# [MS-DOM2TR]:
# Internet Explorer Document Object Model (DOM) Level 2 Traversal and Range 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.
- **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.

## Revision Summary

| Date | Revision History | Revision Class | Comments |
|---|---|---|---|
| 09/08/2010 | 0.1 | New | Released new document. |
| 02/10/2011 | 1.0 | Minor | Clarified the meaning of the technical content. |
| 02/22/2012 | 3.0 | Major | Significantly changed the technical content. |
| 07/25/2012 | 3.1 | Minor | Clarified the meaning of the technical content. |

# Table of Contents

# 1 Introduction

This document describes the level of support provided by Windows® Internet Explorer® 9 and Windows® Internet Explorer® 10 for the *Document Object Model (DOM) Level 2 Traversal and Range Specification* [DOM Level 2 - Traversal Range] Version 1.0, W3C Recommendation 13 November 2000.

The [DOM Level 2 - Traversal Range] specification may contain guidance for authors of webpages and browser users, in addition to user agents (browser applications). Statements found in this document apply only to normative requirements in the specification targeted to user agents, not those targeted to authors.

## 1.1 Glossary

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

## 1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 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. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[DOM Level 2 - Traversal Range] W3C, "Document Object Model (DOM) Level 2 Traversal and Range Specification Version 1.0", W3C Recommendation 13 November, 2000, http://www.w3.org/TR/DOM-Level-2-Traversal-Range/

[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

### 1.2.2 Informative References

None.

## 1.3 Microsoft Implementations

The following Microsoft products implement some portion of the [DOM Level 2 - Traversal Range] specification:

- Windows® Internet Explorer® 9
- Windows® Internet Explorer® 10

In addition, each version of Windows® 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 9 | Quirks mode<br>IE7 mode<br>IE8 mode<br>IE9 mode |
| Internet Explorer 10 | Quirks mode<br>IE7 mode<br>IE8 mode<br>IE9 mode<br>IE10 Mode |

Note that IE9 mode and IE10 Mode are the only document modes that support [DOM Level 2 - Traversal Range].

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.

**Note** "Standards mode" in Windows® Internet Explorer® 7 and "IE7 mode" in Windows® Internet Explorer® 8 refer to the same document mode. "IE7 mode" is the preferred way of referring to this document mode across all versions of the browser.

## 1.4 Standards Support Requirements

To conform to [DOM Level 2 - Traversal Range], 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 [DOM Level 2 - Traversal Range] and whether they are considered normative or informative.

| Sections | Normative/Informative |
|---|---|
| 1 | Normative |
| 2 | Normative |

## 1.5 Notation

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

| Notation | Explanation |
|---|---|
| C#### | 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#### | 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. |

| Notation | Explanation |
|---|---|
| E#### | Identifies extensibility points (such as optional implementation-specific data) in the target specification, which can impair interoperability. |

For document mode and browser version notation, see section 1.3.

# 2 Standards Support Statements

This section contains a full list of variations, clarifications, and extension points in the Microsoft implementation of [DOM Level 2 - Traversal Range].

- 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 [DOM Level 2 - Traversal Range].

### 2.1.1 [DOM Level 2 - Traversal Range] Section 1.2, Formal Interface Definition

V0001:

The specification states:

```
filter of type NodeFilter, readonly
    The NodeFilter used to screen nodes.
```

*IE9 Mode and IE10 Mode (All Versions)*

The **filter** property returns a function pointer to the **NodeFilter** callback.

V0002:

The specification states:

```
acceptNode
Test whether a specified node is visible in the logical view of a TreeWalker or NodeIterator.
This function will be called by the implementation of TreeWalker and NodeIterator; it is not
normally called directly from user code. (Though you could do so if you wanted to use the
same filter to guide your own application logic.)

Parameters
n of type Node
The node to check to see if it passes the filter or not.

Return Value
short  a constant to determine whether the node is accepted, rejected, or skipped, as defined
above.

No Exceptions
```

*IE9 Mode and IE10 Mode (All Versions)*

The **acceptNode** method can return a hexadecimal string of the form "0x1", which indicates FILTER_ACCEPT behavior. However, in this case, the string is not converted into the FILTER_ACCEPT constant value.

V0003:

The specification states:

```
filter of type NodeFilter, readonly
The filter used to screen nodes.
```

*IE9 Mode and IE10 Mode (All Versions)*

The **filter** property returns a function pointer to the **NodeFilter** callback.

V0007:

The specification states:

```
createNodeIterator

Create a new NodeIterator over the subtree rooted at the specified node. Parameters
root of type Node

The node which will be iterated together with its children. The iterator is initially
positioned just before this node. The whatToShow flags and the filter, if any, are not
considered when setting this position. The root must not be null.
whatToShow of type unsigned long

This flag specifies which node types may appear in the logical view of the tree presented by
the iterator. See the description of NodeFilter for the set of possible SHOW_ values.

These flags can be combined using OR.
filter of type NodeFilter

The NodeFilter to be used with this TreeWalker, or null to indicate no filter.
entityReferenceExpansion of type boolean

The value of this flag determines whether entity reference nodes are expanded.

Return Value
NodeIterator  The newly created NodeIterator.
Exceptions
DOMException NOT_SUPPORTED_ERR: Raised if the specified root is null.
```

*IE9 Mode and IE10 Mode (All Versions)*

The **NodeIterator** returned by the **createNodeIterator** method has the **whatToShow** property value set as `-1` instead of the expected value of `0xFFFFFFFF`.

### 2.1.2 [DOM Level 2 - Traversal Range] Section 2.6, Deleting Content with a Range

V0011:

The specification states:

```
Note that if deletion of a Range leaves adjacent Text nodes, they are not
automatically merged, and empty Text nodes are not automatically removed. Two Text
 nodes should be joined only if each is the container of one of the boundary-points
of a Range whose contents are deleted. To merge adjacent Text nodes, or remove
empty text nodes, the normalize() method on the Node interface should be used.
```

*IE9 Mode and IE10 Mode (All Versions)*

If a Range corresponds to a complete text node and **deleteContents()** is called, the empty text node is removed from the tree.

### 2.1.3 [DOM Level 2 - Traversal Range] Section 2.12, Range modification under document mutation

V0004:

The specification states:

```
There are two general principles which apply to Ranges under document mutation: The first is
that all Ranges in a document will remain valid after any mutation operation and the second
is that, as much as possible, all Ranges will select the same portion of the document after
any mutation operation.
```

*IE9 Mode and IE10 Mode (All Versions)*

If the boundary points are positioned on the end points of a text node, and if methods such as **replaceWholeText** or **appendData** are used to modify the text node, the boundary points after the operation are incorrect. This error occurs because Windows® Internet Explorer® cannot distinguish between the first position within a text node and the beginning offset of the text node within its parent.

### 2.1.4 [DOM Level 2 - Traversal Range] Section 2.13, Formal Description of the Range Interface

E0001:

The specification describes the **Range** interface.

*IE10 Mode (All Versions)*

The **createContextualFragment()** method has been added to the **Range** interface. The **createContextualFragment()** method parses a string of HTML into a **DocumentFragment** using the starting node of a DOM Range as the parsing context.

## 2.2 Clarifications

There are no additional clarifications to [DOM Level 2 - Traversal Range].

## 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 the [MS-DOM2TR] protocol document between the February 2012 and July 2012 releases. 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.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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 language and 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 or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

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.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- 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 protocol@microsoft.com.

| Section | Tracking number (if applicable) and description | Major change (Y or N) | Change type |
|---|---|---|---|
| 1 Introduction | Updated document to remove beta tagging. | N | Content updated. |

# 4 Index