

# Accessible Rich Internet Applications using Dreamweaver CS3 and Flash CS3

Matt May

Andrew Kirkpatrick

Adobe

13 March 2008



# What are Rich Internet Applications?

- “Next-generation Web applications that combine the benefits of desktop software with the reach of the Web to create more engaging user experiences.” – Adobe, 2006

# What are Rich Internet Applications?

- Beyond basic HTML
- UI controls and components written in JavaScript
- Integration between HTML/JS and Flash/Flex
- New ways of interacting with information
  - Maps
  - Calendaring
  - User-user interaction

# Demo: Dreamweaver and Spry

- Creating keyboard-accessible tabbed navigation
- Creating accessible live regions

# What is Flex?

- Language for creating Flash applications
- XML-based syntax
- CSS-based layout
- ActionScript-based behaviors
- Open-source framework
- Open bug tracking
- Flex 3 released 24 February 2008
  - JAWS 8/9 scripts for Flex 3 this month

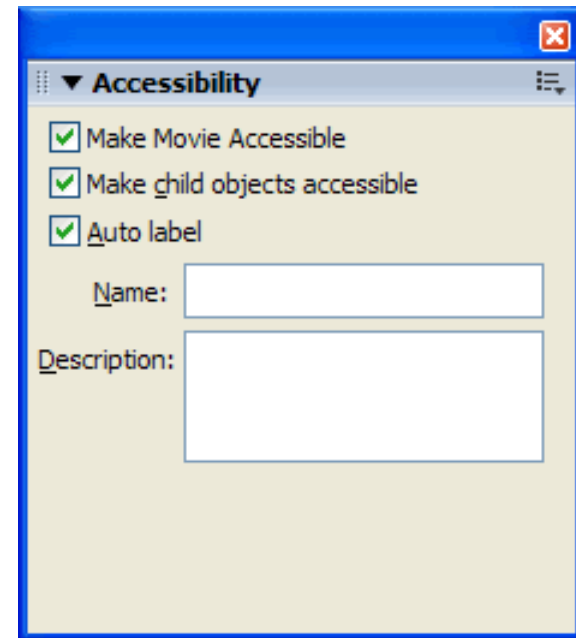
# Flex Accessibility

## 28 Accessible Flex 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

# Flash Accessibility panel

- Assigning name, equivalents, and labels
  - Auto-labeling is enabled by default
  - Enabled for an entire movie
  - Assumes text contained within an object serves as the label
  - Will only use one text object
  - For components, text above or to the left will be read as the label
  - Assigning a .name property value overrides auto-label feature



# Provide Equivalents and Names – ActionScript 2

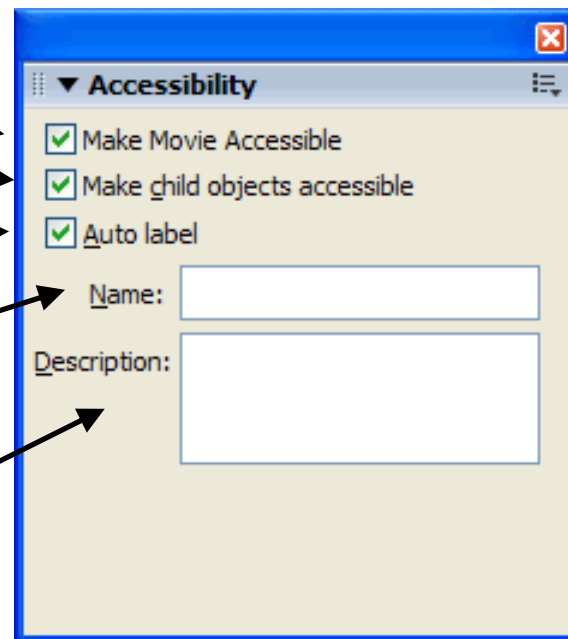
`_accProps.silent`

`_accProps.forceSimple`

`_accProps.noAutolabeling`

`_accProps.name`

`_accProps.description`





# Provide Equivalents and Names – ActionScript 3

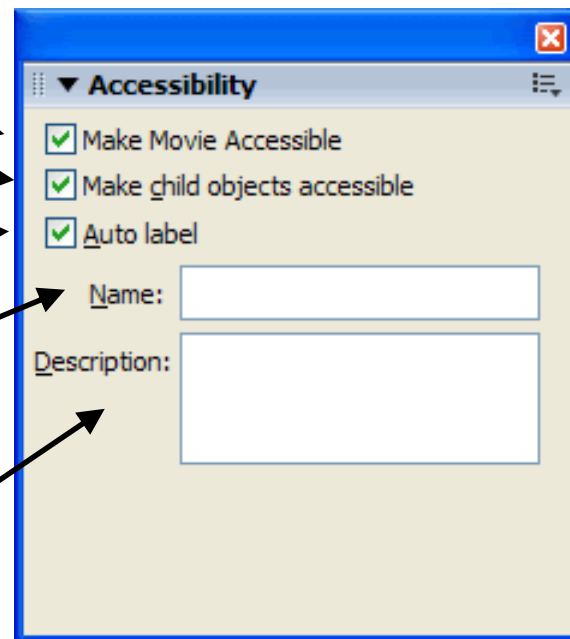
accessibilityProperties.silent

accessibilityProperties.forceSimple

accessibilityProperties.noAutolabeling

accessibilityProperties.name

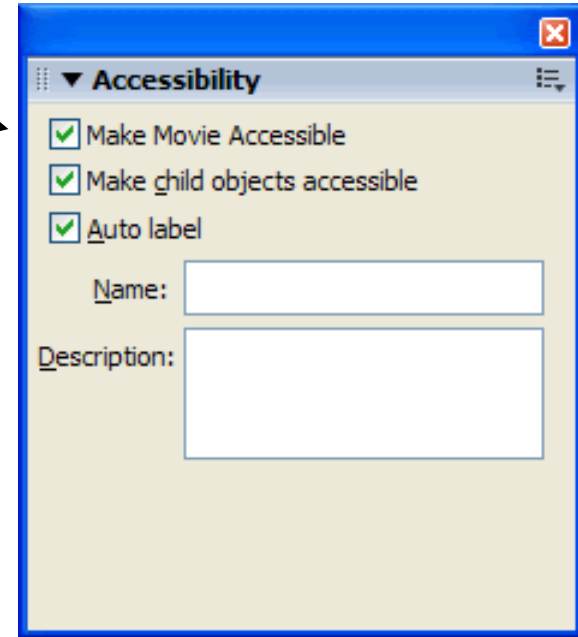
accessibilityProperties.description



# Provide Equivalents and Names – ActionScript 3

`accessibilityProperties.silent`

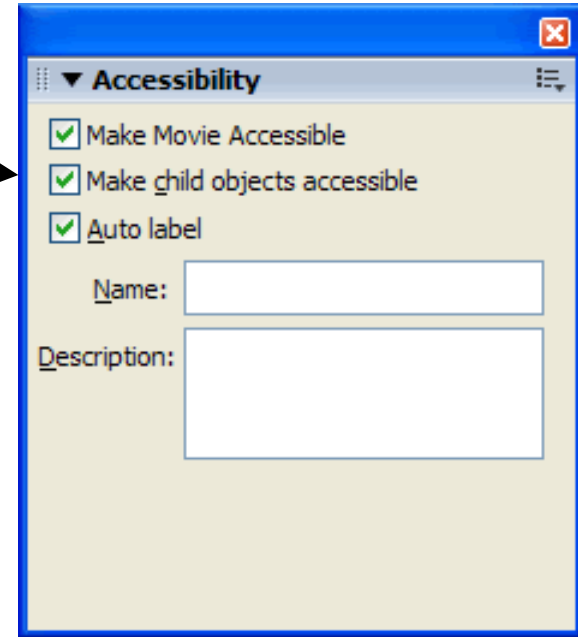
Used to make the movie clip or root movie inaccessible to assistive technologies.



# Provide Equivalents and Names – ActionScript 3

`accessibilityProperties.forceSimple`

Used to hide all child objects for a object, resulting in a single accessible object.

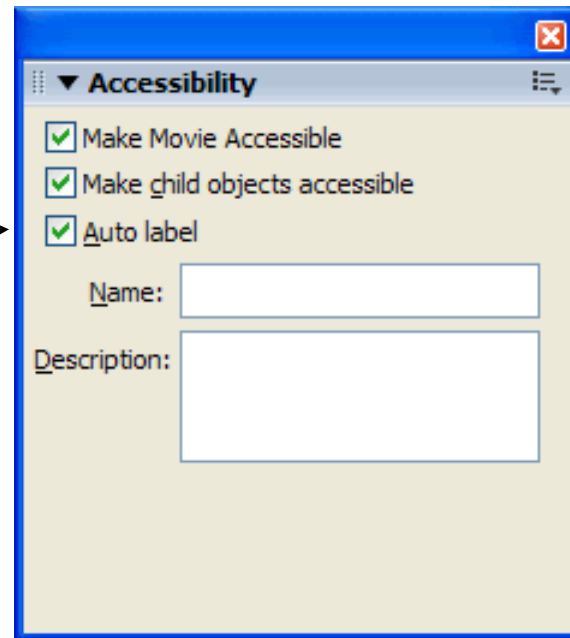


# Provide Equivalents and Names – ActionScript 3

`accessibilityProperties.noAutolabeling`

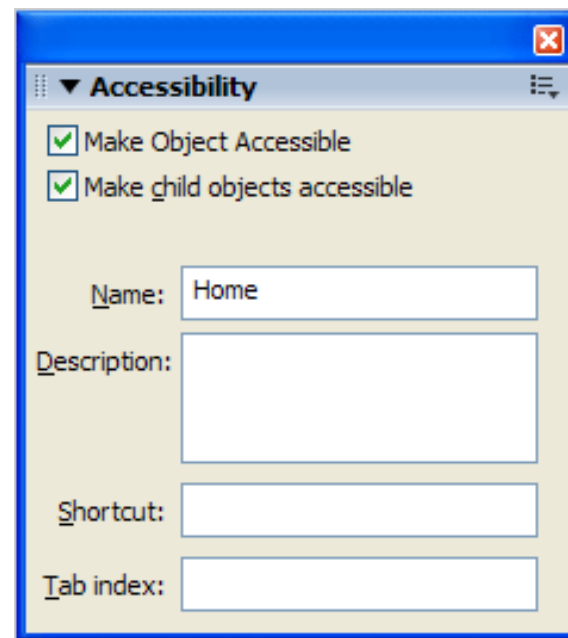
Used to toggle the Flash player's ability to assign accessibility names automatically. If this property is disabled, developers need to assign names for all objects.

Only defined at the root movie level.



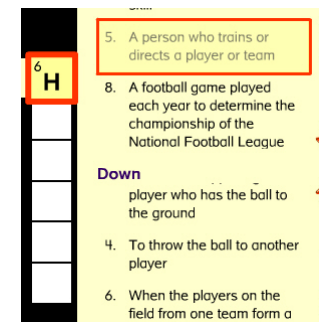
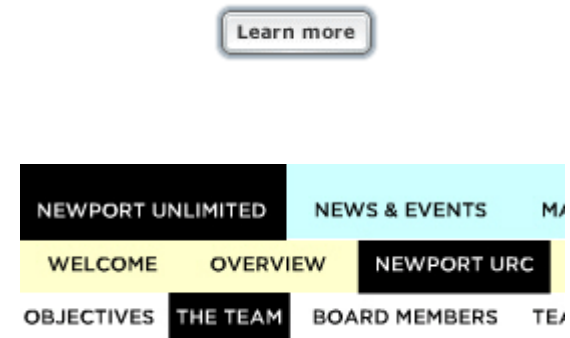
# Provide Equivalents and Names

- Labels can be assigned using the `.name` property.
- There are two ways of assigning the `.name` property.
- First, you can use the Accessibility panel.
- Keep values descriptive of function, not the control itself



# Provide Structure, Role, and State Information

- Screen reader user should know what every control does
  - Buttons must be correctly identified
  - Controls emulating standard windows controls should be identified appropriately
  - Unusual controls should provide cues to users as to their identification, operation and state information
  - Flash CS3's UI components handle role for you.



# Provide Structure, Role, and State Information

- Every control should indicate state:
  - Current selection.
  - Number of possible selections.
  - Update when selection changes.
  - Flash CS3's UI components handle state for you.



# Provide Equivalents and Names (AS2)

- Labels can be assigned using ActionScript
- Three step process
  - Create accessibility object for each instance (once)
  - Assign properties
  - Update accessibility object (once per event)

```
_root.city4_mc._accprops = new Object();  
_root.city4_mc._accprops.name = "Home";  
Accessibility.updateProperties();
```



# Provide Equivalents and Names (AS3)

- Changes in ActionScript 3 affect how accessibility names are defined

```
this.city4_mc.accessibilityProperties = new AccessibilityProperties();  
this.city4_mc.accessibilityProperties.name = "Home";  
Accessibility.updateProperties();
```

# Controlling Tab and Reading Order

**How do we ensure that the content reads in the correct order?**



# Controlling Tab and Reading Order

## Ensure objects are read in a sensible order

- The tab order for a Flash or Flex application is used to determine the order of the items in the MSAA tree, which is used for the reading order
  - The MSAA tree is used in the same way that JAWS uses the DOM for HTML reading order.
- Use MSAA inspector tools and a screen reader to:
  - ensure labels are read before controls
  - ensure any instructions for controls are read before controls themselves

# Controlling Tab and Reading Order

## **Controlling reading order is different than controlling what the user reads**

- Flash allows the focus to be programmatically moved, screen readers will respond to this in “forms” mode only.
- Screen readers maintain an off-screen model which is followed instead of the system focus, except in forms mode.

# Captions and Subtitles

- Flash CS3 enables easy-to-implement captioning support
  - FLVPlaybackCaptioning component
- Support for W3C's DFXP Timed Text format built in.
- Includes support for captioning full screen video.
- Captioning session on Friday

# Provide Structure, Role, and State Information

## Enabling Flash components

- **Button**

```
import fl.accessibility.ButtonAccImpl;  
ButtonAccImpl.enableAccessibility();
```

- **Check Box**

```
import fl.accessibility.CheckBoxAccImpl;  
CheckBoxAccImpl.enableAccessibility();
```

- **Radio Button**

```
import fl.accessibility.RadioButtonAccImpl;  
RadioButtonAccImpl.enableAccessibility();
```

- **Combo Box**

```
import fl.accessibility.ComboBoxAccImpl;  
ComboBoxAccImpl.enableAccessibility();
```

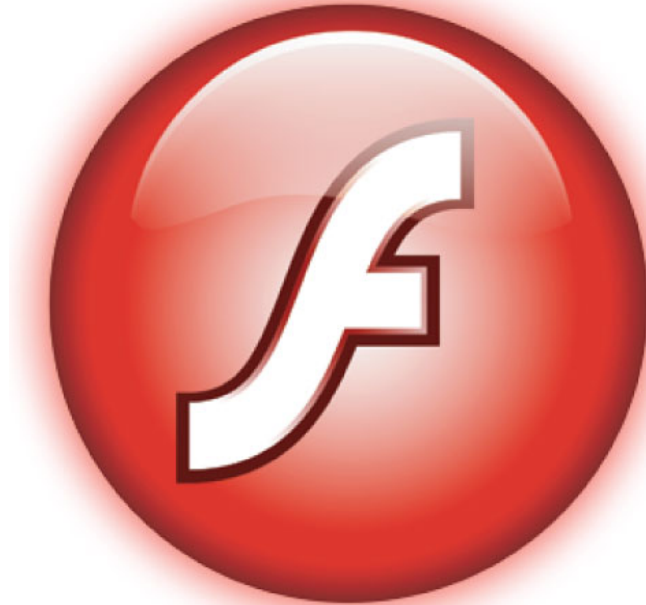
- **List**

```
import fl.accessibility.ListAccImpl;  
ListAccImpl.enableAccessibility();
```

- **Tile List**

```
import fl.accessibility.TileListAccImpl;  
TileList.enableAccessibility();
```

# Demo: Flash accessibility



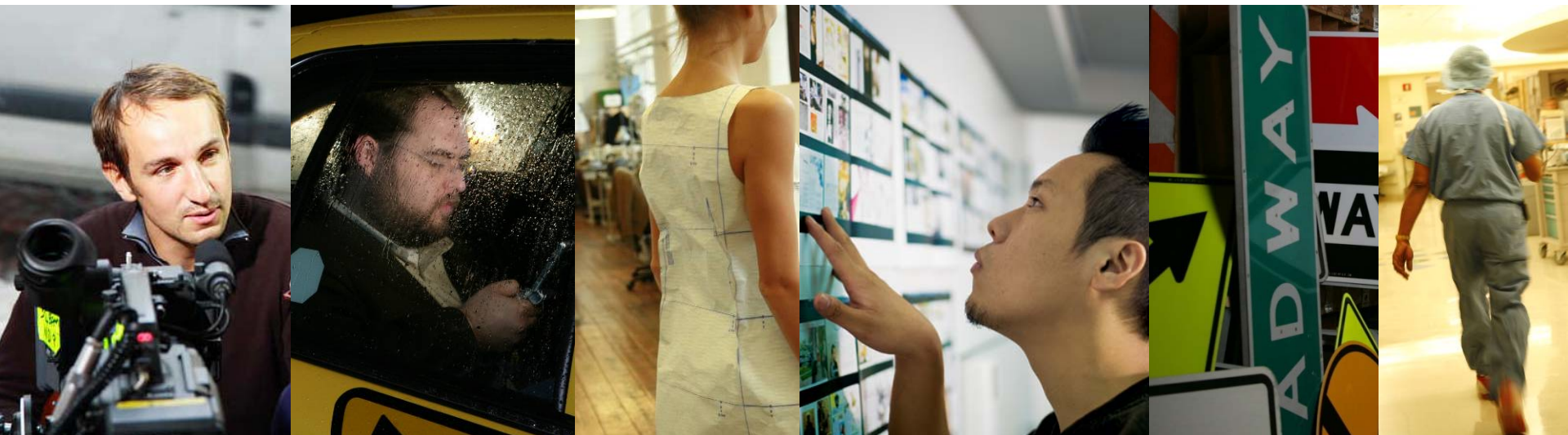
- Using components and accessibilityProperties

# Thanks!

- <http://adobe.com/accessibility/>
- <http://blogs.adobe.com/accessibility/>

## Q&A





**Revolutionizing**  
how the world engages  
with ideas and information

