# AccName 1.2 Logic Tree Outline

## Key Terms

Explicitly Hidden: Any element that is specifically hidden using the HTML hidden attribute, CSS display:none or visibility:hidden, or if it includes aria-hidden="true".

Implicitly Hidden: Any element contained within a parent element that is explicitly hidden.

Presentational: Any element containing role="presentation" or role="none" that is not focusable and does not include any global ARIA attributes.

Native Markup: Any element that includes a native host-defined mechanism for labelling, such as form fields with label elements, images with alt attributes, and svg elements with embedded desc and title elements.

Name From Content: Any element with an ARIA role mapping that supports 'name from content' as part of its role definition within the ARIA specification.

Exceptions Rule Checklist: A checklist that determines when to process name from contents as part of the recursion algorithm, which is necessary to differentiate names computed on focusable roles, as opposed to parsing name from contents of the same roles when embedded within different roles. To understand the background of this, please read the discussion thread: <https://lists.w3.org/Archives/Public/public-aria/2017Jun/0057.html>

 // Always include name from content when the referenced node matches list1, as well as when child nodes match those within list3

 // Note: gridcell was added to list1 to account for focusable gridcells that match the ARIA 1.0 paradigm for interactive grids.

 // So too was row to match 'name from author' and 'name from content' in accordance with the spec.

var list1 = {

roles: ["button", "checkbox", "link", "option", "radio", "switch", "tab", "treeitem", "menuitem", "menuitemcheckbox", "menuitemradio", "row", "cell", "gridcell", "columnheader", "rowheader", "tooltip", "heading"],

tags: ["a", "button", "summary", "input", "h1", "h2", "h3", "h4", "h5", "h6", "menuitem", "option", "tr", "td", "th"]

};

 // Never include name from content when current node matches list2

 // The rowgroup role was added to prevent 'name from content' in accordance with relevant ARIA 1.1 spec changes.

 // The fieldset element and group role was added to account for implicit mappings where name from content is not supported.

var list2 = {

roles: ["application", "alert", "log", "marquee", "timer", "alertdialog", "dialog", "banner", "complementary", "form", "main", "navigation", "region", "search", "article", "document", "feed", "figure", "img", "math", "toolbar", "menu", "menubar", "grid", "listbox", "radiogroup", "textbox", "searchbox", "spinbutton", "scrollbar", "slider", "tablist", "tabpanel", "tree", "treegrid", "separator", "rowgroup", "group"],

tags: ["article", "aside", "body", "select", "datalist", "optgroup", "dialog", "figure", "footer", "form", "header", "hr", "img", "textarea", "input", "main", "math", "menu", "nav", "section", "thead", "tbody", "tfoot", "fieldset"]

};

 // As an override of list2, conditionally include name from content if current node is focusable, or if the current node matches list3 while the referenced parent node (root node) matches list1.

var list3 = {

roles: ["term", "definition", "directory", "list", "note", "status", "table", "contentinfo"],

tags: ["dl", "ul", "ol", "dd", "details", "output", "table"]

};

 // Subsequent roles added as part of the Role Parity project for ARIA 1.2.

 // Tracks roles that don't specifically belong within the prior process lists, but shouldn't have any impact on name from content processing.

 // More need to be added …

var list4 = {

roles: ["legend"],

tags: ["legend"]

};

## AccName 1.2 Logic Tree

### Step 1: Process Root Node: Initialize P as {name: "", description: ""}

 A. Set the root node and the current node to the same node to begin processing.

 Continue.

 B. Set the values of P.name and P.description to the returned values of step 2 (Process Current Node) for the current node.

 Continue.

 C. Convert all whitespace characters in the values for P.name and P.description to single spaces and flatten each string into single lines.

 Continue.

 D. If (the values for P.name and P.description are not blank, and both contain the same textual content),

 then set P.description to "".

 Continue.

 E. Trim any leading or trailing whitespace characters from the values for P.name and P.description,

 then return the name and description values as computed for P.

 (Stop)

### Step 2: Process Current Node: Initialize Q as {name: "", description: ""}

 A. If (the root node and the current node are not the same node, and the current node is not directly referenced by aria-labelledby or aria-describedby, and the current node is explicitly hidden),

 or if (the root node and the current node are not the same node, and the current node is directly referenced by aria-labelledby or aria-describedby, and the current node is implicitly hidden),

 or if (the current node has already been traversed while computing the name or description for the root node),

 then abort and skip to step 2.J. Otherwise, continue.

 B. If (the current node includes an attached CSS :before pseudo element),

 then set the value of Q.name to the pseudo element content,

 plus a space character if (the pseudo element is styled as a block level element).

 Continue.

 C. Append the values of Q.name and Q.description with the returned values of step 3 (Process Filter Criteria) for the current node.

 Continue.

 D. if (the value of Q.name is blank, and the current node’s role supports name from content, and the current node’s role is allowed in the exception rule checklist),

 then process each successive first level child node using step 2 (Process Current Node) for the current node, and append the returned name string to the value of Q.name.

 Continue.

 E. If (the current node natively supports embedded child nodes, and the current node includes aria-owns),

 then process each successive id reference and perform step 2 (Process Current Node) on each to get its name, separate each id referenced instance of name with a space character, and combine all into one name string in the order they were processed and append this to the value of Q.name with a space character between them.

 Continue.

 F. If (the current node includes an attached CSS :after pseudo element),

 then append the CSS pseudo element content to the value of Q.name,

 preceded with a space character if (the pseudo element is styled as a block level element).

 Continue.

 G. If (the root node and the current node are the same node, and the value of Q.name is empty, and the current node includes a non-empty title),

 then set the value of Q.name to the title.

 Continue.

 H. If (the root node and the current node are the same node, and the value of Q.name is not empty, and the value of Q.description is empty, and the current node includes a non-empty title),

 then set the value of Q.description to the title.

 Continue.

 I. If (the value of Q.name is not empty, and the current node is visually styled as a block level element or separated on a new line),

 or if (the current node is a control or widget with a value),

 then add a space character to the beginning and end of Q.name.

 Continue.

 J. Return the name and description values as computed for Q.

 (Stop)

### Step 3: Process Filter Criteria: Initialize R as {name: "", description: ""}

 A. If (the root node and the current node are the same node, and the current node includes aria-describedby),

 then process each successive id reference and perform step 2 (Process Current Node) on each to get its description, separate each id referenced instance of description with a space character, and combine all into one description string in the order they were processed and set this as the value of R.description.

 Continue.

 B. If (the current node includes aria-labelledby, and the current node is not already part of an aria-labelledby traversal),

 then process each successive id reference and perform step 2 (Process Current Node) on each to get its name, separate each id referenced instance of name with a space character, and combine all into one name string in the order they were processed and set this as the value of R.name.

 If (R.name is not empty), then abort and skip to step 3.G. Otherwise, continue.

 C. If (the current node includes a non-empty value for aria-label, and the current node is not an embedded control or widget),

 then set the value of R.name to the value of aria-label.

 If (R.name is not empty), then abort and skip to step 3.G. Otherwise, continue.

 D. If (the current node is not an embedded control or widget, and the current node's native markup provides a supported host-defined labelling mechanism, and the current node is not presentational),

 then set the value of R.name as the host-defined label content.

 If (R.name is not empty), then abort and skip to step 3.G. Otherwise, continue.

 E. If (the root node and the current node are not the same node, and the current node is an embedded control or widget),

 then set the value of R.name as the current widget value, or to "" if no value is defined.

 If (the current node is an embedded control or widget as specified), then abort and skip to step 3.G. Otherwise, continue.

 F. If (the current node is a standard text node),

 then set the value of R.name to the textual content of the node.

 If (the current node is a standard text node as specified), then abort and skip to step 3.G. Otherwise, continue.

 G. Return the name and description values as computed for R.

 (Stop)