



# Adobe Flex and AIR for Accessible Rich Internet Applications

Matt May (@mattmay)

25 Mar 2010



# What is AIR?

- Application framework
- Build desktop applications using Flash, HTML and PDF
- AIR 2.0 adds support for MSAA
- <http://adobe.com/products/air>
- <http://labs.adobe.com/technologies/air>

# What is Flex?

- Language for creating Flash applications
- XML-based syntax
- CSS-based layout
- ActionScript-based behaviors
- Open-source framework
- Open bug tracking
- <http://adobe.com/products/flex>
- <http://labs.adobe.com/technologies/flex>

## 28 Accessible Flex 3 Components

- Accordion
- AdvancedDataGrid
- Alert
- Button
- CheckBox
- ColorPicker
- ComboBox
- DataGrid
- DateChooser
- DateField
- Form
- Image
- Label
- LinkButton
- List
- Menu
- MenuBar
- Panel
- RadioButton
- RadioButtonGroup
- TabNavigator
- Text
- TextArea
- TextInput
- TitleWindow
- ToolTipManager
- Tree
- Validator

## 27 Accessible Flex 4 Components

- Button
- CheckBox
- RadioButton
- Panel
- Application
- TextInput
- TextArea
- ScrollBar
- Spinner
- NumericStepper
- Slider
- List
- Label
- RichText
- RichEditableText
- PopUpAnchor
- ComboBox
- DropDownList
- ButtonBar
- HSlider
- TabBar
- TitleWindow
- ToggleButton
- VideoPlayer
- VSlider
- WindowedApplication
- Window

# Flex 4 and Flash Builder demo

## Consider non-mouse-driven modes of use

- Keyboard-driven navigation
- Keyboard-driven input
- Voice control

## Consider accessibility-related usage scenarios

- Color differentiation difficulties
- Visual acuity limitations
- Audio and visual feedback requirements

## Best Practices

- Provide keyboard access
- Use accessibility-enabled components
- Convey Relationships
- Provide closed captions for video
- Provide instructions
- Enlist the experts



## Provide Keyboard Access

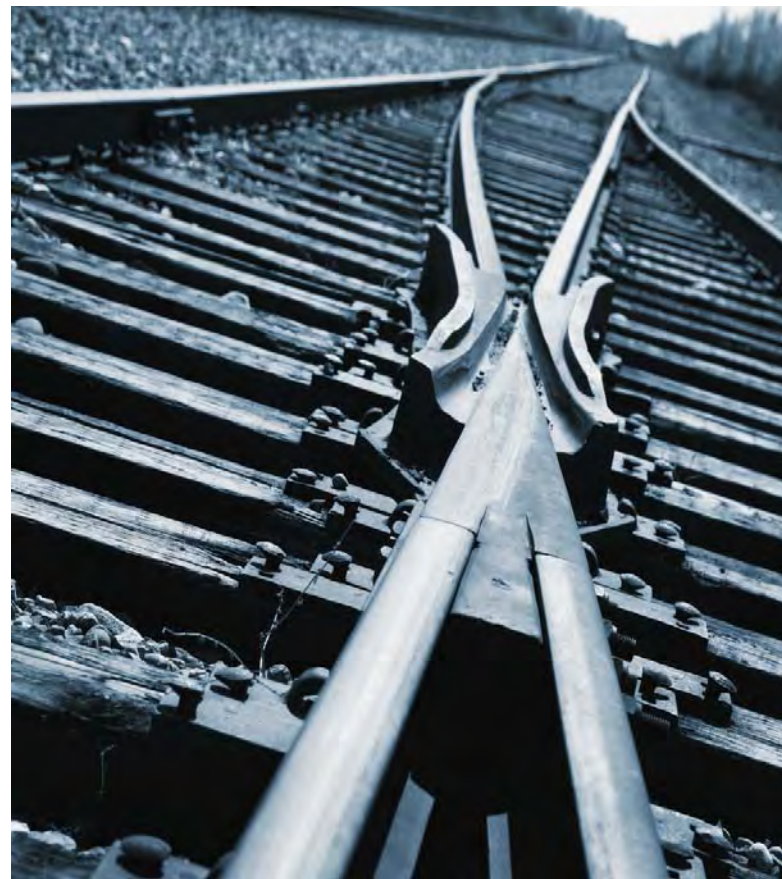
- The spec should lay out the tab sequence
- Necessary to detail ways users move through an application
- Providing a fast, keyboard-driven path for navigating through the application can be beneficial.
- Need to decide how to reveal the available keyboard sequences to the user.
- Demo: Defining tab and reading order

## Provide Keyboard Access: Testing

- Tabbing through controls is a good start
- Make sure to shift+tab backwards
- Learn about expected control behaviors
  - Flex Keyboard Documentation:  
[http://livedocs.adobe.com/flex/2/docs/wwhelp/wwhimpl/common/html/wwhelp.htm?context=LiveDocs\\_Parts&file=00001025.html](http://livedocs.adobe.com/flex/2/docs/wwhelp/wwhimpl/common/html/wwhelp.htm?context=LiveDocs_Parts&file=00001025.html)
  - Controls:  
<http://msdn.microsoft.com/library/en-us/dnwue/html/ch08c.asp>
  - Guidelines for Keyboard User Interface Design:  
<http://msdn2.microsoft.com/en-us/library/ms971323.aspx>

## Use Accessibility-Enabled Components

- “Accessible” components are those with built-in MSAAs support
- Even with accessibility enabled, there are accessibility concerns to address.
- Using custom components adds substantially to development time
  - Assistive technology interoperability
  - Complex keyboard support



# Building Accessible Applications



## JAWS for Windows

- [http://www.freedomscientific.com/fs\\_products/software\\_jaws.asp](http://www.freedomscientific.com/fs_products/software_jaws.asp)
- Flex 3 requires JAWS 8
- Flex 4 requires JAWS 10

## NVDA for Windows

- <http://nvda-project.org>



## Flash Components Scripts

- <http://www.adobe.com/accessibility/products/flex/jaws.html>
- These scripts handle issues related to Flash components used in Adobe Flex applications

## Other Assistive Technologies

- Window-Eyes, IBM HomePage Reader, HAL/SuperNova, ZoomText, PC-Talker (Japanese), and JAWS-J also interoperate with Flash and Flex content.

## Microsoft MSAA Resources

- Active Accessibility 2.0 Documentation
  - [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msaa/msaastart\\_9w2t.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msaa/msaastart_9w2t.asp)
- Microsoft Active Accessibility 2.0 SDK Tools (Inspect32, AccExplore32, AccEvent32)
  - <http://www.microsoft.com/downloads/details.aspx?FamilyId=3755582A-A707-460A-BF21-1373316E13F0&displaylang=en>
- Microsoft Active Accessibility: Architecture
  - <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnacc/html/actvaccess.asp>

# What to know about Flex 3

- You have to turn accessibility on
- Use the JAWS scripts (up to version 10)
  - <http://www.adobe.com/accessibility/products/flex/jaws.html>
- JAWS 11 works without scripts
  - May add scripts for better performance in the future
- Latest versions are always better
  - JAWS 11 Update, Flex 3.5 SDK and scripts

## Enabling Component Accessibility

### For all applications

- Edit the flex-config.xml file to set the <accessible> value to true (in frameworks directory)

```
<mxml-compiler>  
  <accessible>true</accessible>  
</mxml-compiler>
```

### Using the command line compiler

- Use the accessible option with the command line compiler

```
mxmlc -accessible c:/dev/myapps/appl.mxml  
mxmlc -compiler.accessible c:/dev/myapps/viewer.mxml
```

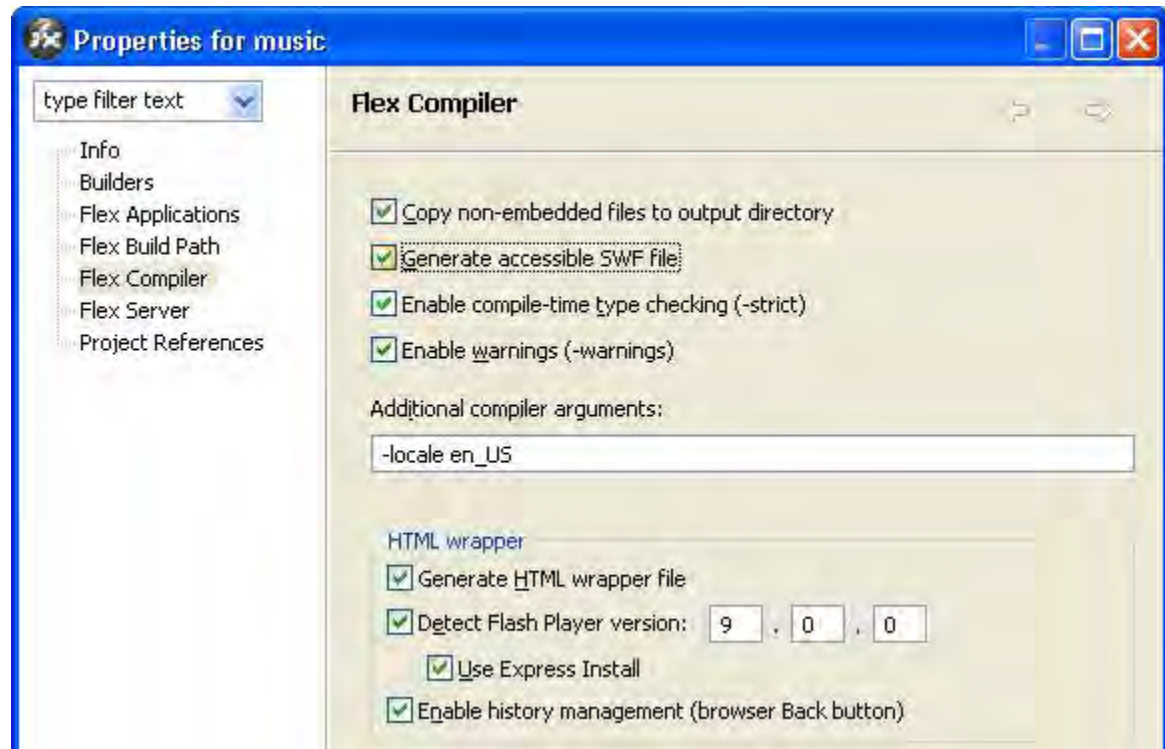
### Performance impact

- Adds about 1k in overhead per component

# Building Accessible Applications

## In Flex Builder 3

- Select “Generate Accessible SWF file” in the Project Properties dialog
- Setting must be modified for each project





## What Does Enabling Component Accessibility Do?

- Adds **AccessibilityImplementation** code for components
- Adds support for methods and properties that are important for MSAA accessibility support.
- Support for assistive technologies (e.g. JAWS screen reader) requires MSAA support

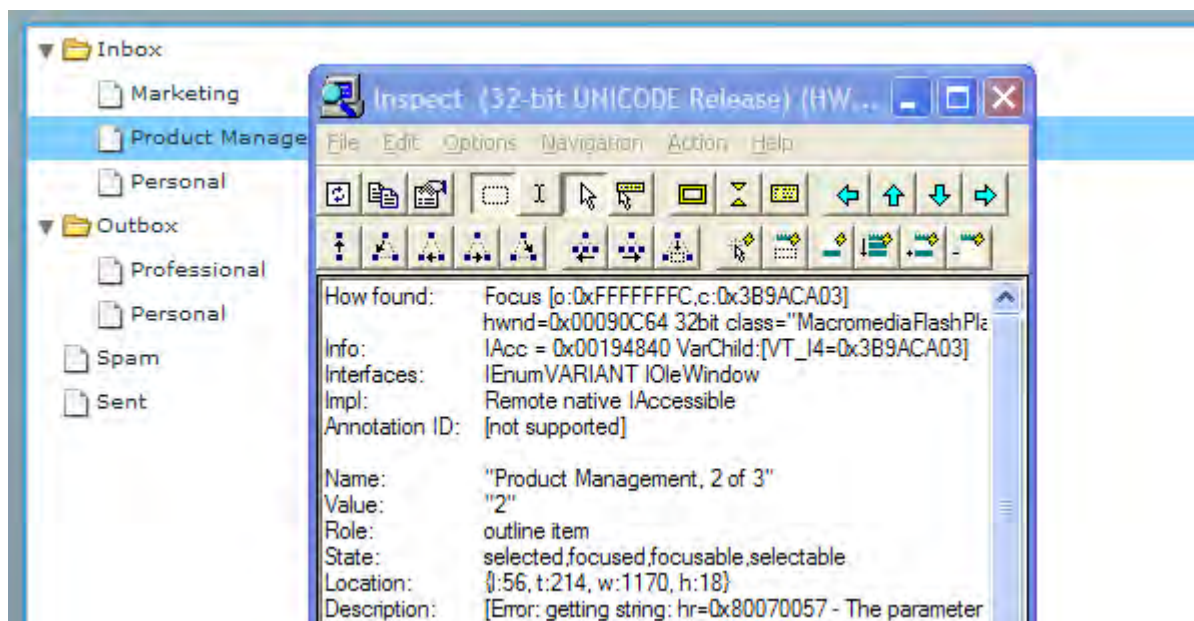
# What to know about Flex 4

- Use it!
- Works with JAWS <11 with scripts
- JAWS 11 works without scripts
  - May add scripts for better performance in the future
- Accessibility is on by default in the compiler
- Major improvements in Flex 3 (Halo) components
- New support for Flex 4 (Spark) components
- Available free (open source) at <http://www.adobe.com/products/flex/>
- Flash Builder
- Flash Catalyst

# Building Accessible Applications

## Test with MSAA Inspector Tools

- Quick tool for identifying role and state info



## Provide Keyboard Access

- The spec should lay out the tab sequence
- Necessary to detail ways users move through an application
- Providing a fast, keyboard-driven path for navigating through the application can be beneficial.
- Need to decide how to reveal the available keyboard sequences to the user.
- Flex Documentation:  
[http://livedocs.adobe.com/flex/3/html/accessible\\_5.html](http://livedocs.adobe.com/flex/3/html/accessible_5.html)
- Demo: Defining tab and reading order

## Provide Keyboard Access: Moving Focus

- Flash allows the focus to be programmatically moved, until recently screen readers responded to this in “forms” mode only.
  - Demo
- Screen readers maintain an off-screen model which is old versions of screen readers followed instead of the system focus, except in forms mode.
- “Forms mode” is the mode that most interaction with Flex applications will occur with screen readers – ensure that users can enter forms mode.

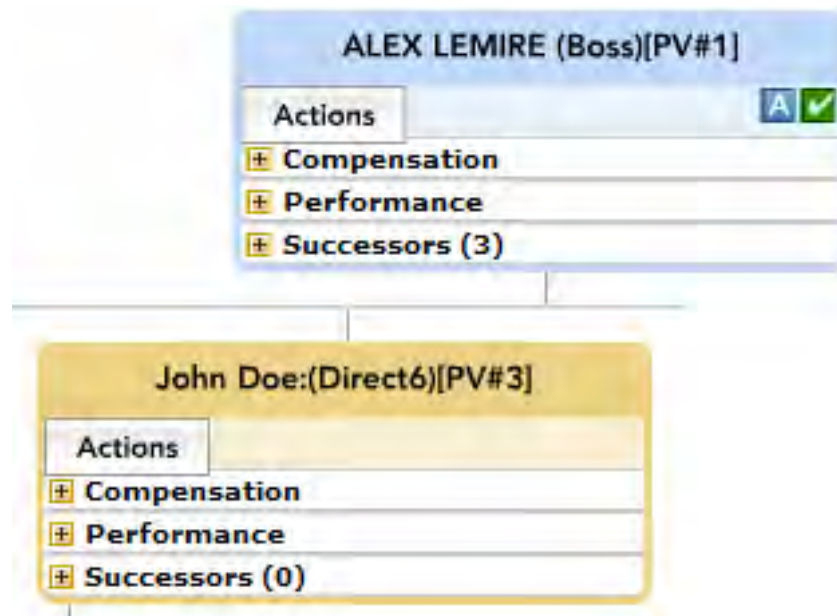
## Provide Keyboard Access: Testing

- Tabbing through controls is a good start
- Make sure to shift+tab backwards
- Learn about expected control behaviors
  - Flex 3 Keyboard Documentation:  
[http://livedocs.adobe.com/flex/3/html/accessible\\_5.html](http://livedocs.adobe.com/flex/3/html/accessible_5.html)
  - Controls:  
<http://msdn.microsoft.com/library/en-us/dnwue/html/ch08c.asp>
  - Guidelines for Keyboard User Interface Design:  
<http://msdn2.microsoft.com/en-us/library/ms971323.aspx>

# Building Accessible Applications

## Convey Relationships

- Communicating what is going on in an application is largely a design issue.
- Assistance can be provided within the application, in the form of a "accessibility information" screen.
- Identifying issues is the most significant challenge.



# Building Accessible Applications

## Provide Closed Captions for Video

- Use captioning tools such as MAGpie or Captionate, with DFXP support
- Flash can parse XML caption data for display





## Provide Instructions

- Use a Help or Site Info screen to provide information about the purpose of the application and accessibility specific instructions
- Provide detail about how to use controls created using components
- Non-obvious functionality needs to be revealed for users

## Test with Assistive Technologies - JAWS Forms Mode

- Screen readers rely on arrow keys to read line by line
- In order to enter data into form fields the user must enter a "Forms Mode"
- Press Enter to go into forms mode on form controls
- This also impacts components such as accordion pane, tree control that require system focus
- Submitting the form will cause the screen reader to exit forms mode
- To manually exit, press Numpad+Plus to exit forms mode. This is not a common use case. (on a laptop you can either use the integrated keyboard or set JAWS up for "laptop" layout and use Caps Lock + semi-colon)

## Detecting Assistive Technologies

- The Flash Player is aware of running MSAA clients.
- Not all MSAA clients are screen readers, so use carefully
- It is **OK** to swap out a complex custom control for a simple standard control (or controls) to accomplish the same task.
- It is **NOT OK** to replace an entire Flex app with a version that has a linear off-screen reading order.
- More: <http://www.paciellogroup.com/blog/?p=61>

## Enlist the Experts

- May be disabled users
- May also be people on your staff who have developed the required skills and sensitivities
- People familiar with accessibility, even if not Flex developers, can provide valuable and time-saving insight into accessibility issues, starting in the design phase.

## User feedback is the final test

- Regular users of assistive technologies will provide the most accurate information
- An accessible application is not necessarily usable – allow time for user testing
- User expectations are a possible issue
  - Many users are not familiar with using RIAs
  - User training may be beneficial

## Implementation details to keep in mind 1

- `Capabilities.hasAccessibility`
  - Does the runtime environment support accessibility?
- `Accessibility.active`
  - Is a screen reader running?
- `AccessibilityProperties` class
  - Six properties worth understanding
- `DisplayObject.accessibilityProperties`
  - Actually only applies to certain classes: `swf`, `container`, `button`, and `text`

## Implementation details to keep in mind 2

- `Accessibility.updateProperties()`
  - Republishes MSAA data
  - Call this after setting one or more `accessibilityProperties`
- `Accessibility.noAutoLabeling`
  - `AutoLabeling` is on by default
- `InteractiveObject.tabIndex`
  - For controlling tab sequence and reading order.
  - Especially important for dynamic applications
- `InteractiveObject.keydown` including `Stage.keyDown`
  - Easy hook for keyboard shortcuts

## What does AccessibilityImplementation Provide?

- Methods for getting and setting roles and states
- Support for object role constants (~60 roles supported)
- Support for object state constants (~30 states supported)
- Methods for setting MSAA focus and selection, labeling relationships
- <http://www.adobe.com/livedocs/flex/3/langref/flash/accessibility/AccessibilityImplementation.html>



# The future of Flash, Flex and AIR

- IAccessible2 support in the next major version of Flash Player
- IAccessible2 support in Flex and AIR
- <http://blogs.adobe.com/accessibility>

# Flash and WCAG 2.0

- Adobe is working with the W3C's WCAG Working Group to develop and approve techniques for Flash
- A set of techniques designed to cover all WCAG 2.0 success criteria is expected to be published on the W3C site in about 1 month

# Resources

- <http://www.adobe.com/accessibility/> (public site)
- <http://blogs.adobe.com/accessibility/> (external blog)