

Creating Accessible Content with Flash Professional

Matt May
Adobe Systems
January 17, 2011



- **Android Accessibility**
 - 2 parts, starts Thurs. Jan. 20
- **HTML5 Accessibility**
 - Tues. Feb. 2
- **PDF Accessibility**
 - 4 parts, starts Mon. Feb. 8
- <http://easi.cc>

What's New in Accessibility

- W3C's WCAG 2.0 being implemented globally
 - US, EU, Canada, Japan, Korea, Australia
- Section 508 – draft aligned with WCAG 2.0
- 21st Century Communications and Video Accessibility Act
- Department of Justice initiative to apply ADA to the public web

➤ Read WCAG 2.0 and Flash techniques

<http://www.w3.org/TR/WCAG20>

Tools for creating accessible Flash

- Adobe offers several tools to author accessible Flash content



Who can access accessible Flash?

- Blind and visually-impaired users
- Deaf and Hard-of-hearing users
- Keyboard-only users
- Users with cognitive disabilities



Common Trouble Spots

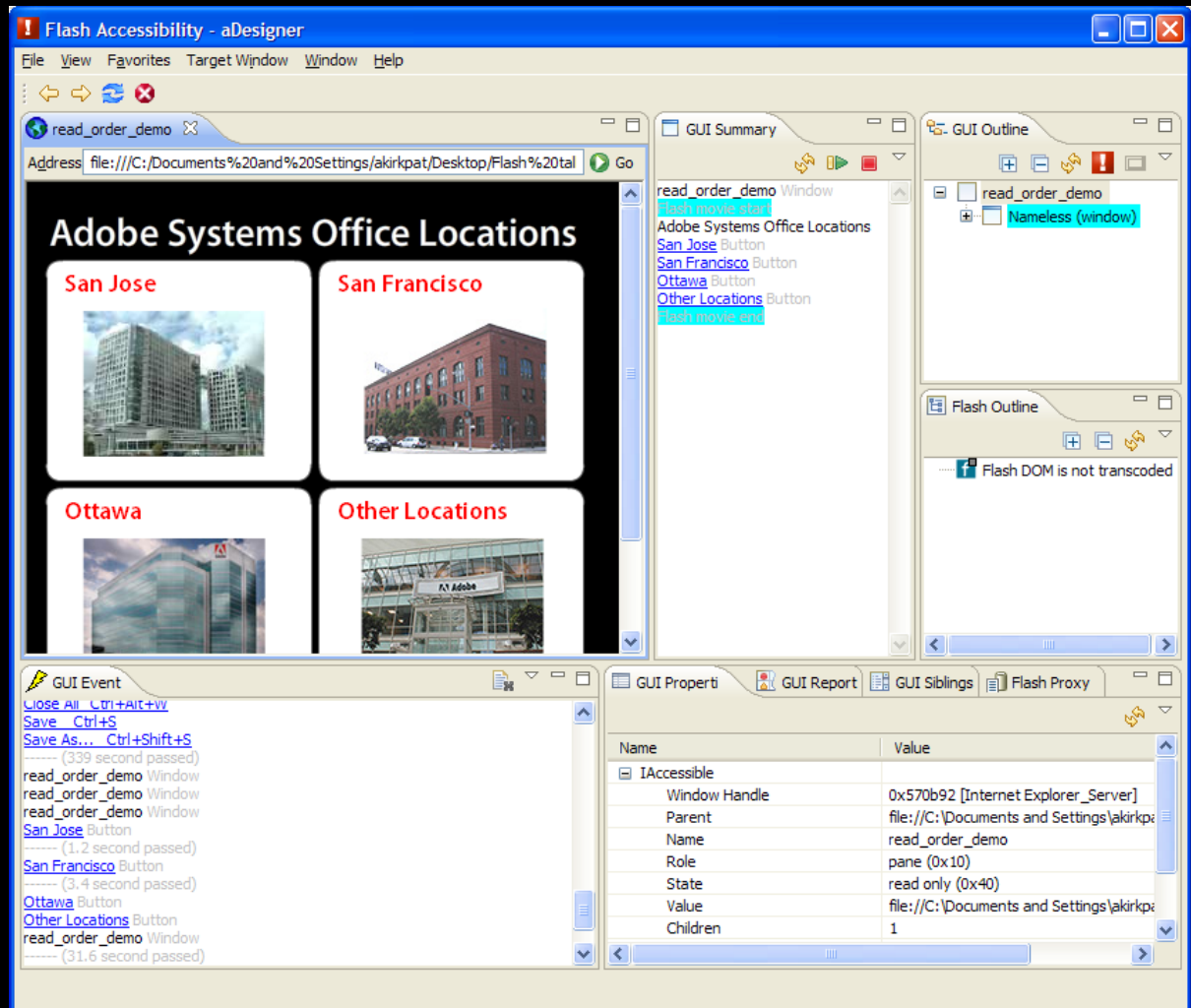
- “Flash movie start. Flash movie end.”
- “Button 1, Button 2, Button 3...”
- Using `wmode=transparent` or `wmode=opaque`
- Support on non-Windows platforms
 - IAccessible2 support and support for Mac and Linux are in the Flash Player roadmap, starting with the next major release of the Flash Player

Should I use Flash?

- Video benefits – superior support for closed captioning
- Development choice – variety of factors
 - Accessible framework preferences: Flex, YUI Astra, jQuery, Dojo
 - Tooling support
 - Testing support
- HTML5 <canvas> – a viable replacement for Flash?
 - Not yet as far as accessibility is concerned
- Standards environment
 - WCAG 1.0 vs WCAG 2.0

Testing accessible Flash

- aDesigner helps review flash content before screen reader testing
- Inspect32
- AccEvent32
- AccExplorer32



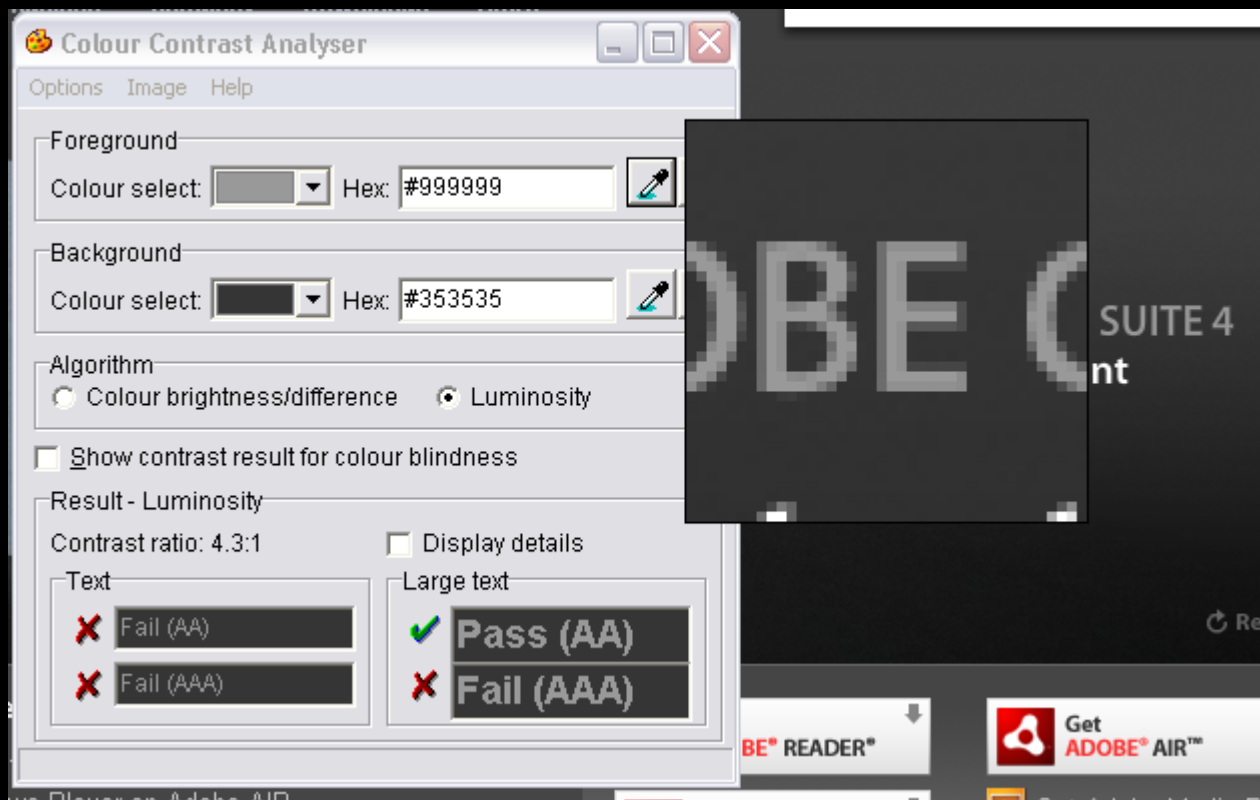
Testing accessible Flash

- Test with assistive technologies
 - JAWS – screen reader
 - Window Eyes - screen reader
 - NVDA – screen reader
 - ZoomText – screen magnifier
 - MAGic – screen magnifier
- Perform keyboard-only tests
- Check color contrast
- Other tests required for WCAG 2.0 compliance

Testing accessible Flash

Color Contrast Analyser Tool

(<http://www.paciellogroup.com/resources/contrast-analyser.html>)



Accessible Flash with Video at US Dept. of Education

- Flash with accessible video controls and captions at DoEd.
- <http://federalstudentaid.ed.gov/mystory/>



Home About "My Story" Media Info

Decided

WHEN I FIRST DECIDED TO GO

Jared's Story

Jared knew he wanted more out of life than a dead-end job. With the help and encouragement of his mother, this older student discovered that college – and his goal of a job in journalism – was well within reach.

[Read Jared's Story](#)

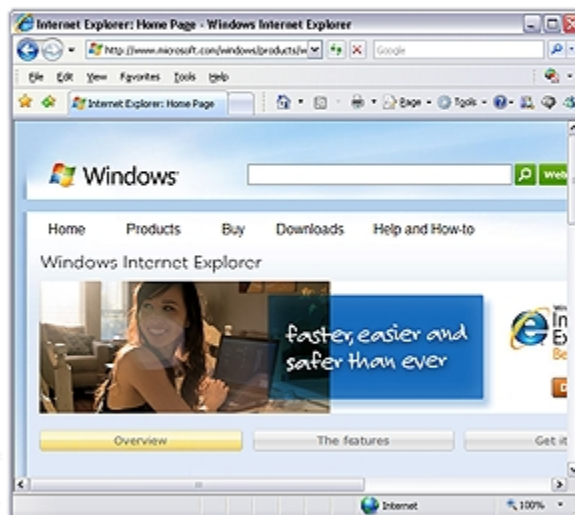
< Back

A video player showing a close-up of Jared speaking. The video has a play button and a back button. Below the video is a caption: "WHEN I FIRST DECIDED TO GO". To the right of the video is the text "Jared's Story" followed by a paragraph about Jared's journey. A link "Read Jared's Story" is provided. A blue arrow points to a "Back" link in the top right corner of the content area.

Creating Accessible Sites in Flash

ADOBE ACCESSIBILITY

🔊 Browsers use software called Microsoft® Active Accessibility® or MSAA that feeds information about the Flash movie to the screen reader.



Microsoft®
Active
Accessibility
(MSAA)



2. Flash Design for Blind Users



- <http://www.adobe.com/accessibility/products/flash/tutorial/index.html>

Key Flash techniques for WCAG 2.0 compliance*

- Accessible names for objects
- Hiding content that shouldn't be voiced
- Labeling form controls
- Providing accessible controls
- Providing logical reading and tab order
- Providing closed captions for audio and video
- Avoiding keyboard trapping
- Setting the language for the content
- Text resizing

* General techniques also apply (e.g. color contrast)

Accessible names for objects

- Objects need names to be voiced by assistive technology
- Names (and descriptions) can be added via ActionScript or using the accessibility control panel
- Separate objects can be grouped into one object for naming

```
print_btn.accessibilityProperties = new  
AccessibilityProperties();  
print_btn.accessibilityProperties.name = "Print";
```

- Related techniques:
 - [Adding Name: http://www.w3.org/TR/WCAG20-TECHS/FLASH1.html](http://www.w3.org/TR/WCAG20-TECHS/FLASH1.html)
 - [Adding desc: http://www.w3.org/TR/WCAG20-TECHS/FLASH2.html](http://www.w3.org/TR/WCAG20-TECHS/FLASH2.html)
 - [Naming a combined object: http://www.w3.org/TR/WCAG20-TECHS/FLASH5.html](http://www.w3.org/TR/WCAG20-TECHS/FLASH5.html)

Hiding content that shouldn't be voiced

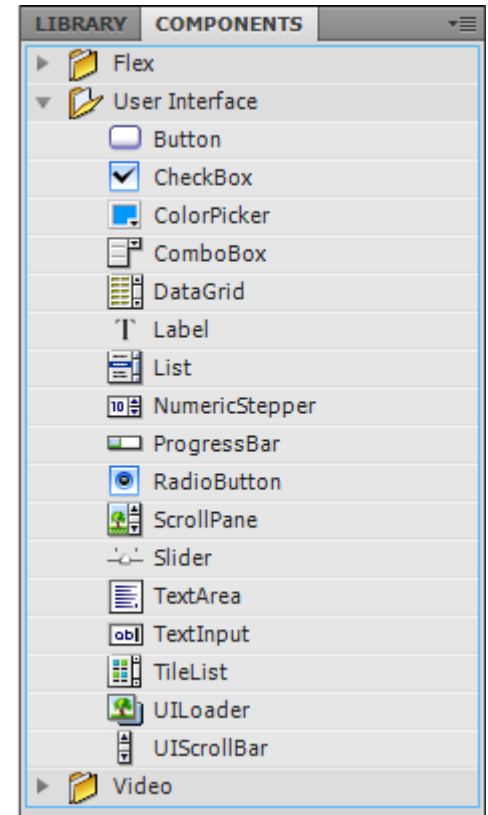
- Content that is meant to not be heard can be hidden from assistive technology
 - Equivalent to using `alt=""` on a spacer image in HTML
- Use the `visible` property (set to `false`) to hide objects from the reading order when also hidden from view
- Use the `AccessibilityProperties` object's `silent` property to hide decorative objects from the reading order

```
_root.decorative_mc._accProps = new Object();  
_root.decorative_mc._accProps.silent = true;
```

- Related techniques:
 - Hiding objects: <http://www.w3.org/TR/WCAG20-TECHS/FLASH3.html>

Providing accessible controls

- Using controls that have built-in support for assistive technologies saves time.
 - Flex SDK includes a large set of accessible controls
 - Flash Professional contains a similar, but smaller set
- Components in Flash Professional need accessibility enabled using `enableAccessibility()`
- Flex SDK component accessibility is enabled by default



Labeling form controls

- Form controls need labels for users to identify what the control is for.
- Flash Player includes an autolabeling mechanism
 - Explicitly defined labels always work, autolabeling often works
- Demo: [Controls with label property](#)
- Related techniques:
 - Component label property <http://www.w3.org/TR/WCAG20-TECHS/FLASH29.html>
 - Autolabeling: <http://www.w3.org/TR/WCAG20-TECHS/FLASH32.html>
 - Naming grouped controls: <http://www.w3.org/TR/WCAG20-TECHS/FLASH8.html>

Providing logical reading and tab order

- For content that is not limited to a simple linear visual layout it is critical to define the reading order
 - The Flash Player assigns a reading order that is often incorrect
- Focusable objects need a correct position in the tab order
 - e.g. `myObject.tabIndex = 10;`
- Related technique:
 - Defining tab and reading order - <http://www.w3.org/TR/WCAG20-TECHS/FLASH15.html>

Avoiding keyboard trapping

- Firefox doesn't allow keyboard focus into Flash content without a mouse click or JavaScript encouragement
- Demo: [Focusing Flash in Firefox](#)

- Related Techniques:
 - keyboard trap: <http://www.w3.org/TR/WCAG20-TECHS/FLASH17.html>

Avoiding keyboard trapping

- A JavaScript `<script>` tag will be generated and added to the HTML document containing the Flash movie.
 - Set a `tabindex` value of "0" on the `<object>` element of each Flash movie found in the page. This causes the Flash objects to become part of the tab order.
 - Optionally, create a hidden anchor element before and after the Flash movie, which is used by the SWFFocus class to move focus out of the Flash movie back into the HTML page. Alternatively, the developer can specify existing focusable HTML elements as adjacent tab stops for the Flash movie.
 - Set event handlers for the Flash movie object, so that when it receives focus, the SWFFocus class is notified to manage the movie's internal tab order.
 - The SWFFocus class monitors changes in focus within the Flash movie. When a focus wrap is detected in the movie, a JavaScript function will be called to instead move focus back to the neighboring HTML content.

Providing closed captions for audio and video

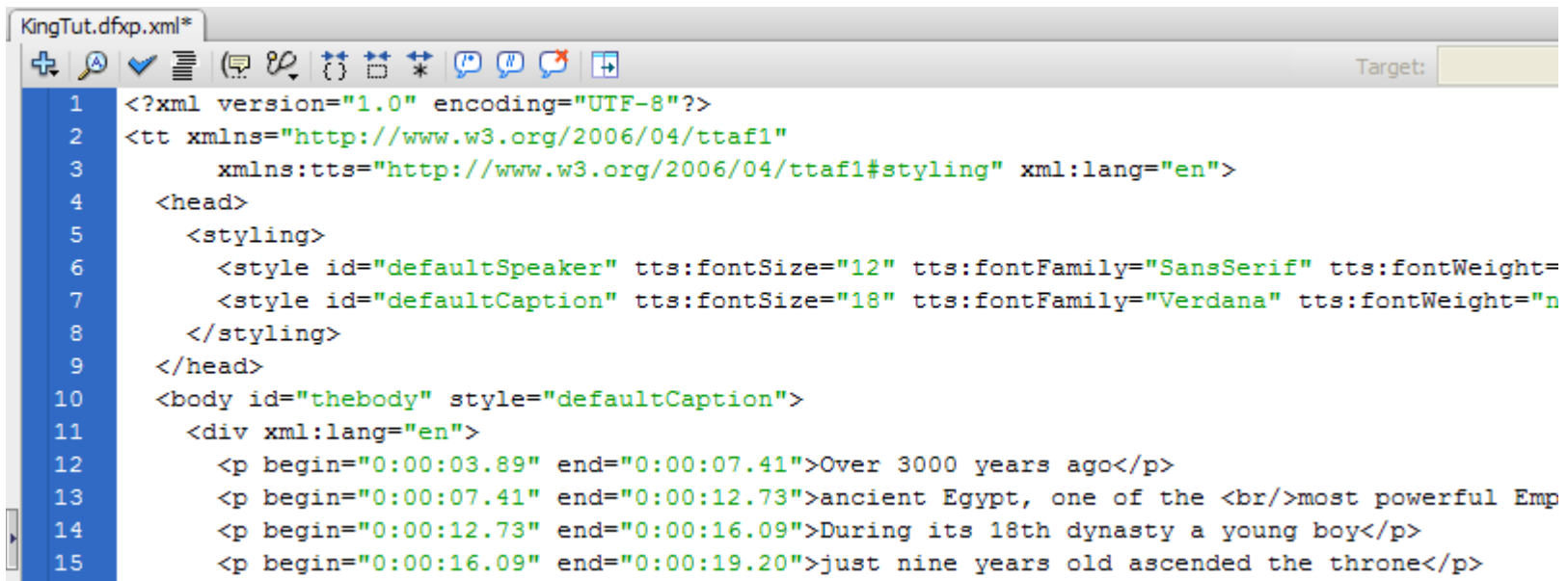
- FLVPlaybackCaptioning provides support for closed captioning
- FLVPlayback handles accessibility of controls for video



- Related technique:
 - Providing captions: <http://www.w3.org/TR/WCAG20-TECHS/FLASH9.html>

How Does Captioning Work in Flash?

- The FLVPlaybackCaptioning component supports two methods
 - Embedded caption data in cue points
 - Parsing DFXP/TTML caption data and creating ActionScript cue points, by TimedTextManager
 - Cue points are triggered at specific times and displayed

The image shows a screenshot of an XML editor window titled 'KingTut.dfxp.xml*'. The editor displays XML code for a timed text document. The code includes a root element '<tt>' with namespaces for 'http://www.w3.org/2006/04/ttaf1' and 'http://www.w3.org/2006/04/ttaf1#styling'. It contains a <head> section with <styling> elements for 'defaultSpeaker' and 'defaultCaption'. The <body> section contains a <div> with four <p> elements, each with a 'begin' and 'end' time attribute and text content. The text content describes the history of ancient Egypt.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <tt xmlns="http://www.w3.org/2006/04/ttaf1"
3   xmlns:tts="http://www.w3.org/2006/04/ttaf1#styling" xml:lang="en">
4   <head>
5     <styling>
6       <style id="defaultSpeaker" tts:fontSize="12" tts:fontFamily="SansSerif" tts:fontWeight=
7       <style id="defaultCaption" tts:fontSize="18" tts:fontFamily="Verdana" tts:fontWeight="n
8     </styling>
9   </head>
10  <body id="thebody" style="defaultCaption">
11    <div xml:lang="en">
12      <p begin="0:00:03.89" end="0:00:07.41">Over 3000 years ago</p>
13      <p begin="0:00:07.41" end="0:00:12.73">ancient Egypt, one of the <br/>most powerful Emp
14      <p begin="0:00:12.73" end="0:00:16.09">During its 18th dynasty a young boy</p>
15      <p begin="0:00:16.09" end="0:00:19.20">just nine years old ascended the throne</p>
```

- YouTube now offers automatic captions
 - Employs Speech To Translation to generate transcript
 - Utilizes programmatic alignment of text to audio to generate captions
- Uploading transcripts also possible for more accurate results.

Add New Captions or Transcript

File

Type
 Caption file (includes time codes)
 Transcript file (English only) (*beta*)

Language:
Name (optional):

[Return to Available Tracks](#)

Captions and subtitles help viewers with hearing disabilities and people who speak other languages to enjoy your videos. To learn more about this feature, see the [Help Center](#).

Important: By clicking "Upload file", you are representing that the uploaded text does not violate [YouTube's Terms of Service](#), and that you own all copyrights in this video and caption track or have authorization to upload the video with captions.

Setting the language for Flash content

- Setting the language for Flash content is accomplished via the lang attribute for the object element or the HTML element.
- Setting different languages for individual phrases within a Flash asset is not currently possible.
- Demo: [Setting language in Flash](#)
- Related technique:
 - Language of page: <http://www.w3.org/TR/WCAG20-TECHS/FLASH13.html>

Text resizing in Flash

- Flash content can often work with browser zoom features
- When not possible to use the browser zoom (e.g. Flex) different styles can be applied to alternative views to achieve the same effect.
- Demo: [CSS Styles in Flash](#)
- Related techniques:
 - Resizing: <http://www.w3.org/TR/WCAG20-TECHS/FLASH33.html>

Resources

- Adobe Flash Professional
adobe.com/products/flash
- Adobe Accessibility Resource Center
adobe.com/accessibility
- Adobe Accessibility Blog
blogs.adobe.com/accessibility
- Flash techniques for WCAG 2.0:
w3.org/TR/WCAG20-TECHS/flash.html

Matt May

Accessibility Evangelist

mattmay@adobe.com

ADOBE® ACCESSIBILITY

