**ARIA Technique - Using ARIA role of alert for Error Feedback in Forms (updated draft per comments as of April 11 2012)**

**Applicability**

As of this writing this technique applies to HTML technologies used with scripting.

This technique relates to:

* Success Criterion 3.3.3 Error Suggestion

**User Agent and Assistive Technology Support Notes**

To beprovided**..**

**Description**

This technique uses the ARIA role attribute with the value of “alert” (on an allowed HTML element used in HTML forms) to inform assistive technologies to monitor an applicable node and convey changes, and in order to provide a suggestion to the user for correction on form input errors. The specific suggestion conveyed would be contained in the “content” of the associated alert.

NOTE: The ARIA describedby attribute may be provided as a helpful addition to provide a reference to the error if the user desires the error message to be conveyed again, but is not strictly necessary for successful application of this technique, and is incidental to the application of this technique.

**Example:**

The following example (derived from: <http://html.cita.illinois.edu/nav/form/aria/index.php?example=3>

) shows using ARIA role attribute with value of “alert” attached to an HTML DIV element (and aria-describedby as an additional help) for error feedback on an HTML form. The form has a text field of last name (required). If an error is made (no text entered for last name), role alert is activated. Associated JavaScript processes the alert if activated. First the HTML source code is presented, and then the associated Javascript:

**HTML Source:**

<div>
    <h2>Sample Form</h2>
    <p>All fields are required.</p>
    <form id="SampleForm" action="#" method="post">

      <div class="text">
         <input type="text"
                    name="lname"
                    id="lname"
                    size="20"
                    onblur="checkLast(event)"
                    **aria-describedby="*lname\_error*"**/>
          <div class="error" **id="*lname\_error*" role="alert"**></div>
      </div>

            <div class="button">
        <input type="button"
                    value="Check Form"
                    onclick=" validateForm()"/>
      </div>
    </form>

  </div>

 **JavaScript Source:**

 function validateForm() {
      var errorMessage = "Please complete the following fields:";
      var errorElements = new Array();
      var node;
      var result;
      var node\_focus = null;

 // Check last name
      node = document.getElementById("lname");
      if( node && node.value == "" ) {

          result = "\nYou must enter your last name";
          errorElements.push(result);
          showError(node, result);

          if( node\_focus == null )
             node\_focus = node;
       }

 }

function checkLast(event) {
     var node = getTarget(event);
     var result;

     if( node && node.value == "" ) {
       result =  "You must enter your last name";
      } else {
       result = "";
      } // endif

    showError(node, result);

    return stopPropagation( event);

}

**Resources**

<http://html.cita.illinois.edu/nav/form/aria/>

<http://www.msfw.com/accessibility/>

<http://www.w3.org/TR/2008/REC-WCAG20-20081211/#minimize-error>

<http://www.w3.org/TR/2011/CR-wai-aria-20110118/states_and_properties#aria-describedby>

<http://www.w3.org/TR/2011/CR-wai-aria-20110118/roles#alert>

<http://www.w3.org/TR/2011/WD-html5-20110525/>

**Related Techniques**

<http://www.w3.org/TR/2012/NOTE-WCAG20-TECHS-20120103/G83>

<http://www.w3.org/TR/2012/NOTE-WCAG20-TECHS-20120103/G85>

<http://www.w3.org/TR/2012/NOTE-WCAG20-TECHS-20120103/G84>

(note: not sure of these?)

**Tests**

**Procedure**

1. Check that there is an ARIA role attribute with value of “alert” associated with an allowed HTML element used in an HTML form
2. Check that the script associated with the HTML form includes a correct error message as the “contents” of the above-referenced “alert” value

**Expected Results**

#1 and #2 are both true