# ARIA-AT Assertions

**Status**: draft, intended as a conversation starter
**Author**: Michael Fairchild

This document aims to create a list of assertions that define what exactly “support” means in the context of ARIA and assistive technologies. This information is required in order to compile and deliver consistent and widely understand support information.

# Table of contents

# Note on intent

The intent of this document is not to force assistive technologies to conform to our interpretation of what “support” means. The intent of the document is to create one possible definition of support in an effort to make the definition clearer to developers that use ARIA in their software, to testers, and to assistive technology developers. We welcome the involvement of assistive technology developers.

# Scoring system

The support for a given ARIA attribute will be a boolean result (pass/fail). Any given ARIA role, state or property can have multiple assertions to determine support. I’llEach criterion follows RFC-2119 keywords.

- If any “MUST” or “MUST NOT” assertions are not met, the attribute is not considered to be supported (fail). However, if the attribute meets some but not all of the “MUST” or “MUST NOT” assertions, support MAY be listed as “partial”.
- “SHOULD” or “SHOULD NOT” assertions will not affect our consideration of whether or not an attribute is supported. However, the result of such assertions SHOULD be listed with the conformance results.
- “MAY” or “MAY NOT” assertions will not affect our consideration of whether or not an attribute is supported. However, the result of such assertions SHOULD be listed with the conformance results.

## Reason for no support

The reason why there is no support MUST not affect the scoring result. For example, the browser or accessibility API that the assistive technology depends on may not provide the required information for the assistive technology to provide support. The reason for no support SHOULD be listed with the conformance results if it can be accurately determined.

## Incorrect support that negatively affects the meaning of content

Sometimes assistive technology support can be so incorrect that it negatively affects the meaning of the content. Conformance results MUST indicate if support is incorrect and negatively affects the meaning of the content.

## The configuration of assistive technology

An ARIA role, state, or property MUST be considered to be supported regardless of whether or not support is hidden behind a configuration setting. If an ARIA role, state, or property is not supported by the default configuration, the relevant configuration options to enable support SHOULD be listed in the conformance results.

## Mode of operation

Many assistive technologies provide many modes of operation. For example, many screen readers provide different modes for browsing documents and interacting with applications or forms. This document does not attempt to provide specific assertions for each possible mode of operation. A criterion MUST be supported by at least one available mode of operation in the assistive technology, unless this document explicitly states otherwise in the assertion itself, OR if the mode of operation does not logically provide support. For example, the application mode that is found in many screen readers would not logically support roles and attributes intended for static objects.

## Types of assistive technology

There are several types of assistive technologies that use ARIA. Two of the most common are screen readers and voice control software. It is important to recognize that these technologies use ARIA in different ways, and thus the support assertion might be different for different types of assistive technologies. For example, screen readers MUST support properties such as “aria-activedescendant”, but voice control software does not. This document contains assertions for both screen readers and voice control software. Whenever support assertions differ, the support assertions will explicitly state which type of technology the assertion is for.

All assertions MUST include one of the following keywords to define scope

- “Assistive Technology” - applies to both screen readers and voice control software.
- “Screen readers” - applies to just screen readers
- “Voice control software” - applies to just voice control software

## Assertions and context

There might be different expectations when a role, state or property is within the context of another role, state, or property. Assertions must be conditional in these cases.

## Different Commands to accomplish the same task

There are many different ways to accomplish the same task in many screen readers. For example, different screen readers will provide many different keyboard commands to access different elements. Assertions SHOULD be written to be agnostic in this regard, but conditional assertions MAY exist. An assertion MUST pass if at least one command passes unless the assertion explicitly says otherwise. Additionally, all commands used to test an assertion SHOULD be listed in the conformance results.

## Required information for testing results

Support for a given ARIA role, state, or property may change depending on several factors, including the mode of operation or keystroke used to access the target element. Because of this, any given test for an assistive technology/browser combination may have many support entries, with each entry being a unique combination of the information below. Only one entry must pass for the ARIA role, state, or property to be considered as having minimal support.

- The ARIA role, state, or property that is being tested MUST be provided
- A CSS selector MUST be provided. If the selector matches multiple elements in the example, each matching element must be tested.
- The assertion tested MUST be provided
- The version of the operating system MUST be provided
- The mode of operation MUST be provided
- The exact keystroke or command used to access the target object MUST be provided
- Whether or not the test resulted in a pass or fail MUST be provided
- Whether or not the meaning of content was negatively affect MUST be provided
- The output of the assistive technology SHOULD be provided
- If support is hidden behind a configuration setting, that setting SHOULD be provided
- The Name of the assistive technology MUST be provided
- The version of the assistive technology MUST be provided
- The name 0f the browser MUST be provided
- The version of the browser MUST be provided

# ARIA 1.1 Assertions

## Roles

See ARIA 1.1 section 5.4 for a list of roles.

### Alert

- Assistive technology: the role MUST be conveyed
- Screen readers: The contents MUST be conveyed via aria-live functionality

### Button

- Assistive technology: the role MUST be conveyed

### Combobox

- Assistive technology: the role MUST be conveyed

### Listbox

- Assistive technology: the role MUST be conveyed

### Textbox

- Assistive technology: the role MUST be conveyed

### Option

- Assistive technology: the role MUST be conveyed

## States and Properties

See ARIA 1.1 section 6.6 for a list of states and properties.
An entry must be created for each possible state and property + value combination.

### aria-activedescendant

- Screen readers: when an element with aria-activedescendant is focused, the assistive technology MUST convey that the element referenced by aria-activedescendant is active.

### aria-autocomplete=”list”

- Screen readers: must convey that the element will display a list containing predictions once text is entered

### aria-controls

- Screen readers: must convey that the element controls the referenced element

### aria-expanded=”true”

- Screen readers: must convey that the element is expanded

### aria-expanded=”false”

- Screen readers: must convey that the element is collapsed

### aria-haspopup=”listbox”

- Screen readers: must convey that the element has a listbox popup.

### aria-labelledby

- Screen readers: must convey the element’s accessible name according to the accname spec

### aria-owns

- Screen readers: must convey relationships with the referenced DOM elements and DOM hierarchy, per the ARIA spec.

### aria-selected=”true”

- Screen readers: must convey that the option is in the selected state