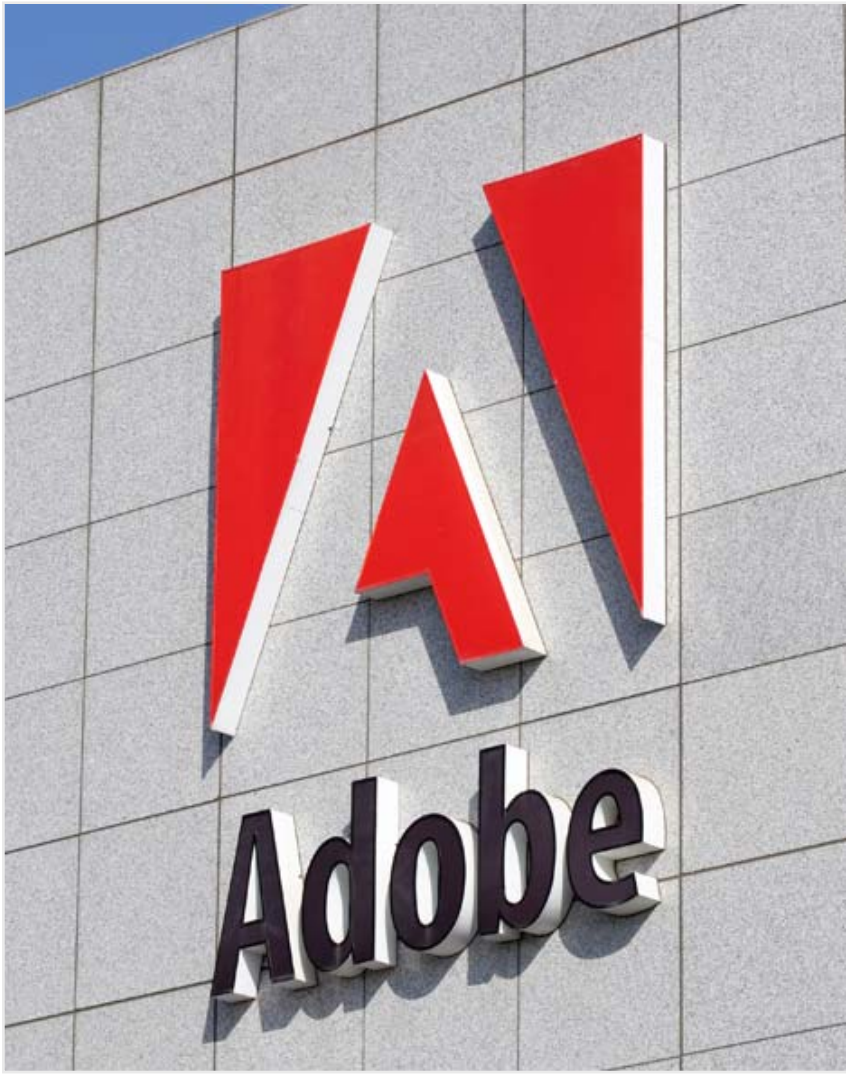
Developing Accessible Flex Applications

Andrew Kirkpatrick

Adobe Systems

akirkpat@adobe.com



Adobe

Agenda

- Introductions
- Motivation for Accessibility
- Planning for Accessibility
- Building Accessible Applications
- Q&A

Adobe

Motivation for Accessibility

"Feel Good" Reasons

- Accessibility is the right thing to do
- Aging population
- Improves application use for everyone
- it means the difference between "impossible to use" and "easy to use" for people who deserve to be considered.

Convincing Skeptics

- Customer requirements
- Legal mandates
- It can be achieved with little additional cost

Motivation for Ignoring Accessibility

Not A Requirement

- But good human factors is an implicit requirement

Too Costly

- 10% incremental cost if done wisely

Technical Uncertainty

- Just a few things to keep in mind

Planning for Accessibility

Minimizing the Cost of Accessibility

- Consider it from project inception
- Address it throughout the project
- Understand that it effects every phase of project

Promote the concept

- A differentiator for the client
- A differentiator for the service provider

Check related policy and legal requirements

- Progressive firms often have a policy that requires this functionality
- Government-related projects typically require it.

Planning for Accessibility

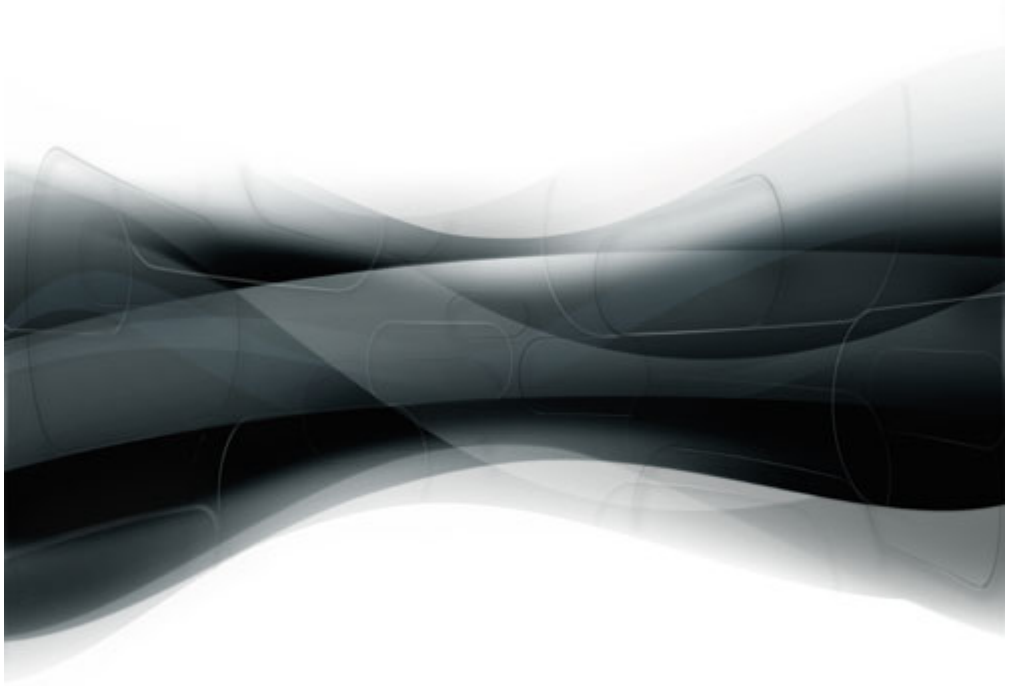
Consider non-mouse-driven modes of use

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

Consider accessibility-related usage scenarios

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

Developing Accessible Applications



Best Practices

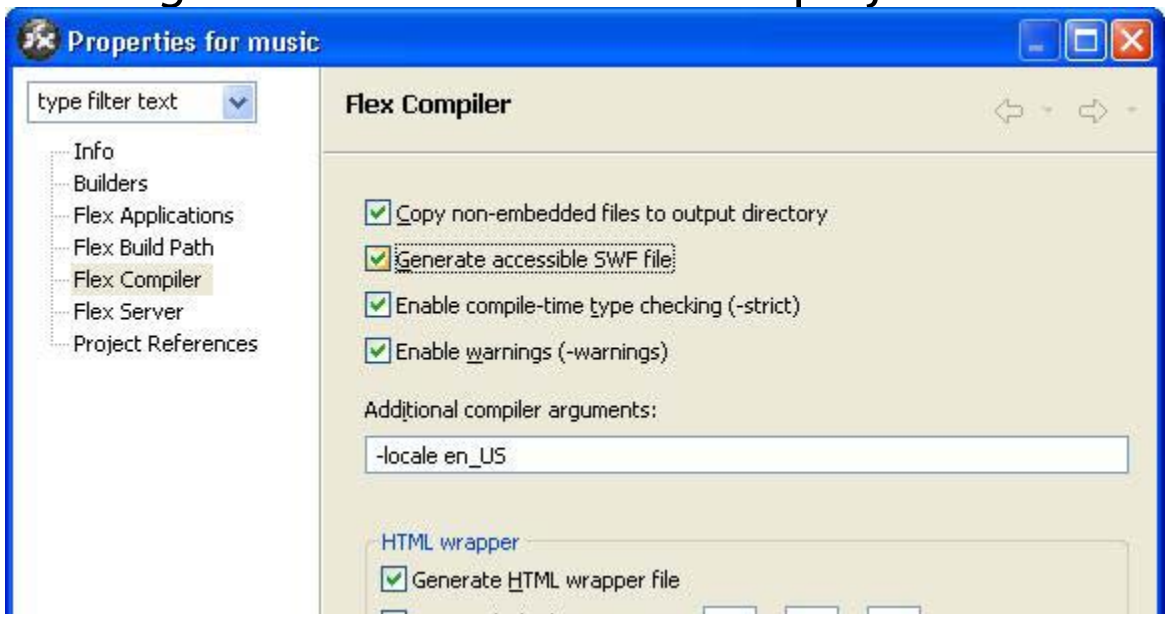
- Enable Component Accessibility
- Provide keyboard access
- Use accessibility-enabled components
- Convey Relationships
- Provide closed captions for video
- Provide instructions
- Enlist the experts

Adobe

Enabling Component Accessibility

In FlexBuilder

- Do this first – make it a habit
- Select "Generate Accessible SWF file" in the Project Properties dialog
- Setting must be modified for each project



Adobe

Enabling Component Accessibility

At runtime

- Append parameter, accessible=true, to URL, e.g.

```
http://www.mydomain.com/index.mxml?accessible=true
```

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

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

Assistive Technology support has multiple dependencies

Assistive technology support for the accessibility API, including any connection to the Flash player's support

Flash player support for an accessibility API - Flash player uses Microsoft Active Accessibility (MSAA)

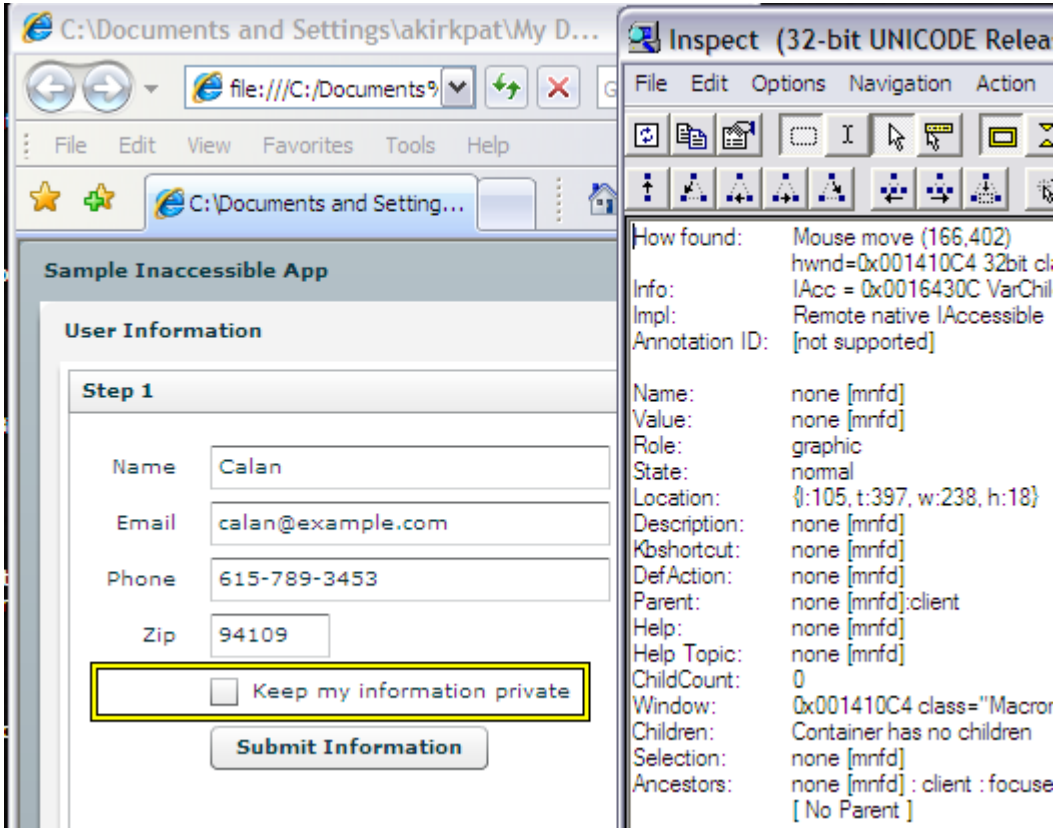
Presence of appropriate and required information within the Flash SWF file.

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
- Presently not documented, look for changes soon.

Enabling Component Accessibility

- Not enabling accessibility is common and easy to identify
- Use Inspect32 to view the role for a non-text component
- A role of “Graphic” indicates accessibility has not been enabled.



Adobe

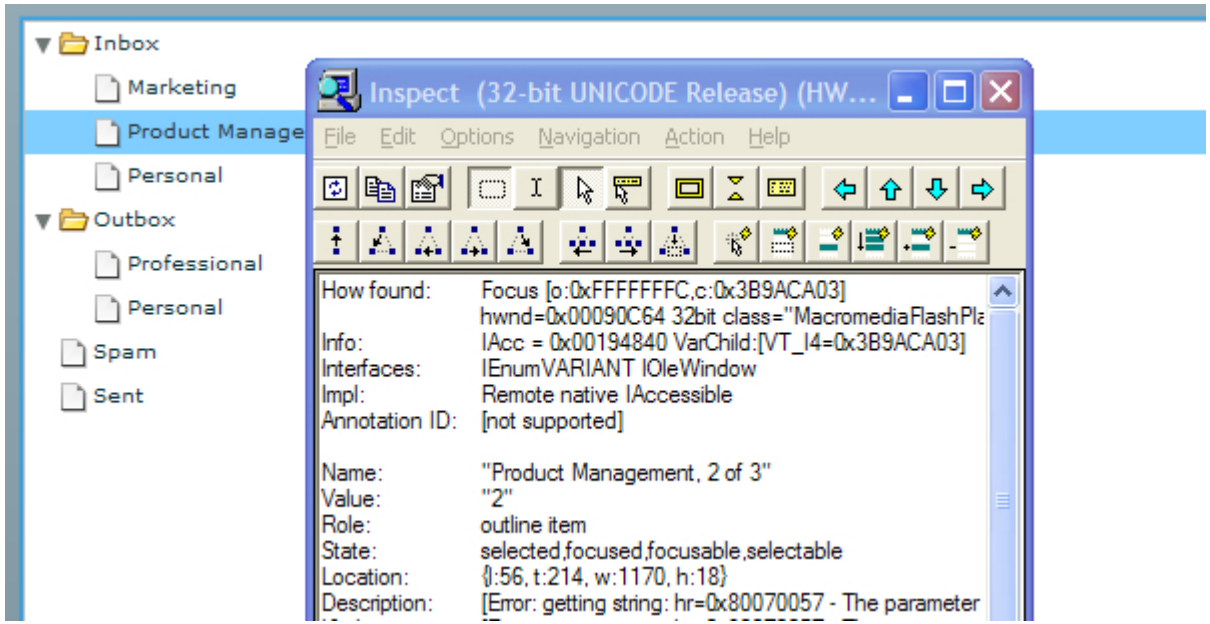
Microsoft MSAA Resources

- Active Accessibility 2.0 Documentation
  - 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

Developing Accessible Applications

Test with MSAA Inspector Tools

- Quick tool for identifying role and state info



Adobe

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
- 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 2.0 Keyboard Documentation: 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

Use Accessibility-Enabled Components

- "Accessible" components are those with built-in MSAA 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

Developing Accessible Applications

28 Accessible Flex Components

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

Developing Accessible Applications

JAWS for Windows

- http://www.freedomscientific.com
- JAWS 4.5, 6.1, 6.2, 7.0, 8.0, 9.0 provide solid support for Flash and Flex content

Flash Components Scripts for JAWS

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

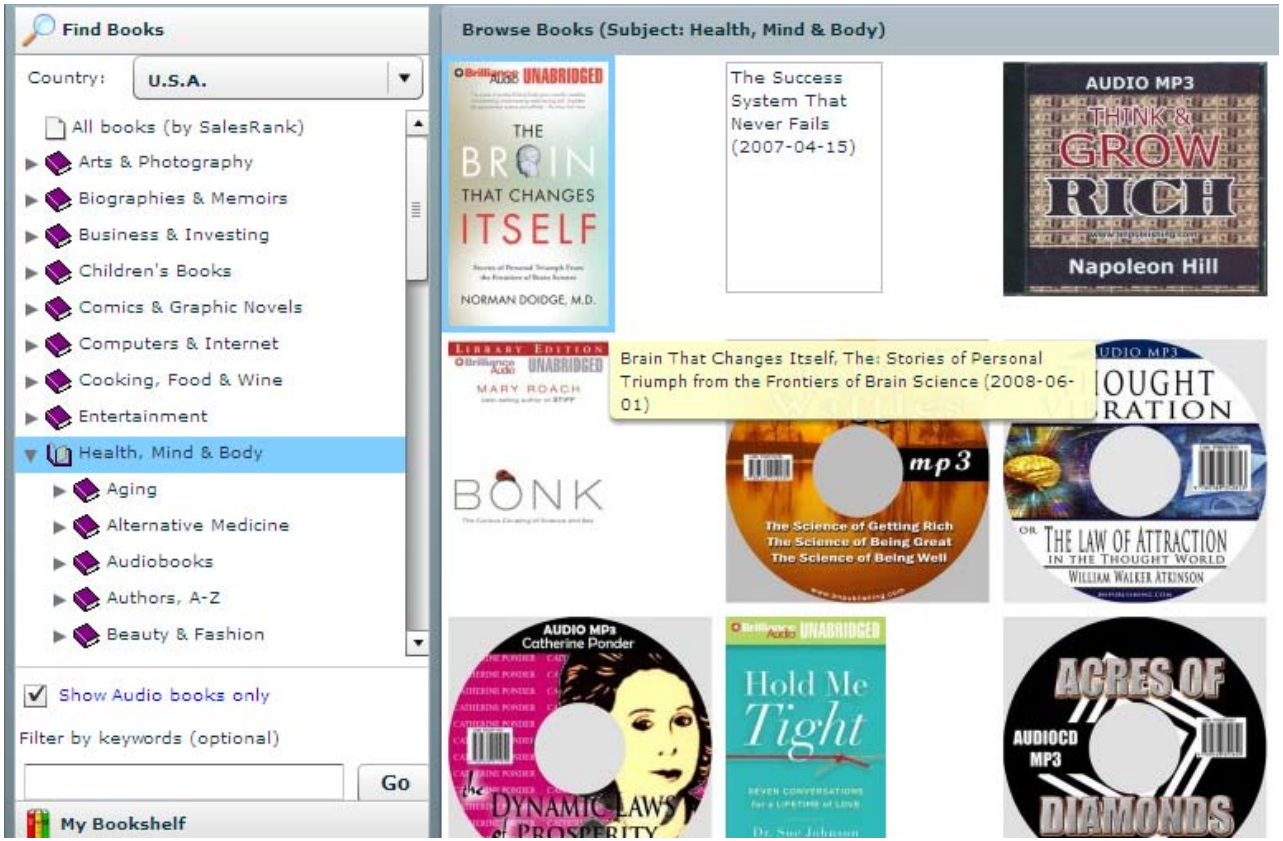
Adobe

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)

Real World Example

- Amit Gupta's E41st App – demonstration
  http://www.amitgupta.info/e41st/

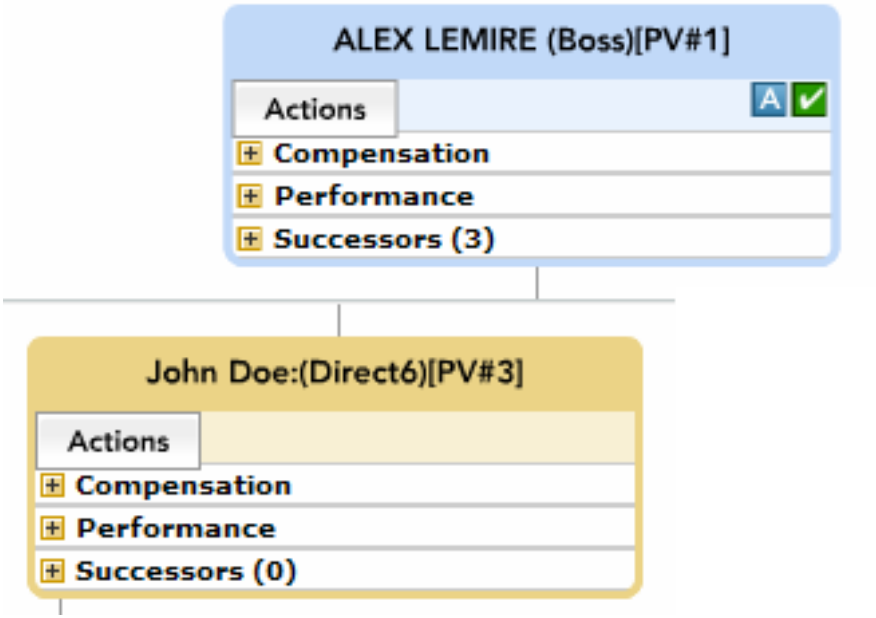


Detecting Assistive Technologies

- 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

Convey Relationships

- Applications that assume comprehension of visual relationships require extra attention.
- 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.



Developing 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
- Demo: Flex Captioning

Adobe

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

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

Q&A

Adobe

Resources

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

Adobe

Better by Adobe™

Adobe