

Dijit Accessibility



Simon Bates - ATRC

Becky Gibson - IBM

Accessibility Overview



- ✓ Cognitive
- ✓ Hearing
- ✓ Mobility
- ✓ Vision

Cognitive



Issues

- ✓ Learning disabilities
- ✓ Distractions
- ✓ Time limits
- ✓ Seizures

Technologies

- ✓ Specific color schemes
- ✓ Screen readers
- ✓ Close captioning
- ✓ Spell and grammar checkers

Hearing



Issues

- ✓ Deaf
- ✓ Hearing impairments

Technologies

- ✓ Captioning
- ✓ Signed captioning
- ✓ Increased volume

Mobility



Issues

- ✓ Limited mouse movement
- ✓ Keyboard only access
- ✓ No use of hands

Technologies

- ✓ Simple mouse navigation
- ✓ Keyboard only navigation
- ✓ Single click devices
- ✓ On screen keyboard
- ✓ Touch screen
- ✓ Speech recognition
- ✓ Screen magnification
- ✓ Large fonts

Vision



Issues

- ✓ Blind
- ✓ Low vision
- ✓ Color blindness

Technologies

- ✓ Keyboard only navigation
- ✓ Screen readers
- ✓ Braille displays
- ✓ Large fonts
- ✓ High contrast mode
- ✓ Screen magnifiers
- ✓ Specific color schemes

Web 2.0 A11y Concerns

- ✓ Rich Interface controls
 - ✓ Reliance on Mouse
 - ✓ Lack of Semantics
- ✓ Incremental Updates via Ajax
- ✓ Changes in focus
- ✓ Excessive Navigation via Tab key
- ✓ Content aggregation from various resources (Mashups)
- ✓ Multimedia
 - ✓ Lack of captioning
 - ✓ Interaction issues
- ✓ Paradigm Shift
- ✓ Outdated Accessibility Standards

Addressing Accessibility



- ✓ Full keyboard support
- ✓ Low vision support
 - ✓ High contrast mode & CSS Background Images
 - ✓ Font resizing
- ✓ Assistive Technology Support with ARIA
 - ✓ Screen reader
 - ✓ Screen magnifier
 - ✓ Alternative input

Accessible Rich Internet Applications (ARIA)



- ✓ Add role semantics to scripted user interface (UI) elements
- ✓ Update state information dynamically
- ✓ Make items focusable via tabindex attribute
- ✓ Add keyboard event handling
 - ✓ Mimic the keyboard behavior of the rich client UI
 - ✓ Minimize tab key navigation
- ✓ Add live region information and notification behavior*

*Live region support coming in Firefox 3

A11y in Dojo



- ✓ ARIA Specification Implemented
 - ✓ Fully keyboard accessible in Firefox and Internet Explorer
 - ✓ Screen reader accessible in Firefox
- ✓ Work in Operating System High Contrast mode
- ✓ Work with images off
 - ✓ Currently only auto-detected in Firefox
 - ✓ Must be in high contrast mode for image off support in IE
- ✓ No support for Drag and Drop
 - ✓ Drag and drop functionality must be duplicated via alternate mechanism

Dijit A11y Demo



Pre-recorded demos available at

<http://www.weba11y.com/demos/index.html>

Dijit Accessibility



How we did it and how you can, too!

Dojo/Dijit A1 1y Infrastructure



- ✓ Navigation / Keyboard Support
- ✓ ARIA Support
- ✓ High Contrast Detection and Support
- ✓ Documentation
- ✓ Testing Accessibility

Navigation / Keyboard Support



- ✓ Extended tabindex (IE, FF, Opera 9.5, HTML5)
- ✓ Keyboard event normalization
- ✓ Keyboard navigation widget mixin
- ✓ Functions for finding elements in the tab order

Extended tabindex

- ✓ HTML4: form controls and hyperlinks
- ✓ IE, Firefox, Opera 9.5, HTML5: all elements
- ✓ no tabindex as HTML4
- ✓ `tabindex = -1` programmatic, mouse
- ✓ `tabindex = 0` +tab, document order
- ✓ `tabindex > 0` +tab, numerical order

Extended tabindex



```
<div tabindex="0">  
  <div tabindex="-1">item 1</div>  
  <div tabindex="-1">item 2</div>  
  <div tabindex="-1">item 3</div>  
</div>
```

Extended tabindex



- ✓ In templates
- ✓ API (new in 1.1)
- ✓ `dojo.hasAttr(elem, "tabindex ")`
- ✓ `dojo.attr(elem, "tabindex")`
- ✓ `dojo.attr(elem, "tabindex", -1)`
- ✓ `dojo.removeAttr(elem, "tabindex")`

Normalized onkeypress event




- ✓ `event.charCode` for printable
- ✓ `event.keyCode` for non-printable
- ✓ `dojo.keys.LEFT_ARROW`

ondijitclick event



- ✓ Consistent clicking across controls and browsers
- ✓ Click
- ✓ Enter-down
- ✓ Space-up

Attaching events in templates



```
<table class="dijit dijitMenu dijitReset
dijitMenuTable" waiRole="menu"
dojoAttachEvent="onkeypress:_onKeyPress">

<tr class="dijitReset dijitMenuItem"
dojoAttachEvent="onmouseenter:_onHover,onmouseleave
:_onUnhover,ondijitclick:_onClick">
```

_KeyNavContainer mixin



```
dojo.declare("dijit.Menu", [dijit._Widget,
dijit._Templated, dijit._KeyNavContainer],
{
  postCreate: function(){
    this.connectKeyNavHandlers(
      [dojo.keys.UP_ARROW], [dojo.keys.DOWN_ARROW]);
  },

  startup: function(){
    this.startupKeyNavChildren();
  },
}
```

Finding elements in the tab order



in dijit.Dialog._setup():

```
var focusItem
  = dijit.getFirstInTabbingOrder(this.domNode);
this._firstFocusItem
  = focusItem ? focusItem : this.domNode;
```

in dijit.Dialog.onOpen():

```
dijit.focus(this._firstFocusItem);
```

ARIA Support



- ✓ Setting roles and states in templates
 - ✓ `waiRole="menuitem"`
 - ✓ `waiState="haspopup-true"`
- ✓ Dynamically set and update roles and states
 - ✓ `dijit.setWaiRole/State()`
 - ✓ `dijit.getWaiRole/State()`
 - ✓ `dijit.hasWaiRole/State()`
 - ✓ `dijit.removeWaiRole/State()`

Role and states in templates

```
<div class="dijitDialog" tabindex="-1"
waiRole="dialog" waiState="labelledby- $\{id\}$ _title">
  <div dojoAttachPoint="titleBar">
    <span id=" $\{id\}$ _title"> $\{title\}$ </span>
    <span class="dijitDialogCloseIcon"
dojoAttachEvent="onclick: onCancel">
      <span dojoAttachPoint="closeText"
class="closeText">x</span>
    </span>
  </div>
  <div dojoAttachPoint="containerNode"
class="dijitDialogPaneContent"></div>
</div>
```

dijit.setWaiState()



From `dijit.ProgressBar.update()`:

```
dijit.setWaiState(this.internalProgress,  
    "valuenow", this.progress);  
dijit.setWaiState(this.internalProgress,  
    "valuemin", 0);  
dijit.setWaiState(this.internalProgress,  
    "valuemax", this.maximum);
```

High Contrast Detection & Support



- ✓ Detect high contrast mode
- ✓ Detect images off mode (FF only)
- ✓ dijit_a11y class to trigger styles
- ✓ For example to provide text alternatives for CSS Background images
- ✓ No hard coded font sizes

High contrast with CSS

```
<div class="dijitDialog" tabindex="-1"
waiRole="dialog" waiState="labelledby-#{id}_title">
  <div dojoAttachPoint="titleBar">
    <span id="#{id}_title">#{title}</span>
    <span class="dijitDialogCloseIcon"
dojoAttachEvent="onclick: onCancel">
      <span dojoAttachPoint="closeText"
class="closeText">x</span>
    </span>
  </div>
  <div dojoAttachPoint="containerNode"
class="dijitDialogPaneContent"></div>
</div>
```

High contrast with CSS



```
.tundra .dijitDialogCloseIcon {  
    background : url("images/tabClose.png");  
}  
  
.dijitDialog .closeText {  
    display:none;  
}  
  
.dijit_a11y .dijitDialog .closeText {  
    display:inline;  
}
```

Accessibility in the Book of Dojo



- ✓ A11y documentation on each widget
- ✓ General A11y documentation
- ✓ Creating & Testing Widgets doc

Testing Accessibility



- ✓ Use without mouse
- ✓ Assistive technologies
- ✓ Windows high contrast mode
- ✓ Testing Tools
 - ✓ Inspect
 - ✓ ACCEvent
 - ✓ ACCExplorer

Changes for Firefox 3



- ✓ New ARIA markup
- ✓ ARIA datatype property
- ✓ divs are exposed, can use role="presentation"

Resources



- ✓ Accessibility Section of Dojo Book
 - ✓ <http://www.dojotoolkit.org/book/dojo-book-0-9/part-2-dijit/a11y>
- ✓ ARIA Roadmap
 - ✓ <http://www.w3.org/TR/aria-roadmap/>
- ✓ Mozilla Developer Center - Firefox ARIA Information
 - ✓ http://developer.mozilla.org/en/docs/Accessible_DHTML
- ✓ IBM Human Ability & Accessibility Center
 - ✓ <http://www.ibm.com/able>
- ✓ Adaptive Technology Resource Centre
 - ✓ <http://atrc.utoronto.ca/>