

# Accessibility Automated Testing

Grasp ideas floating in the air

What to test

Web

Web

Browsers

WebApps

Web

Browsers

Desktop / Mobile Apps  
OS

How to test

Unit testing



# Accessibility APIs

# Accessibility APIs

## Spec compliance

- ARIA mapping spec
- HTML mapping spec
- SVG mapping spec

# Accessibility APIs

- States/properties
- Methods/interfaces
- Relations
- Tree (hierarchies)
- Text
- Selection
- Actions
- Events

# Accessibility APIs

Low level

Building blocks testing

AT testing

Test flow

# Test flow

- Query
- Change or trigger action
- Receive event

# Test flow

- User actions or accessible actions
- Eventing or wait until



What we have now

AOM

Cross platform testing

# Gecko

- JavaScript based
- Mature cross-platform testing
- Mac NSAccessibility native API

# Gecko: Mac example

```
addAccessibleTask(`<div id="section" role="group">Hello</div>`, async (browser, accDoc) => {
  let section = getNativeInterface(accDoc, "section");
  ok(!section.getAttributeValue("AXElementBusy"), "section is not busy");
  let busyChanged = waitForMacEvent("AXElementBusyChanged", "section");
  await SpecialPowers.spawn(browser, [], () => {
    content.document
      .getElementById("section")
      .setAttribute("aria-busy", "true");
  });
  await busyChanged;
  ok(section.getAttributeValue("AXElementBusy"), "section is busy");
});
```

# Webkit

- JavaScript based
- Mac NSAccessibility native API
- ATK and MSAA support

# Webkit: an example

```
// test
```

```
if (window.accessibilityController) {  
  // aria-state=false  
  document.getElementById("box1").focus();  
  shouldBe("accessibilityController.focusedElement.intValue", "0");  
}
```

```
// expected file
```

```
PASS accessibilityController.focusedElement.intValue is 0
```

# WebKit: eventing example

```
// test
```

```
function notifyCallback(element, notification) {  
    if (notification == "AXLoadComplete")  
        finishJSTest();  
}  
if (window.accessibilityController) {  
    jsTestIsAsync = true;  
    accessibilityController.enableEnhancedAccessibility(true);  
    var addedNotification = accessibilityController.addNotificationListener(notifyCallback);  
    shouldBe("addedNotification", "true");  
}
```

```
// expected file
```

```
PASS addedNotification is true
```

# Chromium

- Unit tests
- Browser tests
- Platform tests



# Chromium: tree dump example

```
<!--  
@MAC-ALLOW:AXSelected=*  
  
@MAC-ALLOW:AXSelectedChildren=*  
  
@WIN-ALLOW:SELECTED  
  
@AURALINUX-ALLOW:select*  
  
-->  
  
<html><body><div role="listbox">  
  <div tabIndex="-1" aria-selected="true" aria-label="1" role="option">Item 1</div>  
  <div tabIndex="-1" aria-selected="false" aria-label="2" role="option">Item 2</div>  
</div></body></html>
```

# Chromium: tree dump example

```
// Mac expectation file
```

```
AXWebArea AXSelected=0
```

```
++AXList AXSelected=0 AXSelectedChildren=[ :3]
```

```
++++AXStaticText AXSelected=1 AXValue='1'
```

```
++++AXStaticText AXSelected=0 AXValue='2'
```

# Chromium: events example

```
<select size=3>
  <option selected>Apple</option>
  <option>Orange</option>
  <option>Banana</option>
</select>

<script>

document.querySelector('select').focus();

function go() {
  document.querySelector('select').selectedIndex = 1;
}

</script>
```

# Chromium: events example

```
// Mac expectation file
```

```
AXFocusedUIElementChanged on AXStaticText AXValue="Orange"
```

```
AXSelectedChildrenChanged on AXList
```

```
// MSAA/IA2 expectation file
```

```
EVENT_OBJECT_FOCUS on <option> role=ROLE_SYSTEM_LISTITEM name="Orange" SELECTED,FOCUSED,FOCUSABLE,SELECTABLE PosInSet=2 SetSize=3
```

```
EVENT_OBJECT_SELECTION on <option> role=ROLE_SYSTEM_LISTITEM name="Orange" SELECTED,FOCUSED,FOCUSABLE,SELECTABLE PosInSet=2 SetSize=3
```

```
EVENT_OBJECT_SELECTIONWITHIN on <select> role=ROLE_SYSTEM_LIST FOCUSABLE IA2_STATE_VERTICAL SetSize=3
```

```
EVENT_OBJECT_STATECHANGE on <option> role=ROLE_SYSTEM_LISTITEM name="Apple" FOCUSABLE,SELECTABLE PosInSet=1 SetSize=3
```

```
EVENT_OBJECT_STATECHANGE on <option> role=ROLE_SYSTEM_LISTITEM name="Orange" SELECTED,FOCUSED,FOCUSABLE,SELECTABLE PosInSet=2 SetSize=3
```

```
IA2_EVENT_ACTIVE_DESCENDANT_CHANGED on <select> role=ROLE_SYSTEM_LIST FOCUSABLE IA2_STATE_VERTICAL SetSize=3
```

# Chromium: scripting example

```
// test file
```

```
<!--
```

```
@MAC-SCRIPT:p.AXStartTextMarker
```

```
-->
```

```
<p id="p">Paragraph</p>
```

```
// expectation file
```

```
p.AXStartTextMarker={:1, 0, down}
```

# Chromium: test flow example

```
<!--  
@EXECUTE-AND-WAIT-FOR:open_modal()  
@EXECUTE-AND-WAIT-FOR:remove_span()  
-->  
  
<span>some text</span>  
  
<div><dialog><div id="dialog-text" tabindex="0">Some Text</div></dialog></div>  
  
<script>  
  
function open_modal() { document.querySelector("dialog").showModal(); return "dialog"; }  
  
function remove_span() { document.querySelector("span").remove(); document.getElementById("dialog-text").innerText = "Done"; return "Done"; }  
  
</script>
```

Chromium

Platform APIs

Chromium

Rebaselining



Let's pull things together

# Chromium testsuite

- Multiple platforms
- Multiple testables
  - Properties
  - Methods
  - Events
- Multiple applications (including Desktop and Mobile)
- Test flow control
- Rebaseline feature

Ways to improve

JavaScript driven

# Dynamics

- Browsers
  - JS self-driven
  - Webdriver backed
  - Puppeteer
- All apps
  - Platform-dependent user input emulation

# Chromium dependency

- Chromium internals dependency
- Hosting/owning

Thoughts?