



**Beyond Basic Access**

Andrew Kirkpatrick  
Adobe Systems  
March 26, 2007



2007 Adobe Systems Incorporated. All Rights Reserved.



---

---

---

---

---

---


---

---

**Accessibility is not simple**

*Accessibility is more than @alt,  
<label>, and <th>*

2007 Adobe Systems Incorporated. All Rights Reserved.



---

---

---

---

---

---


---

---

**Accessibility is not simple**

*Accessibility is more than retaining  
information when JavaScript is disabled*

2007 Adobe Systems Incorporated. All Rights Reserved.



---

---

---

---

---

---

---

---

Accessibility is not simple

*Accessibility is more than providing access to the basic information on a page or in a web app*



---

---

---

---

---

---

---

---

Accessibility is not simple

*Accessibility requires that we better understand user needs*



---

---

---

---

---

---

---

---

Accessibility is not simple

*Accessibility requires that we get more sophisticated in our view of the web.*



---

---

---

---

---

---


---

---



**The Basics**

Name	Role	State
<ul style="list-style-type: none"> <li>"What is this object called?"</li> </ul>	<ul style="list-style-type: none"> <li>"What kind of object is this?"</li> </ul>	<ul style="list-style-type: none"> <li>"What are the current settings for this object?"</li> </ul>

2007 Adobe Systems Incorporated. All Rights Reserved. 10 

---

---

---

---

---

---


---

---

**Name**

Name is already a familiar concept, and easy to address

- Not the same as
  - `<input name="myText" ... />`
- A `<label>` is used to define the name for a form control
- An alt attribute value is used for the name of an image
- In some cases the text of an object is used for the name
  - `<h2>Products</h2>`
- In some cases the name has a composite origin
  - `<a href="x.htm">Product X</a>`

2007 Adobe Systems Incorporated. All Rights Reserved. 11 

---

---

---

---

---

---


---

---

**Role**

Several familiar roles

- Graphic
- Checkbox
- Text
- Radio Button
- Grouping ( `<fieldset>` )
- List
- Combo Box
- Button
- Text Area

2007 Adobe Systems Incorporated. All Rights Reserved. 12 

---

---

---

---

---

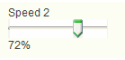
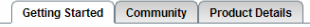
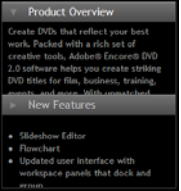
---

---

---

**Role**

Examples where role is not conveyed

- Slider
 
- Tab
 
- Accordion
 

2007 Adobe Systems Incorporated. All Rights Reserved. 13

---

---

---

---

---

---

---

---

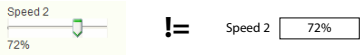
---

---

**Role**

Is a text input field equivalent to a slider?

- We need to communicate roles and provide expected behaviors to provide an equivalent experience for screen reader, screen magnifier, and keyboard users.



2007 Adobe Systems Incorporated. All Rights Reserved. 14

---

---

---

---

---

---

---

---

---

---

**Role**

*How we can convey role today?*

2007 Adobe Systems Incorporated. All Rights Reserved. 15

---

---

---

---

---

---

---

---

---

---

## Back to... Name

Exploiting the name for other purposes is sometimes needed.

- Assistive technology support for name is solid
- Augmenting the name for an object is easy
- Use standard controls whenever possible

*In the absence of a sufficiently rich API,  
overloading an object's name may be necessary*



---

---

---

---

---

---

---

---

## Role

The functional solutions are simple

```
<label for="mySlider">Delay (sec)
<span class="roleinfo">Horizontal Slider</span>
</label>
<div id="mySlider"> ...
</div>
```

**CSS**  
`.roleinfo {margin-left: -999em}`

or

```
<label for="mySlider">Delay (sec)<img alt="Horizontal
Slider" ... /></label>
<div id="mySlider"> ...
</div>
```

or

```
<div id="mySlider" title="Horizontal Slider" ...>
```



---

---

---

---

---

---

---

---

## Role

User agents are limited in how they handle composite names

```
<a href="z.htm">
Product X</a>
```

- Current screen readers don't read all items within an anchor element, usually just two.



---

---

---

---

---

---

---

---

## Role

- It is important to test that when the user or user agent focuses the control that the role information is read.
- The Yahoo! UI slider widget provides a good example



---

---

---

---

---

---

---

---

## Future directions for conveying role



- The W3C's WAI-ARIA (Accessible Rich Internet Applications) Roadmap includes a means to convey roles not defined in HTML.
- <http://www.w3.org/TR/aria-roadmap/>
- For example:

```
<div wairole="menu">
```



---

---

---

---

---

---

---

---

## State

The state of a control is equally important

- Is the control disabled?
- Is a list item selected?
- Is the checkbox checked?
- Is the control open?
- Is this tab the active tab?



---

---

---

---

---

---

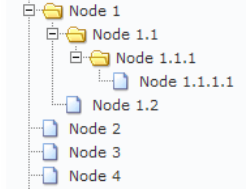
---

---

**State**

Tree controls provide a good example of a control that conveys a lot of state information

- Folder or file selected?
- Folder open?
- Level in tree?
- Tree enabled or disabled?



---

---

---

---

---

---

---

---

**Role**

*How we can convey state today?*

---

---

---

---

---

---

---

---

**State**

How to convey state today

- Use standard controls whenever possible
- Utilize overloading strategy as described for role.

---

---

---

---

---

---

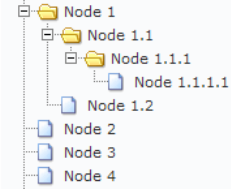
---

---

## State

### Sample solution – assumes JavaScript

```
<div id="myTree">
<ul>
  <li>
    <a href="javascript:closeNode();return false;">
      
      
    </a>
    <a href="node1.html">Node 1</a>
  </li>
  ...
</ul>
```



---

---

---

---

---

---

---

---

---

---

## Future directions for conveying state



WAI-ARIA has a module for state information

- <http://www.w3.org/TR/aria-state/>

---

---

---

---

---

---

---

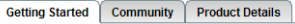
---

---

---

## A Sample Set of Tabs

### Common Coding Practice



```
<h2 class="tabtitle" id="getstarted">Getting Started</h2>
<div class="tabcontent">
  <p>This is the content for the Getting started tab</p>
</div>
<h2 class="tabtitle" id="community">Community</h2>
<div class="tabcontent">
  <p>This is the content for the Community tab</p>
</div>
<h2 class="tabtitle" id="details">Product Details</h2>
<div class="tabcontent">
  <p>This is the content for the Product Details tab</p>
</div>
```

---

---

---

---

---

---

---

---

---

---

## A Sample Set of Tabs

### Possible Solutions



1. Hide content for inactive tabs in a way that screen readers can read (e.g. `visibility:hidden` or move off-screen to the left).
  - Screen reader users will hear "clickable" for tabs - links that don't go anywhere.
  - Users can navigate semantic content by headings.
  - The longer or more complex the tab's content, the less association will be made between the present tab's content and the other options.
  - Keyboard users can't activate other tabs

2007 Adobe Systems Incorporated. All Rights Reserved.

28



---

---

---

---

---

---

---

---

---

---

## A Sample Set of Tabs

### Possible Solutions



2. Hide content for inactive tabs in a way that screen readers can *not* read (e.g. `display:none` added as an inline style). Add anchor to switch tabs and extra image to each heading to indicate role and state information.

```

```

- Screen reader users will activate tab sections to read
- Keyboard users can activate other tabs
- Users only need to read the information that they want to read
- Need to deal with screen reader refreshing page model

2007 Adobe Systems Incorporated. All Rights Reserved.

29



---

---

---

---

---

---

---

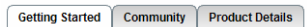
---

---

---

## A Sample Set of Tabs

### Common Coding Practice + Role + State



```
<h2 class="tabtitle" id="getstarted">Getting Started
</h2>
<div class="tabcontent">
  <p>This is the content for the Getting started tab</p>
</div>
<!-- add the anchor in remote script, but like below -->
<h2 class="tabtitle" id="community">
<a href="javascript:clicktab();return false;">Community
</a></h2>
<div class="tabcontent">
  <p>This is the content for the Community tab</p>
</div>
```

2007 Adobe Systems Incorporated. All Rights Reserved.

30



---

---

---

---

---

---

---

---

---

---

## A Sample Set of Tabs

What is left to do?



- Provide keyboard support expected for a set of tabs
  - Left/right and pgup/pgdn to move between tabs
  - Home/end to move to first/last tab
- When users see or hear that they are interacting with a certain control type, they will expect certain behaviors.
- Notify screen reader that the page has changed
  - Toggling an element's display property to "none" and back to "block"
  - Changing or adding a class name on an element



---

---

---

---

---

---

---

---

## Moving Forward

Next Steps for Developers

- Consider Name, Role, and State constantly
  - Easiest way to gain an understanding of user needs
- Keyboard behaviors are worth learning
  - Controls:  
<http://msdn.microsoft.com/library/en-us/dnwue/html/ch08c.asp>
  - Guidelines for Keyboard User Interface Design:  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnacc/html/ATG\\_KeyboardShortcuts.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnacc/html/ATG_KeyboardShortcuts.asp)
- It's never too late to familiarize yourself with assistive technologies
  - Screen readers and Screen Magnifiers
  - Some exposure to assistive technologies for mobility-impaired users



---

---

---

---

---

---

---

---

## Beyond Basic Accessibility

**Thank You**



---

---

---

---

---

---

---

---



---

---

---

---

---

---

---

---