

[MS-ARIA]:

Internet Explorer Accessible Rich Internet Applications (WAI-ARIA) 1.0 Standards Support Document

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology,

there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Preliminary

Revision Summary

Date	Revision History	Revision Class	Comments
9/20/2014	1.0	New	Released new document.
1/22/2015	2.0	Major	Significantly changed the technical content.

Preliminary

Table of Contents

1	Introduction	5
1.1	Glossary	5
1.2	References	5
1.2.1	Normative References	5
1.2.2	Informative References	5
1.3	Microsoft Implementations	5
1.4	Standards Support Requirements	6
1.5	Notation.....	7
2	Standards Support Statements.....	8
2.1	Normative Variations	8
2.1.1	[WAI-ARIA1.0] Section 5.2.4 Inherited States and Properties.....	8
2.1.2	[WAI-ARIA1.0] Section 5.2.7.3, Text Alternative Computation	8
2.1.3	[WAI-ARIA1.0] Section 5.2.8, Presentational Children	8
2.1.4	[WAI-ARIA1.0] Section 5.3.4, Landmark Roles	9
2.1.5	[WAI-ARIA1.0] Section 5.4, Definition of Roles.....	9
2.1.6	[WAI-ARIA1.0] Section 6.5.4, Relationship Attributes.....	10
2.1.7	[WAI-ARIA1.0] Section 6.6, Definitions of States and Properties (all aria-* attributes)	10
2.2	Clarifications	12
2.2.1	[WAI-ARIA1.0] Section 5.4, Definition of Roles.....	12
2.2.2	[WAI-ARIA1.0] Section 6.6, Definitions of States and Properties (all aria-* attributes)	13
2.3	Error Handling	14
2.4	Security	14
3	Change Tracking.....	15
4	Index.....	17

1 Introduction

This document describes the level of support provided by Windows Internet Explorer for the Accessible Rich Internet Applications (WAI-ARIA) 1.0 specification, published 20 March 2014. Internet Explorer provides access to content for assistive technologies through the **Microsoft Active Accessibility (MSAA)** and **Microsoft UI Automation (UIA)** accessibility frameworks.

1.1 Glossary

The following terms are specific to this document:

Microsoft Active Accessibility (MSAA): A Component Object Model (COM)-based technology that improves the way accessibility aids work with applications running on Microsoft Windows. It provides dynamic-link libraries that are incorporated into the operating system as well as a COM interface and API elements that provide reliable methods for exposing information about UI elements.

Microsoft UI Automation (UIA): The accessibility model for Microsoft Windows that programmatically gathers information about an application's User Interface (UI) elements and exposes it to assistive technology products and automated test scripts. UI Automation is the successor to the Microsoft Active Accessibility (MSAA) framework.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](#).

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.rfc-editor.org/rfc/rfc2119.txt>

[WAI-ARIA1.0] Craig, J., and Cooper, M., Eds., "Accessible Rich Internet Applications (WAI-ARIA) 1.0", W3C Recommendation, March 2014, <http://www.w3.org/TR/2014/REC-wai-aria-20140320/>

1.2.2 Informative References

None.

1.3 Microsoft Implementations

The following Microsoft products implement some portion of the [\[WAI-ARIA1.0\]](#) specification:

- Windows Internet Explorer 8
- Windows Internet Explorer 9

- Windows Internet Explorer 10
- Internet Explorer 11
- Internet Explorer 11 for Windows 10

In addition, each version of Internet Explorer implements multiple document modes, which can vary individually in their support of the standard. The following table lists the document modes available in each version of Internet Explorer:

Browser Version	Document Modes Supported
Internet Explorer 8	Quirks Mode IE7 Mode IE8 Mode
Internet Explorer 9	Quirks Mode IE7 Mode IE8 Mode IE9 Mode
Internet Explorer 10	Quirks Mode IE7 Mode IE8 Mode IE9 Mode IE10 Mode
Internet Explorer 11	Quirks Mode IE7 Mode IE8 Mode IE9 Mode IE10 Mode IE11 Mode
Internet Explorer 11 for Windows 10	Quirks Mode IE7 Mode IE8 Mode IE9 Mode IE10 Mode IE11 Mode Edge (beta) Mode

Throughout this document, the document mode appears first followed by the browser version in parentheses. Only those document modes and versions of Internet Explorer for which there is a variation note will be listed. If the document mode is not listed, conformance to the specification can be assumed.

1.4 Standards Support Requirements

To conform to [\[WAI-ARIA1.0\]](#), a user agent must implement all required portions of the specification. Any optional portions that have been implemented must also be implemented as described by the specification. Normative language is usually used to define both required and optional portions. (For more information, see [\[RFC2119\]](#).)

The following table lists the sections of [WAI-ARIA1.0] and whether they are considered normative or informative.

Sections	Normative/Informative
1, 2	Informative
3	Normative
4	Informative
5-9	Normative
10	Informative

1.5 Notation

The following notations are used in this document to differentiate between notes of clarification, variation from the specification, and points of extensibility.

Notation	Explanation
C####	This identifies a clarification of ambiguity in the target specification. This includes imprecise statements, omitted information, discrepancies, and errata. This does not include data formatting clarifications.
V####	This identifies an intended point of variability in the target specification such as the use of MAY, SHOULD, or RECOMMENDED. (See [RFC2119] .) This does not include extensibility points.
E####	Because the use of extensibility points (such as optional implementation-specific data) can impair interoperability, this profile identifies such points in the target specification.

For document mode and browser version notation, see also section [1.3](#).

2 Standards Support Statements

This section contains a full list of variations, clarifications, and extension points in the Microsoft implementation of [\[WAI-ARIA1.0\]](#).

- Section [2.1](#) includes only those variations that violate a MUST requirement in the target specification.
- Section [2.2](#) describes further variations from MAY and SHOULD requirements.
- Section [2.3](#) identifies variations in error handling.
- Section [2.4](#) identifies variations that impact security.

2.1 Normative Variations

The following subsections detail the normative variations from MUST requirements in [\[WAI-ARIA1.0\]](#).

2.1.1 [WAI-ARIA1.0] Section 5.2.4 Inherited States and Properties

V0001:

The specification states:

States and properties are inherited from superclass roles in the role taxonomy, not from ancestor elements in the DOM tree. These properties are not explicitly defined on the role, as the inheritance of properties is automatic.

IE10, IE11, and Edge (beta) Modes (All Versions)

`aria-readonly` is not inherited.

2.1.2 [WAI-ARIA1.0] Section 5.2.7.3, Text Alternative Computation

V0002:

The specification states:

The text equivalent computation outlined below is a description of how user agents acquire a name or description that they then publish through the accessibility API.

IE10, IE11, and Edge (beta) Modes (All Versions)

The text alternative computation algorithm is not supported.

2.1.3 [WAI-ARIA1.0] Section 5.2.8, Presentational Children

V0003:

The specification states:

The DOM descendants are presentational. User agents SHOULD NOT expose descendants of this element through the platform accessibility API. If user agents do not hide the descendant nodes, some information may be read twice.

IE10, IE11, and Edge (beta) Modes (All Versions)

The `img`, `progressbar`, `scrollbar`, `separator`, and `slider` roles expose their child nodes to the accessibility tree, despite being defined in the spec as having presentational children.

2.1.4 [WAI-ARIA1.0] Section 5.3.4, Landmark Roles

V0004:

The specification states:

The following roles are regions of the page intended as navigational landmarks. All of these roles inherit from the landmark base type and, with the exception of `application`, all are imported from the Role Attribute [ROLE]. The roles are included here in order to make them clearly part of the WAI-ARIA Role taxonomy.

- `application`
- `banner`
- `complementary`
- `contentinfo`
- `form`
- `main`
- `navigation`
- `search`

All Modes (All Versions)

Landmark roles are not supported, however the `application` role is supported for IE8, IE9, IE10, IE11, or Edge (beta) Mode (All Versions).

2.1.5 [WAI-ARIA1.0] Section 5.4, Definition of Roles

V0005:

The specification states:

`article`
A section of a page that consists of a composition that forms an independent part of a document, page, or site.
....
`definition`
A definition of a term or concept.
....
`log`
A type of live region where new information is added in meaningful order and old information may disappear. See related `marquee`.
....
`math`
Content that represents a mathematical expression.
....
`note`
A section whose content is parenthetical or ancillary to the main content of the resource.
....
`row`
A row of cells in a grid.
....
`rowgroup`
A group containing one or more row elements in a grid.
....
`scrollbar`

A graphical object that controls the scrolling of content within a viewing area, regardless of whether the content is fully displayed within the viewing area.

....
timer

A type of live region containing a numerical counter which indicates an amount of elapsed time from a start point, or the time remaining until an end point.

IE8 and IE9 Modes (All Versions)

Not supported.

2.1.6 [WAI-ARIA1.0] Section 6.5.4, Relationship Attributes

V0006:

The specification states:

This section lists attributes that indicate relationships or associations between elements which cannot be readily determined from the document structure.

- aria-activedescendant
- aria-controls
- aria-describedby
- aria-flowto
- aria-labelledby
- aria-owns
- aria-posinset
- aria-setsize

IE10, IE11, and Edge (beta) Modes (All Versions)

aria-describedby, aria-labelledby, aria-flowto, and aria-owns are not supported when the referenced object is not accessible.

2.1.7 [WAI-ARIA1.0] Section 6.6, Definitions of States and Properties (all aria-* attributes)

V0007:

The specification states:

aria-atomic

Indicates whether assistive technologies will present all, or only parts of, the changed region based on the change notifications defined by the aria-relevant attribute. See related aria-relevant.

....

aria-autocomplete

Indicates whether user input completion suggestions are provided.

....

aria-dropeffect

Indicates what functions can be performed when the dragged object is released on the drop target. This allows assistive technologies to convey the possible drag options available to users, including whether a pop-up menu of choices is provided by the application. Typically, drop effect functions can only be provided once an object has been grabbed for a drag operation as the drop effect functions available are dependent on the object being dragged.

....

aria-grabbed (state)

Indicates an element's "grabbed" state in a drag-and-drop operation.

....

aria-label
Defines a string value that labels the current element. See related aria-labelledby.
....
aria-multiline
Indicates whether a text box accepts multiple lines of input or only a single line.
....
aria-orientation
Indicates whether the element and orientation is horizontal or vertical.
....
aria-sort
Indicates if items in a table or grid are sorted in ascending or descending order.
....
aria-valuetext
Defines the human readable text alternative of aria-valuenow for a range widget.

IE8 and IE9 Modes (All Versions)

Not supported.

V0008:

The specification states:

aria-level (property)
Defines the hierarchical level of an element within a structure.
....
aria-posinset (property)
Defines an element's number or position in the current set of listitems or treeitems. Not required if all elements in the set are present in the DOM. See related aria-setsize
....
aria-setsize (property)
Defines the number of items in the current set of listitems or treeitems. Not required if all elements in the set are present in the DOM. See related aria-posinset.
....
aria-haspopup (property)
Indicates that the element has a popup context menu or sub-level menu.
....
aria-atomic (property)
Indicates whether assistive technologies will present all, or only parts of, the changed region based on the change notifications defined by the aria-relevant attribute. See related aria-relevant.
....
aria-relevant (property)
Indicates what user agent change notifications (additions, removals, etc.) assistive technologies will receive within a live region. See related aria-atomic.

IE10, IE11, and Edge (beta) Modes (All Versions)

Setting any of the above properties has no effect other than to update the list of current ARIA properties (`IUIAutomationElement::CurrentAriaProperties` property for MSAA and `AriaProperties` property for UIA).

V0009:

The specification states:

Values of aria-checked

True
The element is checked

false

The element supports being checked but is not currently checked.

mixed

Indicates a mixed mode value for a tri-state checkbox or menuitemcheckbox.

undefined (default)

The element does not support being checked

IE10, IE11, and Edge (beta) Modes (All Versions)

The mixed value is not supported.

V0010:

The specification states:

aria-activedescendant (property)

Identifies the currently active descendant of a composite widget.

IE10, IE11, and Edge (beta) Modes (All Versions)

The aria-activedescendant property is only partially supported.

2.2 Clarifications

The following subsections identify clarifications relative to [\[WAI-ARIA1.0\]](#).

2.2.1 [WAI-ARIA1.0] Section 5.4, Definition of Roles

C0001:

The specification states:

presentation (role)

....

For any element with a role of presentation and which is not focusable, the user agent MUST NOT expose the implicit native semantics of the element (the role and its states and properties) to accessibility APIs. However, the user agent MUST expose content and descendant elements that do not have an explicit or inherited role of presentation. Thus, the presentation role causes a given element to be treated as having no role or to be removed from the accessibility tree, but does not cause the content contained within the element to be removed from the accessibility tree.

IE10, IE11, and Edge (beta) Modes (All Versions)

Elements assigned to the presentation role are removed from the accessibility tree; however, they are not placed back into the tree when other accessibility properties (that would normally override the presentation role) are set.

C0002:

The specification states:

Characteristics of menuitemradio

Inherited States and Properties: aria-checked (state) (required)

....

Characteristics of radio

Inherited States and Properties: aria-checked (state) (required)

IE10, IE11, and Edge (beta) Modes (All Versions)

The `aria-checked` state is supported for `menuitemradio` and `radio` roles, but unlike for other roles, it is mapped to the *selected* state value (`STATE_SYSTEM_SELECTED` for MSAA, `IsSelected` for UIA) instead of the *checked* state value (`STATE_SYSTEM_CHECKED` for MSAA, `ToggleState` for UIA).

2.2.2 [WAI-ARIA1.0] Section 6.6, Definitions of States and Properties (all aria-* attributes)

C0003:

The specification states:

```
aria-busy (state)
Indicates whether an element, and its subtree, are currently being updated.
....
aria-haspopup (property)
Indicates that the element has a popup context menu or sub-level menu.
....
aria-sort (property)
Indicates if items in a table or grid are sorted in ascending or descending order.
```

IE10, IE11, and Edge (beta) Modes (All Versions)

MSAA provides corresponding properties for `aria-busy` and `aria-haspopup`, but there are no equivalent UIA properties. Setting them has no effect in UIA other than to update the list of current ARIA properties (`AriaProperties`).

`aria-sort` is only supported in UIA (with the equivalent property of `UIA_ItemStatusPropertyId`), and not in MSAA.

C0004:

The specification states:

```
aria-selected (state)

Indicates the current "selected" state of various widgets. See related aria-checked and aria-pressed.

This attribute is used with single-selection and multiple-selection widgets:

1. Single-selection containers where the currently focused item is not selected. The selection normally follows the focus, and is managed by the user agent.

2. Multiple-selection containers. Authors SHOULD ensure that any selectable descendant of a container in which the aria-multiselectable attribute is true specifies a value of either true or false for the aria-selected attribute.
```

IE10, IE11, and Edge (beta) Modes (All Versions)

The UIA `SelectionPattern` control pattern will function only if a widget has selectable children (as indicated by an `aria-selected` value of `true` or `false`). Thus the UIA `SelectedItemPattern` will not function if `aria-selected` has a value of `undefined` (the default value).

C0005:

The specification states:

Characteristics of aria-selected

Used in Roles:

gridcell
option
row
tab

Inherits into Roles:

columnheader
menuitemradio
radio
rowheader
treeitem

IE10, IE11, and Edge (beta) Modes (All Versions)

The existence of a specified `aria-selected` or `aria-checked` attribute (with `true` or `false` value) is used to determine if an element is selectable or not, rather than the [\[WAI-ARIA1.0\]](#) role inheritance model as specified. Thus in some situations the resulting role behavior may conform to that implied by the specification, and in other situations it may differ.

2.3 Error Handling

There are no additional considerations for error handling.

2.4 Security

There are no additional security considerations.

3 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1.3 Microsoft Implementations	Added IE Edge Mode and Windows 10 to the version support table.	N	Content update.
2.1.1 [WAI-ARIA1.0] Section 5.2.4 Inherited States and Properties	Added IE Edge Mode and Windows 10 version support.	Y	Content update.
2.1.2 [WAI-ARIA1.0] Section 5.2.7.3, Text Alternative Computation	Added IE Edge Mode and Windows 10 version support.	Y	Content update.
2.1.3 [WAI-ARIA1.0] Section 5.2.8, Presentational Children	Added IE Edge and Windows 10 version support.	Y	Content update.
2.1.4 [WAI-ARIA1.0] Section 5.3.4, Landmark Roles	Added IE Edge and Windows 10 version support.	Y	Content update.
2.1.6 [WAI-ARIA1.0] Section 6.5.4, Relationship Attributes	Added IE Edge and Windows 10 version support.	N	Content update.
2.1.7 [WAI-ARIA1.0] Section 6.6, Definitions of States and Properties (all aria-* attributes)	Added IE Edge and Windows 10 version support.	Y	Content update.
2.2.1 [WAI-ARIA1.0] Section 5.4, Definition of Roles	Added IE Edge and Windows 10 version support.	Y	Content update.
2.2.2 [WAI-ARIA1.0] Section 6.6, Definitions of States and Properties (all aria-* attributes)	Added IE Edge and Windows 10 version support.	Y	Content update.

4 Index

C

[Change tracking](#) 15

D

Definition of Roles ([section 2.1.5](#) 9, [section 2.2.1](#) 12)

Definitions of States and Properties (all aria-* attributes) ([section 2.1.7](#) 10, [section 2.2.2](#) 13)

G

[Glossary](#) 5

I

[Informative references](#) 5

[Introduction](#) 5

L

[Landmark Roles](#) 9

N

[Normative references](#) 5

P

[Presentational Children](#) 8

R

References

[informative](#) 5

[normative](#) 5

[Relationship Attributes](#) 10

T

[Text Alternative Computation](#) 8

[Tracking changes](#) 15