Change state of CC Box - "%ccText%" to Hidden When the user clicks Captions button If <code>cc_Status = value True</code>
Change state of Captions button to Normal When the user clicks Captions button If <code>cc_Status = value True</code>
Change state of Captions button to customSelected When the user clicks Captions button If <code>cc_Status = value False</code>
Toggle <code>cc_Status</code> When the user clicks Captions button

Turn captions on by default

If you are using Storyline's built-in captions, you can have them turned on by default (for guidance on turning custom captions on by default, visit the previous section: [Turning these into closed captions](#)).

Open the Variables panel and select the Built-In tab. Scroll down the list of built-in variables until you find a variable named `Player.DisplayCaptions`. It will be a True/False variable set to False by default. Changing the default value to True will enable captions being displayed by default in that Storyline project.