[MS-HTML401]:
Internet Explorer HTML 4.01 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
02/24/2010	0.1	New	Released new document.
03/17/2010	0.2	Minor	Clarified the meaning of the technical content.
03/26/2010	1.0	None	Introduced no new technical or language changes.
05/26/2010	1.2	None	Introduced no new technical or language changes.
06/29/2010	1.21	Editorial	Changed language and formatting in the technical content.
09/08/2010	1.3	Major	Significantly changed the technical content.
10/13/2010	1.4	Minor	Clarified the meaning of the technical content.
02/10/2011	2.0	Major	Significantly changed 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 6
1.1 Glossary 6
1.2 References 6
1.2.1 Normative References 6
1.2.2 Informative References 6
1.3 Microsoft Implementations 7
1.4 Standards Support Requirements 7
1.5 Notation 8

2 Standards Support Statements 9
2.1 Normative Variations 9
2.1.1 [HTML] Section 5.2.2, Specifying the character encoding 9
2.1.2 [HTML] Section 6.5, Colors 9
2.1.3 [HTML] Section 6.6, Lengths 10
2.1.4 [HTML] Section 6.9, Character encodings 10
2.1.5 [HTML] Section 7.4.2, The TITLE element 10
2.1.6 [HTML] Section 8.1.2, Inheritance of language codes 10
2.1.7 [HTML] Section 8.1.3, Interpretation of language codes 11
2.1.8 [HTML] Section 8.2.4, Overriding the bidirectional algorithm: the BDO element 11
2.1.9 [HTML] Section 9.2.2, Quotations: The BLOCKQUOTE and Q elements 11
2.1.10 [HTML] Section 9.3.2, Controlling line breaks 12
2.1.11 [HTML] Section 9.4, Marking document changes: The INS and DEL elements 12
2.1.12 [HTML] Section 11.2.1, The TABLE element 12
2.1.13 [HTML] Section 11.2.2, Table Captions: The CAPTION element 13
2.1.14 [HTML] Section 11.2.3, Row groups: the THEAD, TFOOT, and TBODY elements 14
2.1.15 [HTML] Section 11.2.4, Column groups: the COLGROUP and COL elements 14
2.1.16 [HTML] Section 11.2.5, Table rows: The TR element 16
2.1.17 [HTML] Section 11.2.6, Table cells: The TH and TD elements 16
2.1.18 [HTML] Section 11.3.2, Horizontal and vertical alignment 18
2.1.19 [HTML] Section 12.1.3, Specifying anchors and links 19
2.1.20 [HTML] Section 12.2.1, Syntax of anchor names 19
2.1.21 [HTML] Section 12.2.2, Nested links are illegal 20
2.1.22 [HTML] Section 12.2.3, Anchors with the id attribute 20
2.1.23 [HTML] Section 12.4, Path information: the BASE element 21
2.1.24 [HTML] Section 12.4.1, Resolving relative URIs 21
2.1.25 [HTML] Section 13.3, Generic inclusion: the OBJECT element 22
2.1.26 [HTML] Section 13.3.1, Rules for rendering objects 23
2.1.27 [HTML] Section 13.3.2, Object initialization: the PARAM element 24
2.1.28 [HTML] Section 13.3.4, Object declarations and instantiations 25
2.1.29 [HTML] Section 13.4, Including an applet: the APPLET element 25
2.1.30 [HTML] Section 13.5, Notes on embedded documents 26
2.1.31 [HTML] Section 13.6.1, Client-side image maps: the MAP and AREA elements 26
2.1.32 [HTML] Section 13.6.2, Server-side image maps 28
2.1.33 [HTML] Section 13.7.1, Width and height 28
2.1.34 [HTML] Section 14.2.1, Setting the default style sheet language 29
2.1.35 [HTML] Section 14.3.1, Preferred and alternate style sheets 29
2.1.36 [HTML] Section 15.2.2, Font modifier elements: FONT and BASEFONT 29
2.1.37 [HTML] Section 16.5, Inline frames: the IFRAME element 30
2.1.38 [HTML] Section 17.3, The FORM element 30
2.1.39 [HTML] Section 17.4, The INPUT element 31

2.1.40 [HTML] Section 17.5, The BUTTON element 31
2.1.41 [HTML] Section 17.6, The SELECT, OPTGROUP, and OPTION elements 32
2.1.42 [HTML] Section 17.6.1, Pre-selected options 32
2.1.43 [HTML] Section 17.13.4, Form content types 33
2.2 Clarifications 33
2.2.1 [HTML] Section 6.2, SGML basic types 33
2.2.2 [HTML] Section 6.8, Language codes 34
2.2.3 [HTML] Section 6.12, Link types 35
2.2.4 [HTML] Section 6.13, Media descriptors 36
2.2.5 [HTML] Section 6.16, Frame target names 36
2.2.6 [HTML] Section 9.2.2, Quotations: The BLOCKQUOTE and Q elements 37
2.2.7 [HTML] Section 9.3.3, Hyphenation 37
2.2.8 [HTML] Section 9.3.4, Preformatted text: The PRE element 37
2.2.9 [HTML] Section 9.4, Marking document changes: The INS and DEL elements 37
2.2.10 [HTML] Section 11.1, Introduction to tables 38
2.2.11 [HTML] Section 11.2.3, Row groups: the THEAD, TFOOT, and TBODY elements 38
2.2.12 [HTML] Section 11.2.6, Table cells: The TH and TD elements 38
2.2.13 [HTML] Section 11.3.3, Cell margins 40
2.2.14 [HTML] Section 12.1.3, Specifying anchors and links 40
2.2.15 [HTML] Section 12.1.5, Internationalization and links 40
2.2.16 [HTML] Section 12.2.3, Anchors with the id attribute 41
2.2.17 [HTML] Section 12.3, Document relationships: the LINK element 41
2.2.18 [HTML] Section 12.3.2, Links and external style sheets 42
2.2.19 [HTML] Section 13.3, Generic inclusion: the OBJECT element 42
2.2.20 [HTML] Section 13.3.2, Object initialization: the PARAM element 42
2.2.21 [HTML] Section 13.4, Including an applet: the APPLET element 43
2.2.22 [HTML] Section 13.6.1, Client-side image maps: the MAP and AREA elements 43
2.2.23 [HTML] Section 14.2, Adding style to HTML 44
2.2.24 [HTML] Section 14.2.3, Header style information: the STYLE element 44
2.2.25 [HTML] Section 14.3.1, Preferred and alternate style sheets 45
2.2.26 [HTML] Section 14.3.2, Specifying external style sheets 46
2.2.27 [HTML] Section 14.6, Linking to style sheets with HTTP headers 47
2.2.28 [HTML] Section 16.1, Introduction to frames 48
2.2.29 [HTML] Section 16.2.2, The FRAME element 48
2.2.30 [HTML] Section 16.3, Specifying target frame information 49
2.2.31 [HTML] Section 16.3.2, Target semantics 49
2.2.32 [HTML] Section 16.4.1, The NOFRAMES element 50
2.2.33 [HTML] Section 16.5, Inline frames: the IFRAME element 50
2.2.34 [HTML] Section 17.2, Controls 50
2.2.35 [HTML] Section 17.3, The FORM element 51
2.2.36 [HTML] Section 17.4, The INPUT element 52
2.2.37 [HTML] Section 17.4.1, Control types created with INPUT 52
2.2.38 [HTML] Section 17.5, The BUTTON element 53
2.2.39 [HTML] Section 17.6, The SELECT, OPTGROUP, and OPTION elements 53
2.2.40 [HTML] Section 17.7, The TEXTAREA element 53
2.2.41 [HTML] Section 17.8, The ISINDEX element 54
2.2.42 [HTML] Section 17.9.1, The LABEL element 54
2.2.43 [HTML] Section 17.10, Adding structure to forms: the FIELDSET and LEGEND elements 55
2.2.44 [HTML] Section 17.11.1, Tabbing navigation 55
2.2.45 [HTML] Section 17.11.2, Access keys 56
2.2.46 [HTML] Section 17.12.1, Disabled controls 58
2.2.47 [HTML] Section 17.12.2, Read-only controls 58

2.2.48 [HTML] Section 17.13.1, Form submission method 59
2.2.49 [HTML] Section 17.13.2, Successful controls 59
2.2.50 [HTML] Section 17.13.3, Processing form data 60
2.2.51 [HTML] Section 17.13.4, Form content types 60
2.2.52 [HTML] Section 18.2.1, The SCRIPT element 60
2.2.53 [HTML] Section 18.2.2, Specifying the scripting language 61
2.2.54 [HTML] Section 18.2.3, Intrinsic events 62
2.2.55 [HTML] Section 18.3.1, The NOSCRIPT element 63
2.2.56 [HTML] Section 24.3, Character entity references for symbols, mathematical symbols, and Greek letters 63
2.3 Error Handling 63
2.4 Security 63

3 Appendix A: Test Suite Failures 64
3.1 HTML 4.01 Test: Frame target names must start with an alphabetical character (a-z, A-Z) 64
3.2 HTML 4.01 Test: rowspan attribute for TD and TH 65
3.3 HTML 4.01 Test: colspan attribute for TD and TH 66
3.4 HTML 4.01 Test: Specify an image with an OBJECT element 67
3.5 HTML 4.01 Test: AREA elements are ignored when a MAP element contains mixed content 68

4 Change Tracking 70

5 Index 72

1 Introduction

This document describes the level of support provided by Windows® Internet Explorer® for the *HTML 4.01 Specification* [HTML], W3C Recommendation 24 December 1999. Internet Explorer displays webpages written in HTML.

The [HTML] specification contains 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.

[HTML] World Wide Web Consortium, "HTML 4.01 Specification", December 1999, http://www.w3.org/TR/html4/

[ISO-10646] International Organization for Standardization, "Information Technology - Universal Multiple-Octet Coded Character Set (UCS)", ISO/IEC 10646:2003, December 2003, http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=39921&ICS1

[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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt

[W3C-sRGB] Stokes, M., Anderson, M., Chandrasekar, S., and Motta, R., "A Standard Default Color Space for the Internet - sRGB", November 1996, http://www.w3.org/Graphics/Color/sRGB

1.2.2 Informative References

[IANA-CharSets] IANA, "Character Sets", Last Updated 2010-11-04, http://www.iana.org/assignments/character-sets

[MS-CSS21] Microsoft Corporation, "Internet Explorer Cascading Stylesheets (CSS) 2.1 Standards Support Document".

[W3C-HTML4-TS] W3C, "HTML4 Test Suite", 2002, http://www.w3.org/MarkUp/Test/HTML401/current/

1.3 Microsoft Implementations

The following Microsoft products implement some portion of the HTML specification:

- Windows® Internet Explorer® 7
- Windows® Internet Explorer® 8
- 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	Documents Modes Supported
Internet Explorer 7	Quirks Mode Standards Mode
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

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 Internet Explorer 7 and "IE7 mode" in 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 [HTML], 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 [HTML] and whether they are considered normative or informative.

Sections	Normative/Informative
1-3	Informative
4-18	Normative
19-23	Informative
24	Normative
Appendices A-B	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 [HTML].

- 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 [HTML].

2.1.1 [HTML] Section 5.2.2, Specifying the character encoding

V0001:

The specification states:

```
To sum up, conforming user agents must observe the following priorities when
determining a document's character encoding (from highest priority to lowest):

1. An HTTP "charset" parameter in a "Content-Type" field.

2. A META declaration with "http-equiv" set to "Content-Type" and a value set for
 "charset".

3. The charset attribute set on an element that designates an external resource.
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)

The determination of a document's character encoding is not based on the **charset** attribute on the **LINK** or anchor tag. Instead, it is based on the **charset** of the retrieved resource.

2.1.2 [HTML] Section 6.5, Colors

V0002:

The specification states:

```
Attribute value type "color" (%Color;) refers to color definitions as specified in
[W3C-sRGB].
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)

No color conversion is performed before rendering colors to the display.

2.1.3 [HTML] Section 6.6, Lengths

V0003:

The specification states:

```
MultiLength: The value ( %MultiLength; in the DTD) may be a %Length; or a relative
length. A relative length has the form "i*", where "i" is an integer. When
allotting space among elements competing for that space, user agents allot pixel
and percentage lengths first, then divide up remaining available space among
relative lengths.
```

All Document Modes (All Versions)

Relative length values for attribute lengths of type **MultiLength** are not supported.

2.1.4 [HTML] Section 6.9, Character encodings

V0004:

The specification states:

```
Values must be strings (e.g., "euc-jp") from the IANA registry (see [IANA-CharSets]
for a complete list).
```

All Document Modes (All Versions)

When an invalid value is passed into the **charset** attribute, a valid **SCRIPT** element can still be produced.

2.1.5 [HTML] Section 7.4.2, The TITLE element

V0005:

The specification states:

```
User agents must always make the content of the TITLE element available to users
 (including TITLE elements that occur in frames).
```

All Document Modes (All Versions)

The **TITLE** element of **IFrames** is not available to users.

2.1.6 [HTML] Section 8.1.2, Inheritance of language codes

V0006:

The specification states:

```
An element inherits language code information according to the following order of
 precedence (highest to lowest):

The lang attribute set for the element itself.
```

```
The closest parent element that has the lang attribute set (i.e., the lang attribute is
inherited).

The HTTP "Content-Language" header (which may be configured in a server). For
 example:

    Content-Language: en-cockney- User agent default values and user preferences.
```

All Document Modes (All Versions)

Inherited **lang** attribute values are not supported.

2.1.7 [HTML] Section 8.1.3, Interpretation of language codes

V0007:

The specification states:

```
In the context of HTML, a language code should be interpreted by user agents as a
hierarchy of tokens rather than a single token. When a user agent adjusts rendering
according to language information (say, by comparing style sheet language codes and
lang values), it should always favor an exact match, but should also consider
matching primary codes to be sufficient. Thus, if the lang attribute value of "en-
US" is set for the HTML element, a user agent should prefer style information that
matches "en-US" first, then the more general value "en".
```

Quirks Mode and IE7 Mode (All Versions)

The pseudo-class **:lang** is not supported.

IE8 Mode and IE9 Mode (All Versions)

The pseudo-class **:lang** is applied in cascade order. More exact matches are not favored over less exact ones.

2.1.8 [HTML] Section 8.2.4, Overriding the bidirectional algorithm: the BDO element

V0008:

The specification states:

```
bdo element, Start tag: required, End tag: required
```

Quirks Mode, IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)

The end tag for the **BDO** element is optional.

2.1.9 [HTML] Section 9.2.2, Quotations: The BLOCKQUOTE and Q elements

V0009:

The specification states:

```
Visual user agents must ensure that the content of the Q element is rendered with
 delimiting quotation marks.
```

Quirks Mode and IE7 Mode (All Versions)

Content of the **Q** element is not displayed in quotation marks.

2.1.10 [HTML] Section 9.3.2, Controlling line breaks

V0010:

The specification states:

```
With respect to bidirectional formatting, the BR element should behave the same way
the [ISO-10646] LINE SEPARATOR character behaves in the bidirectional algorithm.
```

Quirks Mode and IE7 Mode (All Versions)

The [ISO-10646] model for interwoven bidirectional text is not supported.

2.1.11 [HTML] Section 9.4, Marking document changes: The INS and DEL elements

V0011:

The specification states:

```
The INS and DEL elements must not contain block-level content when these elements
behave as inline elements.
```

All Document Modes (All Versions)

Block-level elements are supported. Instead of being forced into inline mode, block-level elements remain as block-level elements and are permitted to inherit **INS** styling. **INS** is treated as a block-level element.

2.1.12 [HTML] Section 11.2.1, The TABLE element

C0001:

The specification states:

```
If any of the columns are specified in relative or percentage terms (see the
section on calculating the width of columns), authors must also specify the
width of the table itself.
```

All Document Modes (All Versions)

If the width of the table is not specified, the columns are sized according to their contents. The column with the largest ratio of actual content width to percentage width is used to resize other columns according to their specified width percentage values. For more information on table layout algorithms, see [MS-CSS21] section 2.1.86.

For example, the following table uses ratios of actual column content width to percentage width of 10 and 20, respectively.

```
<table>
<tr>
   <td width="30%">12</td>
   <td width="70%">12345678901234</td>
<tr>
   <td width="30%">123</td>
   <td width="70%">1</td>
</table>
```

This renders as

12	**12345678901234**
123	1

(Grid lines are added to the table for clarity.)

The following table uses column ratios of 20 and 17.1, respectively.

```
<table>
<tr>
   <td width="30%">1234</td>
   <td width="70%">123456789012</td>
<tr>
   <td width="30%">123456</td>
   <td width="70%">1</td>
</table>
```

This renders as

1234	**123456789012**
123456	1

(Grid lines are added to the table for clarity.)

2.1.13 [HTML] Section 11.2.2, Table Captions: The CAPTION element

V0012:

The specification states:

```
Start tag: required, End tag: required
```

All Document Modes (All Versions)

The end tag for the **CAPTION** element is optional.

V0013:

The specification states:

```
A TABLE element may only contain one CAPTION element.
```

All Document Modes (All Versions)

A **TABLE** element may contain any number of **CAPTION** elements, including zero.

2.1.14 [HTML] Section 11.2.3, Row groups: the THEAD, TFOOT, and TBODY elements

V0014:

The specification states:

```
When present, each THEAD, TFOOT, and TBODY contains a row group. Each row group
must contain at least one row, defined by the TR element.
```

Quirks Mode, IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)

THEAD, **TFOOT**, and **TBODY** are not required to contain a row defined by the **TR** element.

V0015:

The specification states:

```
The TBODY start tag is always required except when the table contains only one
table body and no table head or foot sections.
```

All Document Modes (All Versions)

The **TBODY** start tag is always optional and will be implied for any **TR** elements that are direct children of a **TABLE** element.

2.1.15 [HTML] Section 11.2.4, Column groups: the COLGROUP and COL elements

V0016:

The specification states:

```
width = multi-length [CN]

This attribute specifies a default width for each column in the current column
group. In addition to the standard pixel, percentage, and relative values, this
attribute allows the special form "0*" (zero asterisk) which means that the width
of the each column in the group should be the minimum width necessary to hold the
column's contents. This implies that a column's entire contents must be known
before its width may be correctly computed. Authors should be aware that specifying
"0*" will prevent visual user agents from rendering a table incrementally.
```

All Document Modes (All Versions)

Relative values (such as "2*") and the special form "0*" are not supported. When a cell spanning multiple columns has content, the amount of content in the multi-column cell can increase the width

of other cells in any of the spanned columns. For more information on table layout algorithms, see [MS-CSS21] section 2.1.86.

V0017:

The specification states:

```
width = multi-length [CN]

This attribute specifies a default width for each column spanned by the current COL
element. It has the same meaning as the width attribute for the COLGROUP element
and overrides it.
```

All Document Modes (All Versions)

When a cell spanning multiple columns has content, the amount of content in the multi-column cell can increase the width of other cells in any of the spanned columns. In addition, relative widths (such as `2*`) and the special case `0*` are not supported. For more information on table layout algorithms, see [MS-CSS21] section 2.1.86.

V0018:

The specification states:

```
Attributes defined elsewhere

* id, class (document-wide identifiers)
* lang (language information), dir (text direction)
* title (element title)
* style (inline style information )
* onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove,
  onmouseout, onkeypress, onkeydown, onkeyup (intrinsic events)
* align, char, charoff, valign (cell alignment)
```

All Document Modes (All Versions)

The attributes **char** and **charoff** are not supported.

All Document Modes (All Versions)

The **bgcolor** attribute is also supported on **COL** elements.

V0019:

The specification states:

```
Fixed

A fixed width specification is given in pixels (e.g., width="30"). A fixed-width
specification enables incremental rendering.
```

All Document Modes (All Versions)

Content in cells that span multiple columns can cause the width of cells in one of those columns to become larger than intended. For more information on table layout algorithms, see [MS-CSS21] section 2.1.86.

V0020:

The specification states:

```
Proportional

Proportional specifications (e.g., width="3*") refer to portions of the horizontal
space required by a table. If the table width is given a fixed value via the width
attribute of the TABLE element, user agents may render the table incrementally even
with proportional columns.
```

All Document Modes (All Versions)

Proportional width specifications are not supported.

2.1.16 [HTML] Section 11.2.5, Table rows: The TR element

V0021:

The specification states:

```
Attributes defined elsewhere

* id, class (document-wide identifiers)
* lang (language information), dir (text direction)
* title (element title)
* style (inline style information )
* onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove,
onmouseout, onkeypress, onkeydown, onkeyup (intrinsic events)
* align, char, charoff, valign (cell alignment)
```

All Document Modes (All Versions)

The attributes **char** and **charoff** are not supported. The attributes **borderColor**, **borderColorDark**, and **borderColorLight** are supported on **TR** elements.

2.1.17 [HTML] Section 11.2.6, Table cells: The TH and TD elements

V0022:

The specification states:

```
rowspan = number [CN]

This attribute specifies the number of rows spanned by the current cell. The
default value of this attribute is one ("1"). The value zero ("0") means that the
cell spans all rows from the current row to the last row of the table section
(THEAD, TBODY, or TFOOT) in which the cell is defined.
```

All Document Modes (All Versions)

If the **rowspan** attribute for a **TD** or **TH** element is specified with a value of `0`, this value is ignored and a default value of `1` is used instead.

This causes the HTML4 Test Suite [W3C-HTML4-TS] "Test 11_2_6-BF-01 Table cells: The TH and TD elements" test case to fail. For more information about the failure, see section 3.2 in Appendix A: Test Case Failures.

Quirks Mode and IE7 Mode (All Versions)

Values for **rowspan** greater than `1000` will be ignored.

IE8 Mode (All Versions)

Values for **rowspan** greater than `1000` will cause the entire table to fail to render.

V0023:

The specification states:

```
colspan = number [CN]

This attribute specifies the number of columns spanned by the current cell. The
default value of this attribute is one ("1"). The value zero ("0") means that the
cell spans all columns from the current column to the last column of the column
group (COLGROUP) in which the cell is defined.
```

All Document Modes (All Versions)

If the **colspan** attribute for a **TD** or **TH** element is specified with a value of `0` (zero), this value is ignored and a default value of `1` (one) is used instead.

This causes the HTML4 Test Suite [W3C-HTML4-TS] "Test 11_2_6-BF-01 Table cells: The TH and TD elements" test case to fail. For more information about the failure, see section 3.3 in Appendix A: Test Case Failures.

Quirks Mode and IE7 Mode (All Versions)

Values for **colspan** greater than `1000` will be ignored.

IE8 Mode (All Versions)

Values for **colspan** greater than `1000` will cause the entire table to fail to render.

V0024:

The specification states:

```
width = length [CN]

Deprecated. This attribute supplies user agents with a recommended cell width.
```

All Document Modes (All Versions)

When a cell spanning multiple columns has content, the amount of content in the multi-column cell can increase the width of other cells in any of the spanned columns. For more information on table layout algorithms, see [MS-CSS21] section 2.1.86.

V0025:

The specification states:

```
height = length [CN]

Deprecated. This attribute supplies user agents with a recommended cell height.
```

All Document Modes (All Versions)

When a cell spanning multiple rows has content, the amount of content in the multi-row cell can increase the height of other cells in any of the spanned rows. For more information on table layout algorithms, see [MS-CSS21] section 2.1.86.

V0026:

The specification states:

```
Attributes defined elsewhere

* id, class (document-wide identifiers)
* lang (language information), dir (text direction)
* title (element title)
* style (inline style information )
* onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove,
 onmouseout, onkeypress, onkeydown, onkeyup (intrinsic events )
* align, char, charoff, valign (cell alignment).
```

All Document Modes (All Versions)

The attributes **char** and **charoff** are not supported. Additional attributes **background**, **borderColor**, **borderColorDark**, and **borderColorLight** are supported.

2.1.18 [HTML] Section 11.3.2, Horizontal and vertical alignment

V0027:

The specification states:

```
align = left|center|right|justify|char [CI]

This attribute specifies the alignment of data and the justification of text in a
cell.
```

All Document Modes (All Versions)

The values `justify` and `char` are not supported in **TABLE** elements. To justify the content of a cell, use a child element that supports `justify` as a value.

V0028:

The specification states:

```
char = character [CN]
```

```
This attribute specifies a single character within a text fragment to act as an
axis for alignment.
```

All Document Modes (All Versions)

The **char** attribute is not supported.

V0029:

The specification states:

```
charoff = length [CN]

When present, this attribute specifies the offset to the first occurrence of the
alignment character on each line.
```

All Document Modes (All Versions)

The **charoff** attribute is not supported.

2.1.19 [HTML] Section 12.1.3, Specifying anchors and links

V0030:

The specification states:

```
The LINK element may only appear in the head of a document.
```

All Document Modes (All Versions)

LINK elements that are defined in the **BODY** of the document are also supported.

2.1.20 [HTML] Section 12.2.1, Syntax of anchor names

V0031:

The specification states:

```
Comparisons between fragment identifiers and anchor names must be done by exact
(case-sensitive) match.
```

All Document Modes (All Versions)

Case-sensitive matching is not required for comparisons between fragment identifiers and anchor names.

V0032:

The specification states:

```
Anchor names should be restricted to ASCII characters.
```

All Document Modes (All Versions)

Non-ASCII characters are allowed in anchor names.

2.1.21 [HTML] Section 12.2.2, Nested links are illegal

V0033:

The specification states:

```
Since the DTD defines the LINK element to be empty, LINK elements may not be nested
either.
```

All Document Modes (All Versions)

Internet Explorer recognizes nested **LINK** elements and closes the outer **LINK** element before processing the inner **LINK** element.

2.1.22 [HTML] Section 12.2.3, Anchors with the id attribute

V0034:

The specification states:

```
The id attribute may be used to create an anchor at the start tag of any element
(including the A element).
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)

Navigation to the following elements is not supported in the specified document mode:

Element	Quirks Mode	IE7 Mode	IE8 Mode
APPLET	X	X	
BASE	X	X	X
BODY	X		
COL			X
COLGROUP	X	X	X
COMMENT	X	X	X
FRAME			X
FRAMESET			X
HEAD	X	X	
HTML	X	X	
LINK	X	X	X
META	X	X	X

Element	Quirks Mode	IE7 Mode	IE8 Mode
NEXTID	X	X	X
OBJECT	X	X	
OPTGROUP	X	X	X
PARAM	X	X	X
RP			X
SCRIPT			X
TBODY			X
TFOOT	X	X	X
THEAD			X
TITLE	X	X	X
TR			X

2.1.23 [HTML] Section 12.4, Path information: the BASE element

V0035:

The specification states:

```
The base URI is given by meta data discovered during a protocol interaction, such
as an HTTP header (see [RFC2616]).
```

All Document Modes (All Versions)

The use of the HTTP entity-header fields **Content-Location** and **Content-Base** is not supported.

2.1.24 [HTML] Section 12.4.1, Resolving relative URIs

V0036:

The specification states:

```
Additionally, the OBJECT and APPLET elements define attributes that take precedence
over the value set by the BASE element. Please consult the definitions of these
elements for more information about URI issues specific to them.
```

All Document Modes (All Versions)

The **codebase** attribute of **OBJECT** or **APPLET** elements is not combined with any other attributes of **OBJECT** or **APPLET** elements in order to generate a new **BASE URI**. The **BASE** element is applied in order to resolve relative URLs.

V0037:

The specification states:

```
Note. For versions of HTTP that define a Link header, user agents should handle
these headers exactly as LINK elements in the document. HTTP 1.1 as defined by
[RFC2616] does not include a Link header field (refer to section 19.6.3).
```

All Document Modes (All Versions)

The Link HTTP header is not supported.

2.1.25 [HTML] Section 13.3, Generic inclusion: the OBJECT element

V0038:

The specification states:

```
codebase=uri[CT]

This attribute specifies the base path used to resolve relative URIs specified by
the classid, data, and archive attributes. When absent, its default value is the
base URI of the current document.
```

All Document Modes (All Versions)

The **codebase** attribute is not used to overwrite the root used in relative path navigations.

V0039:

The specification states:

```
data=uri[CT]

This attribute may be used to specify the location of the object's data, for
instance image data for objects defining images, or more generally, a serialized
form of an object which can be used to recreate it. If given as a relative URI, it
should be interpreted relative to the codebase attribute.
```

All Document Modes (All Versions)

The **codebase** attribute is not used to establish a new relative path. All paths in the **data** attribute start at the relative root for relative URIs.

Quirks Mode and IE7 Mode (All Versions)

If the **data** attribute is used to specify image data, the image is displayed in a frame.

V0040:

The specification states:

```
Type = content-type[CI]

If the value of this attribute differs from the HTTP Content-Type returned by the
server when the object is retrieved, the HTTP Content-Type takes precedence.
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)

When the value of the **type** attribute differs from the value of the HTTP **Content-Type** returned by the server, the HTTP **Content-Type** does not take precedence.

V0041:

The specification states:

```
Archives specified as relative URIs should be interpreted relative to the codebase
attribute.
```

All Document Modes (All Versions)

The relative root is not overwritten with the value in the **codebase** attribute.

V0042:

The specification states:

```
declare

When present, this boolean attribute makes the current OBJECT definition a
declaration only. The object must be instantiated by a subsequent OBJECT definition
referring to this declaration.
```

All Document Modes (All Versions)

The **declare** attribute is not supported in the **OBJECT** element. The object is loaded as if the **declare** attribute was not set.

V0043:

The specification states:

```
standby=text[CS]

This attribute specifies a message that a user agent may render while loading the
object's implementation and data.
```

All Document Modes (All Versions)

No **standby** message is rendered while loading an object’s implementation or data.

2.1.26 [HTML] Section 13.3.1, Rules for rendering objects

V0044:

The specification states:

```
If the user agent is not able to render the object for whatever reason (configured
not to, lack of resources, wrong architecture, etc.), it must try to render its
contents.
```

Quirks Mode and IE7 Mode (All Versions)

The content of the **OBJECT** element renders in a new frame as the frame attempts to navigate to the resource. If the resource fails to resolve, no further fallback will occur.

V0045:

The specification states:

```
If a user agent cannot render the outermost OBJECT, it tries to render the
contents, which may be another OBJECT element, etc.
```

Quirks Mode and IE7 Mode (All Versions)

The content of the **OBJECT** element renders in a new frame as the frame attempts to navigate to the resource. If the resource fails to resolve, no further fallback will occur.

V0046:

The specification states:

```
Inline vs. external data. Data to be rendered may be supplied in two ways: inline
and from an external resource.
```

Quirks Mode and IE7 Mode (All Versions)

DataURI-based objects are not rendered through the **OBJECT** tag.

2.1.27 [HTML] Section 13.3.2, Object initialization: the PARAM element

V0047:

The specification states:

```
valuetype = data|ref|object [CI]

This attribute specifies the type of the value attribute. Possible values:
data: This is default value for the attribute. It means that the value specified by
value will be evaluated and passed to the object's implementation as a string.
```

All Document Modes (All Versions)

The value of the **valuetype** attribute is not explicitly set to `data`. However, **PARAM** elements which do not set this value will have a **valuetype** attribute default value of `data`.

V0048:

The specification states:

```
Any number of PARAM elements may appear in the content of an OBJECT or APPLET
element, in any order, but must be placed at the start of the content of the
enclosing OBJECT or APPLET element.
```

All Document Modes (All Versions)

All **PARAM** elements are passed to the associated **OBJECT** element regardless of order or placement.

2.1.28 [HTML] Section 13.3.4, Object declarations and instantiations

V0049:

The specification states:

```
It is possible to specify objects as run-time data for other objects.
```

All Document Modes (All Versions)

When the **valuetype** attribute is specified in the **PARAM** tag, object information is not passed into other objects.

2.1.29 [HTML] Section 13.4, Including an applet: the APPLET element

V0050:

The specification states:

```
codebase=uri [CT]

If this attribute is not specified, then it defaults the same base URI as for the
current document.
```

All Document Modes (All Versions)

The relative base location is not overridden with the value specified in the **codebase** attribute.

V0051:

The specification states:

```
Values for this attribute may only refer to subdirectories of the directory
containing the current document.
```

All Document Modes (All Versions)

The relative base location is not overridden with the value specified in the **codebase** attribute.

V0052:

The specification states:

```
An applet should be stopped before it is serialized.
```

All Document Modes (All Versions)

Loading data through the **object** attribute of the **APPLET** element is not supported.

2.1.30 [HTML] Section 13.5, Notes on embedded documents

V0053:

The specification states:

```
Sometimes, rather than linking to a document, an author may want to embed it
directly into a primary HTML document. Authors may use either the IFRAME element or
the OBJECT element for this purpose, but the elements differ in some ways.
```

Quirks Mode and IE7 Mode (All Versions)

OBJECT elements without a set width and height are recognized as a zero-sized element. As a result, the contents of the **OBJECT** element will not be rendered.

V0054:

The specification states:

```
User agents may render selected frames elements in ways that distinguish them from
unselected frames (e.g., by drawing a border around the selected frame).
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)

OBJECT elements without a set width and height are recognized as a zero-sized element. As a result, the contents of the **OBJECT** element will not be rendered.

IE8 Mode and IE9 Mode (All Versions)

When a frame is defined by an **OBJECT** element loading an HTML document, or when a document has a **content-type** of `text/html`, Internet Explorer does not distinguish selected frames from unselected frames.

2.1.31 [HTML] Section 13.6.1, Client-side image maps: the MAP and AREA elements

V0055:

The specification states:

```
Block-level content. This content should include A elements that specify the
geometric regions of the image map and the link associated with each region.
```

All Document Modes (All Versions)

A elements cannot be used to specify geometries; they can only be used to provide block-level content within a **MAP** element.

V0056:

The specification states:

```
Note that the user agent should render block-level content of a MAP element.
```

All Document Modes (All Versions)

A elements cannot be used to specify geometries; they can only be used to provide block-level content within a **MAP** element.

V0057:

The specification states:

```
When a MAP element contains mixed content (both AREA elements and block-level
content), user agents must ignore the AREA elements.
```

All Document Modes (All Versions)

MAP and **AREA** elements are rendered even if they contain mixed content. **AREA** and **MAP** elements may coexist because the use of the **A** element as a geometric region specifier for a **MAP** element is not supported.

This causes the HTML4 Test Suite [W3C-HTML4-TS] "Test 13_6_1-BF-02 Client-side image maps: the MAP and AREA elements" test case to fail. For more information about the failure, see section 3.1 in Appendix A: Test Case Failures.

V0058:

The specification states:

```
If the user agent doesn't support the PNG format, it tries to render the GIF image.
 If it doesn't support GIF (e.g., it's a speech-based user agent), it defaults to
the text description provided as the content of the inner OBJECT element.
```

Quirks Mode and IE7 Mode (All Versions)

The content of the **OBJECT** element renders in a new frame as the frame attempts to navigate to the resource. If the resource fails to resolve, no further fallback will occur.

V0059:

The specification states:

```
The following example illustrates how anchors may be specified to create inactive
zones within an image map. The first anchor specifies a small circular region with
no associated link. The second anchor specifies a larger circular region with the
same center coordinates. Combined, the two form a ring whose center is inactive and
whose rim is active. The order of the anchor definitions is important, since the
smaller circle must override the larger circle.

<MAP name="map1">
<P>
<A shape="circle" coords="100,200,50">I'm inactive.</A>
<A href="outer-ring-link.html" shape="circle" coords="100,200,250">I'm active.</A>
</MAP>
```

All Document Modes (All Versions)

A elements cannot be used to specify geometries; they can only be used to provide block-level content within a **MAP** element.

2.1.32 [HTML] Section 13.6.2, Server-side image maps

V0060:

The specification states:

```
It is only possible to define a server-side image map for the IMG and INPUT
elements.
```

All Document Modes (All Versions)

INPUT elements do not support client-side or server-side image maps. When **usemap** is used, or an **A** tag is used as a reference to a server-side map, the transactions are treated as a form submission to the current page.

2.1.33 [HTML] Section 13.7.1, Width and height

V0061:

The specification states:

```
When specified, the width and height attributes tell user agents to override the
natural image or object size in favor of these values.
```

Quirks Mode and IE7 Mode (All Versions)

The **height** and **width** attributes are not supported for image files loaded by an **OBJECT** tag.

V0062:

The specification states:

```
When the object is an image, it is scaled.
```

Quirks Mode and IE7 Mode (All Versions)

The **height** and **width** attributes are not supported for image files loaded by an **OBJECT** tag.

V0063:

The specification states:

```
User agents should do their best to scale an object or image to match the width and
height specified by the author.
```

Quirks Mode and IE7 Mode (All Versions)

The **height** and **width** attributes are not supported for image files loaded by an **OBJECT** tag.

2.1.34 [HTML] Section 14.2.1, Setting the default style sheet language

V0064:

The specification states:

```
User agents should determine the default style sheet language for a document
according to the following steps (highest to lowest priority):If any META
declarations specify the "Content-Style-Type", the last one in the character stream
determines the default style sheet language. Otherwise, if any HTTP headers specify
the "Content-Style-Type", the last one in the character stream determines the
default style sheet language. Otherwise, the default style sheet language is
"text/css".
```

All Document Modes (All Versions)

META declarations and HTTP headers that specify "Content-Style-Type" are not supported.

2.1.35 [HTML] Section 14.3.1, Preferred and alternate style sheets

V0065:

The specification states:

```
HTML allows authors to associate any number of external style sheets with a
document.
```

All Document Modes (Internet Explorer 7)

Alternate stylesheets are not supported.

All Document Modes (Internet Explorer 8)

The maximum number of alternate style sheets is 31.

2.1.36 [HTML] Section 15.2.2, Font modifier elements: FONT and BASEFONT

V0190:

The specification states:

```
The BASEFONT element sets the base font size (using the size attribute).
```

IE7 Mode, IE8 Mode, and IE9 Mode (All Versions)

The **BASEFONT** element is defined to be empty in the DTD http://www.w3.org/TR/html401/sgml/loosedtd.html#basefont . In IE9 Mode, the element is not treated as empty.

BASEFONT elements are treated in a similar way to **b** and **i** elements when parsing. Additional **BASEFONT** elements might be added to the tree as a consequence of this parsing behavior.

V0066:

The specification states:

```
The base font size does not apply to headings, except where these are
modified using the FONT element with a relative font size change.
```

Quirks Mode and IE7 Mode (All Versions)

The base font size does not apply to headings, even when the base font size is modified using the relative font size change on the **FONT** element.

IE8 Mode and IE9 Mode (All Versions)

Both the base font size and the relative font size change on the **FONT** element apply to headings.

2.1.37 [HTML] Section 16.5, Inline frames: the IFRAME element

V0067:

The specification states:

```
name = cdata [CI]

This attribute assigns a name to the current frame. This name may be used as the
target of subsequent links.
```

All Document Modes (All Versions)

The value of the **name** attribute is treated as if it were case-sensitive. If an incorrect case is used, then navigation does not proceed correctly.

2.1.38 [HTML] Section 17.3, The FORM element

V0068:

The specification states:

```
accept-charset = charset list [CI]

The client must interpret this list as an exclusive-or list, i.e., the server is
able to accept any single character encoding per entity received.
```

All Document Modes (All Versions)

There is no behavior change based on the contents of the **accept-charset** attribute.

V0069:

The specification states:

```
The default value for this attribute is the reserved string UNKNOWN.
```

All Document Modes (All Versions)

The default value of the **accept-charset** attribute is an empty string.

2.1.39 [HTML] Section 17.4, The INPUT element

V0070:

The specification states:

```
size = cdata [CN]

This attribute tells the user agent the initial width of the control. The width is
given in pixels except when type attribute has the value "text" or "password". In
that case, its value refers to the (integer) number of characters.
```

All Document Modes (All Versions)

The **size** attribute has an impact only on **INPUT** elements with a **type** of `text`, `password`, and `file`. For `text`, `password`, and `file` the width of the input text box is set to the (integer) number of characters.

The size attribute has no impact on INPUT elements with a **type** equal to `hidden`, `checkbox`, `radio`, `reset` or `submit`.

V0071:

The specification states:

```
Attributes defined elsewhere

* id, class (document-wide identifiers)
* lang (language information), dir (text direction)
* title (element title)
* style (inline style information)
* alt (alternate text)
* align (alignment)
* accept (legal content types for a server)
* readonly (read-only input controls)
* disabled (disabled input controls)
* tabindex (tabbing navigation)
* accesskey (access keys)
* usemap (client-side image maps)
* ismap (server-side image maps)
* onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup,
 onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup (intrinsic
events)
```

All Document Modes (All Versions)

The **ismap** attribute is not supported for the **INPUT** element.

2.1.40 [HTML] Section 17.5, The BUTTON element

V0072:

The specification states:

```
value = cdata [CS]
```

```
This attribute assigns the initial value to the button.
```

All Document Modes (All Versions)

Although **value** is set appropriately as the initial value, the button values are not subsequently passed when the form is submitted.

Quirks Mode and IE7 Mode (All Versions)

The button's content is passed in the form that is submitted.

V0073:

The specification states:

```
type = submit|button|reset [CI]

This attribute declares the type of the button. Possible values:

    submit: Creates a submit button. This is the default value.
    reset: Creates a reset button.
    button: Creates a push button.
```

Quirks Mode and IE7 Mode (All Versions)

The button **type** defaults to `button`.

2.1.41 [HTML] Section 17.6, The SELECT, OPTGROUP, and OPTION elements

C0098:

The specification states:

```
Attributes defined elsewhereid, class (document-wide identifiers) lang (language
information), dir (text direction) title (element title) style (inline style information)
disabled (disabled input controls) tabindex (tabbing navigation) onclick, ondblclick,
onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
(intrinsic events)
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)

The **click** event is not supported on **option** elements.

All Document Modes (All Versions)

The **onmouseover** and **onmouseout** events are not supported on option elements.

2.1.42 [HTML] Section 17.6.1, Pre-selected options

V0074:

The specification states:

```
value = cdata [CS]
```

```
If this attribute is not set, the initial value is set to the contents of the
OPTION element.
```

Quirks Mode and IE7 Mode (All Versions)

If the **value** attribute is missing, it is not set to the contents of the **OPTION** element.

2.1.43 [HTML] Section 17.13.4, Form content types

V0075:

The specification states:

```
As with all multipart MIME types, each part has an optional "Content-Type" header
that defaults to "text/plain". User agents should supply the "Content-Type" header,
accompanied by a "charset" parameter.
```

All Document Modes (All Versions)

The **Content-Type** header is omitted except when an **INPUT** element has **type** = `file`. For **INPUT** elements of **type** = `file`, the **Content-Type** is provided, but **charset** is omitted and a default of **UTF-8** encoding is used.

2.2 Clarifications

The following subsections identify clarifications to recommendations made by [HTML].

2.2.1 [HTML] Section 6.2, SGML basic types

V0076:

The specification states:

```
CDATA is a sequence of characters from the document character set and may include
character entities. User agents should interpret attribute values as follows:

- Replace character entities with characters,
- Ignore line feeds,
- Replace each carriage return or tab with a single space.
```

All Document Modes (Internet Explorer 7 and Internet Explorer 8)

Tabs and carriage returns are not replaced with a single space in **CDATA** content. Tabs are removed but carriage returns are not removed.

All Document Modes (Internet Explorer 9)

Tabs and carriage returns are not removed in **CDATA** content.

IE10 Mode (All Versions)

CRLF characters are replaced with an LF character.

C0002:

The specification states:

```
The first occurrence of the character sequence "</" (end-tag open delimiter) is
treated as terminating the end of the element's content. In valid documents, this
would be the end tag for the element.
```

All Document Modes (All Versions)

The end-tag open delimiter (</) sequence is sufficient to terminate the element's content only if it is part of the end tag of the element. If only the character sequence is given (without the rest of the closing tag), subsequent content will be consumed until the next tag close (>) delimiter. It is not the character sequence that terminates the content of an element, but the entire end tag.

C0003:

The specification states:

```
ID and NAME tokens must begin with a letter ([A-Za-z]) and may be followed by any
number of letters, digits ([0-9]), hyphens ("-"), underscores ("_"), colons (":"),
and periods (".").
```

All Document Modes (All Versions)

The following clarifications apply:

- **ID** and **NAME** tokens do not need to start with A-Za-z. These tokens can start with digits, hyphens, underscores, colons and periods. Additionally, the **NAME** token can start with or contain non-ASCII characters.
- The **ID** token can start with or contain a non-ASCII character.

C0004:

The specification states:

```
NUMBER tokens must contain at least one digit ([0-9])
```

All Document Modes (All Versions)

Requirements for the **NUMBER** token exceed those in the specification. Not only must a **NUMBER** token contain at least one digit, it must also represent a valid number and cannot contain letters of the alphabet. **NUMBER** tokens containing non-numeric characters are not valid **NUMBER** tokens because they are not valid numbers. As a result, the values are ignored and replaced with a default (such as 2147483647 for input **maxlength**).

2.2.2 [HTML] Section 6.8, Language codes

C0005:

The specification states:

```
White space is not allowed within the language-code.
```

All Document Modes (All Versions)

White space is removed when it is the leading character in the **hreflang** attribute. White space located elsewhere in the attribute string, or that trails the string, is not removed. When **hreflang** is accessed from the DOM, trailing white spaces and embedded white spaces are kept.

White space in the **lang** attribute is handled in the same way.

2.2.3 [HTML] Section 6.12, Link types

C0006:

The specification states:

```
White space characters are not permitted within link types.
```

All Document Modes (All Versions)

White space characters are preserved in %LinkTypes (such as **rel** and **rev**), within anchors, and within **LINK** elements.

C0007:

The specification states:

```
Link types

Next
Refers to the next document in a linear sequence of documents. User agents may
choose to preload the "next" document, to reduce the perceived load time.
```

All Document Modes (All Versions)

A link type of **Next** is not interpreted as a reference to the next document in a linear sequence of documents. The preloading of a "next" document is not supported.

C0008:

The specification states:

```
Link types
Prev
Refers to the previous document in an ordered series of documents. Some user agents
also support the synonym "Previous".
```

All Document Modes (All Versions)

A link type of **Prev** (or the synonym "Previous") is not interpreted as a reference to the previous document in a linear sequence of documents. The pre-loading of a "prev" document is not supported.

C0009:

The specification states:

```
Link types

Contents
Refers to a document serving as a table of contents. Some user agents also support
the synonym ToC (from "Table of Contents").
```

All Document Modes (All Versions)

A link type of **Contents** (or the synonym "ToC") is not interpreted as a reference to a document that serves as the table of contents for a linear sequence of documents.

2.2.4 [HTML] Section 6.13, Media descriptors

V0077:

The specification states:

```
To facilitate the introduction of these extensions, conforming user agents must be
able to parse the media attribute value as follows:

1. The value is a comma-separated list of entries. For example, media="screen, 3d-
glasses, print and resolution > 90dpi" is mapped to:"screen""3d-glasses""print and
resolution > 90dpi"

2. Each entry is truncated just before the first character that isn't a US ASCII
letter [a-zA-Z] (ISO 10646 hex 41-5a, 61-7a), digit [0-9] (hex 30-39), or hyphen
(hex 2d). In the example, this gives: "screen""3d-glasses""print"

3. A case-sensitive match is then made with the set of media types defined above.
 User agents may ignore entries that don't match. In the example we are left with
screen and print.
```

All Document Modes (All Versions)

Media is not truncated to the character preceding the first non-ASCII character in the sequence, and a case-sensitive match is not performed.

2.2.5 [HTML] Section 6.16, Frame target names

V0078:

The specification states:

```
frame target names (%FrameTarget; in the DTD) must begin with an alphabetic
character.  User agents should ignore all other target names.
```

All Document Modes (All Versions)

The value of the **target** attribute for a **FRAME** element does not have to begin with an alphabetic character [a-zA-Z]. A value is not ignored if it begins with a non-alphabetic character.

This causes the HTML4 Test Suite [W3C-HTML4-TS] "Test 6_16-BF-01 Frame target names" test case to fail. For more information about the failure, see section 3.1 in Appendix A: Test Case Failures.

If the name of the target frame begins with an alphanumeric [a-zA-Z0-9] character, the specified content is loaded into the target frame. If it begins with a non-alphanumeric character, a new browser window is opened.

2.2.6 [HTML] Section 9.2.2, Quotations: The BLOCKQUOTE and Q elements

V0079:

The specification states:

```
User agents should render quotation marks in a language-sensitive manner (see the
lang attribute). Many languages adopt different quotation styles for outer and
inner (nested) quotations, which should be respected by user-agents.
```

All Document Modes (All Versions)

Quotation marks are not rendered in a language sensitive manner.

2.2.7 [HTML] Section 9.3.3, Hyphenation

V0080:

The specification states:

```
Those browsers that interpret soft hyphens must observe the following semantics: If
a line is broken at a soft hyphen, a hyphen character must be displayed at the end
of the first line. If a line is not broken at a soft hyphen, the user agent must
not display a hyphen character. For operations such as searching and sorting, the
soft hyphen should always be ignored.
```

All Document Modes (All Versions)

The soft hyphen is not ignored when searching or sorting.

2.2.8 [HTML] Section 9.3.4, Preformatted text: The PRE element

V0081:

The specification states:

```
width = number [CN]

Deprecated. This attribute provides a hint to visual user agents about the desired
width of the formatted block. The user agent can use this information to select an
appropriate font size or to indent the content appropriately.
```

All Document Modes (All Versions)

The **width** attribute of the **PRE** element is not supported.

2.2.9 [HTML] Section 9.4, Marking document changes: The INS and DEL elements

C0010:

The specification states:

```
User agents should render inserted and deleted text in ways that make the change
obvious. For instance, inserted text may appear in a special font, deleted text may
not be shown at all or be shown as struck-through or with special markings, etc.
```

All Document Modes (All Versions)

Inserted text, or text within an **INS** element, is displayed as underlined text. Deleted text, or text within a **DEL** element, is displayed as text with strikethrough applied.

2.2.10 [HTML] Section 11.1, Introduction to tables

V0082:

The specification states:

```
User agents may exploit the head/body/foot division to support scrolling of body
sections independently of the head and foot sections.
```

All Document Modes (All Versions)

Scrolling body sections independently of the head and foot sections is not supported.

2.2.11 [HTML] Section 11.2.3, Row groups: the THEAD, TFOOT, and TBODY elements

C0011:

The specification states:

```
When long tables are printed, the table head and foot information may be repeated
on each page that contains table data.
```

Quirks Mode, IE7 Mode, and IE9 Mode (All Versions)

The table head and foot information is not repeated on each page; it is only displayed once.

IE8 Mode (All Versions)

The table head and foot information is repeated on each page.

2.2.12 [HTML] Section 11.2.6, Table cells: The TH and TD elements

C0012:

The specification states:

```
axis = cdata [CI]

This attribute may be used to place a cell into conceptual categories that can be
considered to form axes in an n-dimensional space. User agents may give users
access to these categories (e.g., the user may query the user agent for all cells
that belong to certain categories, the user agent may present a table in the form
```

```
of a table of contents, etc.). Please consult the section on categorizing cells for
more information. The value of this attribute is a comma-separated list of category
names.
```

All Document Modes (All Versions)

The **axis** attribute does not affect the presentation or categorization of a **TD** or **TH** element; the attribute is ignored.

V0083:

The specification states:

```
abbr = text [CS]

This attribute should be used to provide an abbreviated form of the cell's content,
and may be rendered by user agents when appropriate in place of the cell's content.
Abbreviated names should be short since user agents may render them repeatedly. For
instance, speech synthesizers may render the abbreviated headers relating to a
particular cell before rendering that cell's content.
```

All Document Modes (All Versions)

The value of **abbr** is not rendered in place of the content of the cell.

V0084:

The specification states:

```
Table cells may contain two types of information: header information and data. This
distinction enables user agents to render header and data cells distinctly, even in
the absence of style sheets. For example, visual user agents may present header
cell text with a bold font. Speech synthesizers may render header information with
a distinct voice inflection.
```

All Document Modes (All Versions)

Header cell text is rendered in a bold font and centered by default. Data cell text is rendered in a normal font and aligned left by default.

V0085:

The specification states:

```
User agents must render either the contents of the cell or the value of the abbr
attribute. For visual media, the latter may be appropriate when there is
insufficient space to render the full contents of the cell. For non-visual media
abbr may be used as an abbreviation for table headers when these are rendered along
with the contents of the cells to which they apply.
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)

The value of the **abbr** attribute is never rendered in place of the contents of the cell.

2.2.13 [HTML] Section 11.3.3, Cell margins

V0086:

The specification states:

```
If a table or given column has a fixed width, cellspacing and cellpadding may
demand more space than assigned. User agents may give these attributes precedence
over the width attribute when a conflict occurs, but are not required to.
```

All Document Modes (All Versions)

Cellspacing and **cellpadding** are given precedence over the **width** attribute when a conflict occurs.

2.2.14 [HTML] Section 12.1.3, Specifying anchors and links

C0013:

The specification states:

```
The A element may only appear in the body.
```

All Document Modes (All Versions)

A elements are recognized and displayed no matter where they are defined in the document. However, **A** elements defined within a **TITLE** element are not recognized.

2.2.15 [HTML] Section 12.1.5, Internationalization and links

C0014:

The specification states:

```
User agents should be able to avoid presenting "garbage" to the user. Instead, they
may either locate resources necessary for the correct presentation of the document
or, if they cannot locate the resources, they should at least warn the user that
the document will be unreadable and explain the cause.
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)

On Windows XP and Windows Server® 2003, Windows® Internet Explorer® will prompt to install a language pack installation when navigating to a page for which a character set is not installed. For example, on an en-US Windows XP computer, navigating to a resource with **charset**=`euc-kr` will prompt for installation of the Korean language pack. However, if the user chooses not to install the language pack, Internet Explorer might fail to display characters correctly without further explanation.

On Windows Vista® and later, the operating system fonts include a much larger set of language characters. In the rare case that a character set is unsupported, Internet Explorer does not display the language pack installation prompt or any other warning to the user.

2.2.16 [HTML] Section 12.2.3, Anchors with the id attribute

C0015:

The specification states:

```
The id attribute, on the other hand, may not contain character references.
```

All Document Modes (All Versions)

The **ID** attribute can contain character references.

2.2.17 [HTML] Section 12.3, Document relationships: the LINK element

C0016:

The specification states:

```
Although LINK has no content, it conveys relationship information that may be
rendered by user agents in a variety of ways (e.g., a tool-bar with a drop-down
menu of links).
```

All Document Modes (All Versions)

Only **LINK** elements that have the following **rel** attribute and **type** attribute values are surfaced:

- **LINK** elements of **rel**="`alternate`" and **type**="`application/atom+xml`" produce ATOM feed detection.
- **LINK** elements of **rel**="`alternate`" and **type**="`application/rss+xml`" produce RSS feed detection.
- **LINK** elements of **rel**="`shortcut icon`" produce a favorites icon display.

All Document Modes (Internet Explorer 8)

Only **LINK** elements that have the following **rel** attribute and **type** attribute values are surfaced:

- **LINK** elements of **rel**="`search`" and **type**="`application/opensearchdescription+xml`" produce search provider discovery.
- **LINK** elements of **rel**="`Stylesheet`" with an additional **rel**="`alternate stylesheet`" produce an alternate style sheet selection in the **Page** menu.
- **LINK** elements of **rel**="`default-slice`" and **type**="`application/x-hatom`" set the default Web Slice.

All Document Modes (Internet Explorer 9)

Only **LINK** elements that have the following **rel** attribute and **type** attribute values are surfaced:

- **LINK** elements of **rel**="`Stylesheet`" with an additional **rel**="`alternate stylesheet`" produce an alternate style sheet selection in the **Page** menu.
- **LINK** elements of **rel**="`default-slice`" and **type**="`application/x-hatom`" set the default Web Slice.

2.2.18 [HTML] Section 12.3.2, Links and external style sheets

V0087:

The specification states:

```
When the LINK element links an external style sheet to a document, the type
attribute specifies the style sheet language and the media attribute specifies the
intended rendering medium or media. User agents may save time by retrieving from
the network only those style sheets that apply to the current device.
```

All Document Modes (All Versions)

All style sheets are retrieved whether or not they apply to the current device.

2.2.19 [HTML] Section 13.3, Generic inclusion: the OBJECT element

C0017:

The specification states:

```
To include images, authors may use the OBJECT element or the IMG element.
```

All Documents Modes (Internet Explorer 8)

The **OBJECT** element can specify an image if the value of the **data** attribute is set to the URI of the image. If the **classid** attribute is also specified, the image is not displayed.

2.2.20 [HTML] Section 13.3.2, Object initialization: the PARAM element

C0018:

The specification states:

```
type = content-type [CI]

This attribute specifies the content type of the resource designated by the value
attribute only in the case where valuetype is set to "ref".
```

All Document Modes (All Versions)

The information contained in the **type** attribute is not used directly. However, **PARAM** elements which do not set a value for the **type** attribute behave as if the **valuetype** attribute is set to `data`.

C0019:

The specification states:

```
type = content-type [CI]

This attribute thus specifies for the user agent, the type of values that will be
found at the URI designated by value.
```

All Document Modes (All Versions)

The information contained in the **type** attribute is not used. The value of this attribute is made available to plug-ins loaded through the **OBJECT** tags. It is the responsibility of the loaded application to conform or not conform to this requirement.

2.2.21 [HTML] Section 13.4, Including an applet: the APPLET element

C0020:

The specification states:

```
archive = uri-list [CT]

The classes are loaded using an instance of an AppletClassLoader with the given
codebase. Relative URIs for archives are interpreted with respect to the applet's
codebase.
```

All Document Modes (All Versions)

The **archive** attribute is ignored.

V0088:

The specification states:

```
object = cdata [CS]

This attribute names a resource containing a serialized representation of an
applet's state. It is interpreted relative to the applet's codebase.
```

All Document Modes (All Versions)

The **object** attribute is ignored.

C0021:

The specification states:

```
When the applet is "deserialized" the start() method is invoked but not the init()
method.
```

All Document Modes (All Versions)

This decision is delegated to Java (or an equivalent plug-in), which chooses whether to call **start()** or **init()**.

2.2.22 [HTML] Section 13.6.1, Client-side image maps: the MAP and AREA elements

C0022:

The specification states:

```
The MAP element specifies a client-side image map (or other navigation mechanism)
that may be associated with another elements (IMG, OBJECT, or INPUT).
```

All Document Modes (All Versions)

An **INPUT** element of type `image` that also specifies a **usemap** attribute acts as a submit button rather than an image map.

C0023:

The specification states:

```
An image map is associated with an element via the element's usemap attribute.
```

All Document Modes (All Versions)

An **INPUT** element of type `image` that also specifies a **usemap** attribute acts as a submit button rather than an image map.

2.2.23 [HTML] Section 14.2, Adding style to HTML

V0089:

The specification states:

```
User agents should determine the default style sheet language for a document
according to the following steps (highest to lowest priority):

1. If any META declarations specify the "Content-Style-Type", the last one in the
character stream determines the default style sheet language.

2. Otherwise, if any HTTP headers specify the "Content-Style-Type", the last one in
the character stream determines the default style sheet language.

3. Otherwise, the default style sheet language is "text/css".
```

All Document Modes (All Versions)

Non-CSS style sheet declarations are ignored.

2.2.24 [HTML] Section 14.2.3, Header style information: the STYLE element

V0090:

The specification states:

```
type = content-type [CI]

This attribute specifies the style sheet language of the element's contents and
overrides the default style sheet language. The style sheet language is specified
as a content type (e.g., "text/css"). Authors must supply a value for this
attribute; there is no default value for this attribute.
```

All Document Modes (All Versions)

If the **type** attribute is not used it defaults to **CSS** and applies the style.

2.2.25 [HTML] Section 14.3.1, Preferred and alternate style sheets

V0091:

The specification states:

```
The author may specify that one of the alternates is a preferred style sheet. User
agents should apply the author's preferred style sheet unless the user has selected
a different alternate.
```

All Document Modes (Internet Explorer 7)

Alternate style sheets are not supported.

V0092:

The specification states:

```
Authors may group several alternate style sheets (including the author's preferred
style sheets) under a single style name. When a user selects a named style, the
user agent must apply all style sheets with that name. User agents must not apply
alternate style sheets with a different style name.
```

All Document Modes (Internet Explorer 7)

Alternate style sheets are not supported.

V0093:

The specification states:

```
Authors may also specify persistent style sheets that user agents must apply in
addition to any alternate style sheet.
```

All Document Modes (Internet Explorer 7)

Alternate style sheets are not supported.

V0094:

The specification states:

```
User agents should also allow users to disable the author's style sheets entirely,
 in which case the user agent must not apply any persistent or alternate style
sheets.
```

All Document Modes (Internet Explorer 7)

The user cannot disable the author’s style sheets.

2.2.26 [HTML] Section 14.3.2, Specifying external style sheets

V0095:

The specification states:

```
User agents should provide a means for users to view and pick from the list of
alternate styles. The value of the title attribute is recommended as the name of
each choice.
```

All Document Modes (Internet Explorer 7)

Alternate style sheets are not supported.

V0096:

The specification states:

```
Authors may also use the META element to set the document's preferred style sheet.
```

All Document Modes (Internet Explorer 7)

Alternate style sheets are not supported.

V0097:

The specification states:

```
The preferred style sheet may also be specified with HTTP headers.
```

All Document Modes (Internet Explorer 7)

Alternate style sheets are not supported.

V0098:

The specification states:

```
If two or more META declarations or HTTP headers specify the preferred style sheet,
 the last one takes precedence. HTTP headers are considered to occur earlier than
the document HEAD for this purpose.
```

Quirks Mode and IE7 Mode (All Versions)

Alternate style sheets are not supported.

V0099:

The specification states:

```
If two or more LINK elements specify a preferred style sheet, the first one takes
precedence.
```

Quirks Mode and IE7 Mode (All Versions)

LINK elements are processed in document order, whether or not a **title** attribute is supplied.

2.2.27 [HTML] Section 14.6, Linking to style sheets with HTTP headers

V0101:

The specification defines Linking to style sheets with HTTP headers:

```
The HTTP Link header has the same effect as a LINK element with the same attributes
and values.
```

All Document Modes (All Versions)

HTTP Link headers are not supported.

V0102:

The specification defines Linking to style sheets with HTTP headers:

```
Multiple Link headers correspond to multiple LINK elements occurring in the same
order.
```

All Document Modes (All Versions)

HTTP Link headers are not supported.

V0103:

The specification defines Linking to style sheets with HTTP headers:

```
It is possible to specify several alternate styles using multiple Link headers, and
then use the rel attribute to determine the default style.
```

All Document Modes (All Versions)

HTTP Link headers are not supported.

V0104:

The specification defines Linking to style sheets with HTTP headers:

```
The quote marks are only needed when the attribute values include whitespace.
```

All Document Modes (All Versions)

HTTP Link headers are not supported.

V0105:

The specification defines Linking to style sheets with HTTP headers:

```
Use SGML entities to reference characters that are otherwise not permitted within
```

```
HTTP or email headers, or that are likely to be affected by transit through
gateways.
```

All Document Modes (All Versions)

HTTP Link headers are not supported.

V0106:

The specification defines Linking to style sheets with HTTP headers:

```
LINK and META elements implied by HTTP headers are defined as occurring before any
explicit LINK and META elements in the document's HEAD.
```

All Document Modes (All Versions)

HTTP Link headers are not supported.

2.2.28 [HTML] Section 16.1, Introduction to frames

V0107:

The specification states:

```
If the user agent can't display frames or is configured not to, it will render the
contents of the NOFRAMES element.
```

All Document Modes (All Versions)

Frameset content is blocked for sites that belong to the **Restricted sites** zone. When this is the case, content in the **NOFRAMES** element is also ignored.

2.2.29 [HTML] Section 16.2.2, The FRAME element

V0108:

The specification states:

```
This attribute provides the user agent with information about the frame border.
Possible values:

1: This value tells the user agent to draw a separator between this frame and every
adjoining frame. This is the default value.

0: This value tells the user agent not to draw a separator between this frame and
every adjoining frame. Note that separators may be drawn next to this frame
nonetheless if specified by other frames.
```

All Document Modes (All Versions)

When the **frameborder** attribute is set to `0`, the style of the **frameborder** changes to light gray and remains visible.

C0024:

The specification states:

```
Attributes defined elsewhere

* id, class (document-wide identifiers)
* title (element title)
* style (inline style information)
* onload, onunload (intrinsic events)
```

All Document Modes (All Versions)

The **onunload** event is not supported for the **FRAME** element.

2.2.30 [HTML] Section 16.3, Specifying target frame information

V0109:

The specification states:

```
target = frame-target [CI]

The target attribute may be set for elements that create links (A, LINK), image
maps (AREA), and forms (FORM).
```

All Document Modes (All Versions)

The value of the **name** attribute for a **FRAME** element is not a valid value for the **target** attribute of a **LINK** element. Identifying a **FRAME** element as the target of a **LINK** element does not expose the content of the **FRAME** element to the **LINK** element.

2.2.31 [HTML] Section 16.3.2, Target semantics

V0110:

The specification states:

```
User agents should determine the target frame in which to load a linked resource
according to the following precedences (highest priority to lowest):

If an element has its target attribute set to a known frame, when the element is
activated (i.e., a link is followed or a form is processed), the resource
designated by the element should be loaded into the target frame.

If an element does not have the target attribute set but the BASE element does, the
BASE element's target attribute determines the frame.

If neither the element nor the BASE element refers to a target, the resource
designated by the element should be loaded into the frame containing the element.

If any target attribute refers to an unknown frame F, the user agent should create
a new window and frame, assign the name F to the frame, and load the resource
designated by the element in the new frame.
```

All Document Modes (All Versions)

If any **target** attribute refers to an unknown frame F, the resource is loaded in a new window. The resource is not loaded in a new frame and hence does not apply the unknown name F to a new frame.

2.2.32 [HTML] Section 16.4.1, The NOFRAMES element

V0111:

The specification states:

```
The NOFRAMES element specifies content that should be displayed only by user agents
that do not support frames or are configured not to display frames. User agents
that support frames must only display the contents of a NOFRAMES declaration when
configured not to display frames. User agents that do not support frames must
display the contents of NOFRAMES in any case.
```

All Document Modes (All Versions)

The content of the **NOFRAMES** element is ignored when a site is a member of the **Restricted sites** zone.

2.2.33 [HTML] Section 16.5, Inline frames: the IFRAME element

V0112:

The specification states:

```
The information to be inserted inline is designated by the src attribute of [the
IFRAME] element. The contents of the IFRAME element, on the other hand, should only
be displayed by user agents that do not support frames or are configured not to
display frames.
```

All Document Modes (All Versions)

The content of the **IFRAME** element is ignored when a site is a member of the **Restricted sites** zone.

V0113:

The specification states:

```
Inline frames may not be resized (and thus, they do not take the noresize
attribute).
```

All Document Modes (All Versions)

The **noresize** attribute for the **IFRAME** element is ignored.

2.2.34 [HTML] Section 17.2, Controls

C0025:

The specification states:

```
A control's initial value does not change. Thus, when a form is reset, each
control's current value is reset to its initial value. If a control does not have
an initial value, the effect of a form reset on that control is undefined.
```

All Document Modes (All Versions)

When a form is reset, controls without default values are set to empty values.

2.2.35 [HTML] Section 17.3, The FORM element

C0026:

The specification states:

```
enctype = content-type [CI]

The value "multipart/form-data" should be used in combination with the INPUT
element, type="file".
```

All Document Modes (All Versions)

Files are only transmitted when the **enctype** attribute is set to `multipart/form-data` and the value of the **type** attribute of the **INPUT** element is set to `file`.

C0027:

The specification states:

```
accept-charset = charset list [CI]

User agents may interpret this value as the character encoding that was used to
transmit the document containing this FORM element.
```

All Document Modes (All Versions)

The value of the **accept-charset** attribute does not identify the character encoding that is used to transmit the document. Only UTF-8 character encoding is recognized.

C0028:

The specification states:

```
accept = content-type-list [CI]

User agents may use this information to filter out non-conforming files when
prompting a user to select files to be sent to the server (cf. the INPUT element
when type="file").
```

All Document Modes (All Versions)

The information in the **accept** attribute is not used to filter out non-conforming files that are to be sent to the server. Users may submit any file type.

C0029:

The specification states:

```
User agents may advise the user of the value of the accept-charset attribute and/or
restrict the user's ability to enter unrecognized characters.
```

All Document Modes (All Versions)

The user is not informed when entering unrecognized characters as based on the value of the **accept-charset** attribute. Additionally, the user is able to enter unrecognized characters as based on the value of the **accept-charset** attribute.

2.2.36 [HTML] Section 17.4, The INPUT element

C0030:

The specification states:

```
value = cdata [CA]

It is optional except when the type attribute has the value "radio" or "checkbox".
```

All Document Modes (All Versions)

The **value** attribute is also optional when the **type** attribute is `radio` or `checkbox`. If not defined in the HTML source, the default value is `on`.

C0031:

The specification states:

```
maxlength = number [CN]

This number may exceed the specified size, in which case the user agent should
offer a scrolling mechanism.
```

All Document Modes (All Versions)

When the value of the **maxlength** attribute exceeds the value of the **size** attribute and the text also exceeds the specified size, the user can scroll in the textbox using the arrow keys. There is no scrollbar control.

2.2.37 [HTML] Section 17.4.1, Control types created with INPUT

C0032:

The specification states:

```
file

User agents may use the value of the value attribute as the initial file name.
```

All Document Modes (All Versions)

The value of the **value** attribute is not used as the initial file name.

2.2.38 [HTML] Section 17.5, The BUTTON element

C0033:

The specification states:

```
Visual user agents may render BUTTON buttons with relief and an up/down motion when
clicked, while they may render INPUT buttons as "flat" images.
```

All Document Modes (All Versions)

IMAGE and **BUTTON** elements are rendered identically, with relief as well as up and down motions.

C0034:

The specification states:

```
It is illegal to associate an image map with an IMG that appears as the contents of
a BUTTON element.
```

All Document Modes (All Versions)

When an image is a child element of a button and defines an image map, the image map is rendered outside the visual boundaries of the button element. When the image map is clicked in this scenario, the **onclick** event associated with the button is executed in lieu of the **onclick** event behavior associated with the image map.

2.2.39 [HTML] Section 17.6, The SELECT, OPTGROUP, and OPTION elements

C0035:

The specification states:

```
size = number [CN]

If a SELECT element is presented as a scrolled list box, this attribute specifies
the number of rows in the list that should be visible at the same time. Visual user
agents are not required to present a SELECT element as a list box; they may use any
other mechanism, such as a drop-down menu.
```

All Document Modes (All Versions)

When a value of the **size** attribute is specified, a **SELECT** element is presented as a scrolling list box. If an invalid value is specified for the **size** attribute, then a **SELECT** element with option groups is presented as a scrolled list box with a **size** of 4. Otherwise, **SELECT** elements are displayed as a drop-down menu by default.

2.2.40 [HTML] Section 17.7, The TEXTAREA element

C0036:

The specification states:

```
cols = number [CN]

Users should be able to enter longer lines than this, so user agents should provide
some means to scroll through the contents of the control when the contents extend
beyond the visible area. User agents may wrap visible text lines to keep long lines
visible without the need for scrolling.
```

All Document Modes (All Versions)

Users can enter lines longer than the value of the **cols** attribute. When the text is longer than the visible rows, Windows® Internet Explorer® wraps visible lines of text and adds a vertical scrollbar control.

2.2.41 [HTML] Section 17.8, The ISINDEX element

C0037:

The specification states:

```
User agents may use the value of the prompt attribute as a title for the prompt.
```

All Document Modes (All Versions)

The value of the **prompt** attribute is allowed as a visible label for the **ISINDEX** element.

2.2.42 [HTML] Section 17.9.1, The LABEL element

V0114:

The specification states:

```
for = idref [CS] (case-sensitive)

This attribute explicitly associates the label being defined with another control.
 When present, the value of this attribute must be the same as the value of the id
attribute of some other control in the same document. When absent, the label being
defined is associated with the element's contents.
```

All Document Modes (All Versions)

The value of the **for** attribute (the **id** of the referenced element) is not case-sensitive. Because the value of the **id** attribute for an element is case-sensitive, the **for** attribute may reference multiple elements. In this case, the last matching element (in document order) is recognized as the referenced element.

C0038:

The specification states:

```
Labels may be rendered by user agents in a number of ways (e.g., visually, read by
speech synthesizers, etc.)
```

All Document Modes (All Versions)

The label is rendered visually.

2.2.43 [HTML] Section 17.10, Adding structure to forms: the FIELDSET and LEGEND elements

V0115:

The specification states:

```
{FIELDSET/LEGEND} Start tag: required, End tag: required
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)

An empty **FIELDSET** with a border is rendered to indicate the presence of a **FIELDSET** element when the start tag is absent but the end tag is present.

V0116:

The specification states:

```
LEGEND Attribute definitions

align = top|bottom|left|right [CI] (case-insensitive)

Deprecated. This attribute specifies the position of the legend with respect to the
fieldset. Possible values:

top: The legend is at the top of the fieldset. This is the default value.
bottom: The legend is at the bottom of the fieldset.
left: The legend is at the left side of the fieldset.
right: The legend is at the right side of the fieldset.
```

All Document Modes (All Versions)

The **align** attribute of the **FIELDSET** element does not support a value of `bottom`.

Quirks Mode, IE7 Mode, and IE9 Mode (All Versions)

A value of `center` for the **align** attribute of the **FIELDSET** element is also supported.

2.2.44 [HTML] Section 17.11.1, Tabbing navigation

V0117:

The specification states:

```
tabindex = number [CN]

This attribute specifies the position of the current element in the tabbing order
for the current document. This value must be a number between 0 and 32767.
```

All Document Modes (All Versions)

The value can be a number less than 0 or greater than 32767. If the value is less than 0 or greater than 32767, the element is excluded from the tab cycle of the document.

The upper bound for the value of the **tabindex** attribute is 32768.

C0039:

The specification states:

```
Tabbing keys. The actual key sequence that causes tabbing navigation or element
activation depends on the configuration of the user agent (e.g., the "tab" key is
used for navigation and the "enter" key is used to activate a selected element).
```

All Document Modes (All Versions)

The TAB key is used to navigate between elements and the SPACEBAR is used to activate buttons and input controls. The ENTER key is used to submit a form or navigate a hyperlink or area. For controls like **textarea**, the ENTER key and SPACEBAR have special behaviors such as adding a new line or a space, respectively.

C0040:

The specification states:

```
User agents may also define key sequences to navigate the tabbing order in reverse.
When the end (or beginning) of the tabbing order is reached, user agents may circle
back to the beginning (or end).
```

All Document Modes (All Versions)

Pressing SHIFT+TAB navigates the tabbing order in reverse. When the beginning of the tabbing order is reached, the tab placement cycles back to the end. Controls such as the Address bar and search box participate in the tab sequence and represent the first elements to receive focus upon cycling back to the end.

2.2.45 [HTML] Section 17.11.2, Access keys

C0041:

The specification states:

```
accesskey = character [CN]

This attribute assigns an access key to an element. An access key is a single
character from the document character set.
```

All Document Modes (All Versions)

If the user incorrectly attempts to assign a string to the **accesskey** attribute, the first character of the string is used as the access key.

C0097:

The specification states:

```
Pressing an access key assigned to an element gives focus to the element.
```

All Document Modes (All Versions)

Pressing an access key that is assigned to an element gives focus to that element. If an access key is assigned to more than one element, pressing the access key gives focus to the first element. Pressing the access key again gives focus to the next element that has the same access key. If no other element has that access key, focus is given to a lowercase version of the access key, if there is one.

C0042:

The specification states:

```
The action that occurs when an element receives focus depends on the element. For
example, when a user activates a link defined by the A element, the user agent
generally follows the link.
```

All Document Modes (All Versions)

When the user presses the access key for a hyperlink, focus is shifted to that hyperlink. The link is navigated only when the user activates the focused link by pressing the ENTER key.

C0043:

The specification states:

```
When a user activates a radio button, the user agent changes the value of the radio
button.
```

All Document Modes (All Versions)

When an access key for a radio button is pressed, the focus is set on that radio button and the radio button is activated at the same time.

C0044:

The specification states:

```
The invocation of access keys depends on the underlying system. For instance, on
machines running MS Windows, one generally has to press the "alt" key in addition
to the access key.
```

All Document Modes (All Versions)

The invocation of any access key is caused by pressing the ALT key simultaneously with the access key.

C0045:

The specification states:

```
The rendering of access keys depends on the user agent. We recommend that authors
```

```
include the access key in label text or wherever the access key is to apply. User
agents should render the value of an access key in such a way as to emphasize its
role and to distinguish it from other characters (e.g., by underlining it).
```

All Document Modes (All Versions)

The value of the access key is not automatically underlined or emphasized. The author must specify the markup to be used for emphasis.

2.2.46 [HTML] Section 17.12.1, Disabled controls

V0118:

The specification states:

```
disabled [CI]

When set for a form control, this boolean attribute disables the control for user
input. When set, the disabled attribute has the following effects on an element:

Disabled controls do not receive focus.
Disabled controls are skipped in tabbing navigation.
Disabled controls cannot be successful.

The following elements support the disabled attribute: BUTTON, INPUT, OPTGROUP,
OPTION, SELECT, and TEXTAREA.
This attribute is inherited but local declarations override the inherited value.
```

All Document Modes (Internet Explorer 7)

Child elements (such as **OPTION** elements within a **SELECT** element) cannot disable themselves when their parent is enabled. In other words, the absence of the disabled attribute is inherited but cannot be overridden locally.

All Document Modes (Internet Explorer 8 and Internet Explorer 9)

The behavior differs from previous versions and modes insofar as child elements cannot override an inherited disabled attribute; if the parent is disabled, the child will be as well. However, child elements also inherit the absence of a disabled attribute (that is, the parent is enabled) and can override this by disabling themselves.

2.2.47 [HTML] Section 17.12.2, Read-only controls

C0046:

The specification states:

```
When set, the readonly attribute has the following effects on an element:

Read-only elements receive focus but cannot be modified by the user.

Read-only elements are included in tabbing navigation.

Read-only elements may be successful.
```

All Document Modes (Internet Explorer 7)

While read-only controls are included in tabbing navigation and can receive focus, the user interface does not indicate that the control is read-only.

2.2.48 [HTML] Section 17.13.1, Form submission method

C0047:

The specification states:

```
Note. The "get" method restricts form data set values to ASCII characters.
```

All Document Modes (All Versions)

When submitting form data with the **get** method, non-ASCII characters are percent-encoded from 8-bit Unicode Transformation Format (UTF-8) octets. For example, the character LATIN CAPITAL LETTER A WITH GRAVE would be represented as `%C3%80`.

2.2.49 [HTML] Section 17.13.2, Successful controls

C0048:

The specification states:

```
The current value of a file select is a list of one or more file names. Upon
submission of the form, the contents of each file are submitted with the rest of
the form data. The file contents are packaged according to the form's content type.
```

All Document Modes (All Versions)

Form content **type** = `multipart/form-data` is supported. Only one file per file select control is supported.

V0119:

The specification states:

```
The current value of an object control is determined by the object's
implementation.
```

Quirks Mode, IE7 Mode, and IE8 Mode (All Versions)

Determining the current value of an object control by the object’s implementation is not supported. Instead, `[object]` is submitted as the value of the submitted **OBJECT** element.

IE9 Mode (All Versions)

Nothing is submitted as the current value of a submitted **OBJECT** element.

C0049:

The specification states:

```
If a control doesn't have a current value when the form is submitted, user agents
are not required to treat it as a successful control.
```

All Document Modes (All Versions)

Even if a control does not have a current value specified, the control is treated as successful, and submission of the control occurs when the form is submitted.

2.2.50 [HTML] Section 17.13.3, Processing form data

C0050:

The specification states:

```
User agents should render the response from the HTTP "get" and "post" transactions.
```

All Document Modes (All Versions)

The response is rendered when the server status code is `200 OK`.

2.2.51 [HTML] Section 17.13.4, Form content types

C0051:

The specification states:

```
Each part may be encoded and the "Content-Transfer-Encoding" header supplied if the
value of that part does not conform to the default (7BIT) encoding (see [RFC2045],
section 6)
```

All Document Modes (All Versions)

Only UTF-8 character encoding is recognized.

C0052:

The specification states:

```
If multiple files are to be returned as the result of a single form entry, they
should be returned as "multipart/mixed" embedded within the "multipart/form-data".
```

All Document Modes (All Versions)

Multiple file uploads from a single form input are not supported. When multiple files are uploaded through multiple file **INPUT** elements, each is a separate form-data entry in the **HTTP REQUEST**.

2.2.52 [HTML] Section 18.2.1, The SCRIPT element

C0053:

The specification states:

```
defer [CI] (case-insensitive)
```

```
When set, this boolean attribute provides a hint to the user agent that the script
is not going to generate any document content (e.g., no "document.write" in
javascript) and thus, the user agent can continue parsing and rendering.
```

All Document Modes (All Versions)

If an author incorrectly adds **document.write()** within a deferred script tag, only the script-generated content will be displayed. Rendering of the remainder of the page will fail.

2.2.53 [HTML] Section 18.2.2, Specifying the scripting language

V0120:

The specification states:

```
Authors should specify the default scripting language for all scripts in a document
by including the following META declaration in the HEAD:

<META http-equiv="Content-Script-Type" content="type">
where "type" is a content type naming the scripting language.
```

All Document Modes (All Versions)

META declaration has no effect and is ignored. Rather, if no type is specified for a script tag, JavaScript is used as a default.

V0121:

The specification states:

```
In the absence of a META declaration, the default can be set by a "Content-Script-
Type" HTTP header.Content-Script-Type: typewhere "type" is again a content type
naming the scripting language.
```

All Document Modes (All Versions)

The **HTTP** header is ignored. If the script tag does not specify a **type** the browser default is used.

V0122:

The specification states:

```
User agents should determine the default scripting language for a document
according to the following steps (highest to lowest priority):1. If any META
declarations specify the "Content-Script-Type", the last one in the character
stream determines the default scripting language.
```

All Document Modes (All Versions)

Any **META** element that specifies an **http-equiv** attribute with a value of `Content-Script-Type` is ignored.

V0123:

The specification states:

```
User agents should determine the default scripting language for a document
according to the following steps (highest to lowest priority):2. Otherwise, if any
HTTP headers specify the "Content-Script-Type", the last one in the character
stream determines the default scripting language.
```

All Document Modes (All Versions)

Any **META** element that specifies an **http-equiv** attribute with a value of `Content-Script-Type` is ignored.

C0054:

The specification states:

```
The type attribute must be specified for each SCRIPT element instance in a
document. The value of the type attribute for a SCRIPT element overrides the
default scripting language for that element.
```

All Document Modes (All Versions)

There is no support for a default scripting language that can be set by the page author. JavaScript is always used as a default unless explicitly overridden by a **SCRIPT** element.

C0055:

The specification states:

```
However, scripts should refer to an element according to its assigned name.
Scripting engines should observe the following precedence rules when identifying an
element: a name attribute takes precedence over an id if both are set. Otherwise,
one or the other may be used.
```

All Document Modes (All Versions)

This requirement does not apply to JavaScript or VBScript. These two languages have separate and distinct ways to identify an element by **id** or **name**, based on the author's requirements.

2.2.54 [HTML] Section 18.2.3, Intrinsic events

V0124:

The specification states:

```
HTML documents are constrained to conform to the HTML DTD both before and after
processing any SCRIPT elements.
```

All Document Modes (All Versions)

Rendering of the page will be attempted even if the markup does not conform to the HTML DTD, irrespective of script execution. For example, using script to insert frames in an HTML 4 Strict page is technically invalid, but the frame will be rendered anyway.

2.2.55 [HTML] Section 18.3.1, The NOSCRIPT element

V0125:

The specification states:

```
The content of a NOSCRIPT element should only be rendered by a script-aware user
agent in the following cases: The user agent is configured not to evaluate scripts.
The user agent doesn't support a scripting language invoked by a SCRIPT element
earlier in the document.
```

All Document Modes (All Versions)

If a language invoked by a **SCRIPT** element earlier in the document is not supported, the **NOSCRIPT** content will not appear.

2.2.56 [HTML] Section 24.3, Character entity references for symbols, mathematical symbols, and Greek letters

C0056:

The specification states:

```
To support these entities, user agents may support full [ISO-10646] or use other
means. Display of glyphs for these characters may be obtained by being able to
display the relevant [ISO-10646] characters or by other means, such as internally
mapping the listed entities, numeric character references, and characters to the
appropriate position in some font that contains the requisite glyphs.
```

All Document Modes (All Versions)

Character entities for symbols, mathematical symbols, and Greek letters is supported by displaying the relevant glyphs from [ISO-10646].

2.3 Error Handling

There are no additional considerations for error handling.

2.4 Security

There are no additional security considerations.

3 Appendix A: Test Suite Failures

This section contains a set of test cases from W3C HTML4 Test Suite [W3C-HTML4-TS] that Internet Explorer, as from version 8 in its default settings, does not pass. Unless otherwise specified, any test case failure from executing this set of test cases does not imply lack of conformance to the HTML specification.

3.1 HTML 4.01 Test: Frame target names must start with an alphabetical character (a-z, A-Z)

Test case: http://www.w3.org/MarkUp/Test/HTML401/current/tests/6_16-BF-01.html

The specification states:

```
Frame target names (%FrameTarget; in the DTD) must begin with an alphabetic
character.  User agents should ignore all other target names.
```

The test case states:

Verify that the user agent ignores target names that do not start with an alphabetical character (a-z, A-Z).

Frameset code:

```
<frameset cols="50%, 50%" rows="50%, 50%">
<frame src="6_16-BF-01a.html">
<frame src="6_16-BF-01b.html" name="two">
<frame src="6_16-BF-01c.html">
<frame src="6_16-BF-01d.html" name="4four">
</frameset>
```

Bottom left frame code:

```
<p>This is 6_16-BF-01c.html</p>
<p>When you click on <a href="6_16-BF-01f.html" target="4four">this link</a>,
6_16-BF-01f.html should load in THIS WINDOW, NOT in the window where 6_16-BF-
01d.html is now. According to the spec, user agents must ignore target names
that do not begin with an alphabetic character.</p>
```

Expected results:

Bottom left frame loads the content of 6_16-BF-01f.html.

Bottom right frame (`target="4four"`) does not load any new content.

Actual results for all document modes (all versions):

Bottom right frame (`target="4four"`) loads the content of 6_16-BF-01f.html.

Conclusion:

The value of the **name** attribute for the bottom right frame of the frameset is set to "4four". Because this string value begins with a numeric character instead of an alphabetic character, the value should be ignored and the content loaded into the bottom left frame (the link container).

Based on the actual results, Internet Explorer fails this test case for all document modes (all versions).

See Also

Section 2.2.5

3.2 HTML 4.01 Test: rowspan attribute for TD and TH

Test case: http://www.w3.org/MarkUp/Test/HTML401/current/tests/11_2_6-BF-01.html

The specification states:

```
rowspan = number [CN]
This attribute specifies the number of rows spanned by the current cell. The
default value of this attribute is one ("1"). The value zero ("0") means that the
cell spans all rows from the current row to the last row of the table section
(THEAD, TBODY, or TFOOT) in which the cell is defined.
```

The test case states:

Verify the functionality of the rowspan attribute for TD and TH.

Table code:

```
<table border="1">

<tr><td>row 1</td><td>default</td><td rowspan="0">zero</td>

<td rowspan="1">one</td><td rowspan="2">two</td>

<td rowspan="3">three</td></tr>

<tr><td>row 2</td></tr>

<tr><td>row 3</td></tr>

</table>

<table border="1">

<tr><th>row 1</th><th>default</th><th rowspan="0">zero</th>

<th rowspan="1">one</th><th rowspan="2">two</th>

<th rowspan="3">three</th></tr>

<tr><th>row 2</th></tr>
```

```
<tr><th>row 3</th></tr>

</table>
```

Expected results:

The third column (`<td rowspan="0">zero</td>, <th rowspan="0">zero</th>`) should span all three rows.

Actual results for all document modes (all versions):

The third column spans only one row.

Conclusion:

The value of the **rowspan** attribute for the table cell is set to `0` (zero). A value of `0` implies that the table cell (**TD** or **TH**) should span all rows.

Based on the actual results, Internet Explorer fails this test case for all document modes (all versions).

See Also

Section 2.1.17

3.3 HTML 4.01 Test: colspan attribute for TD and TH

Test case: http://www.w3.org/MarkUp/Test/HTML401/current/tests/11_2_6-BF-02.html

The specification states:

```
colspan = number [CN]
This attribute specifies the number of columns spanned by the current cell. The
default value of this attribute is one ("1"). The value zero ("0") means that the
cell spans all columns from the current column to the last column of the column
group (COLGROUP) in which the cell is defined.
```

The test case states:

Verify the functionality of the colspan attribute for TD and TH.

Table code:

```
<table border="1">

<tr><td>column 1</td><td>column 2</td><td>column 3</td></tr>

<tr><td>default</td></tr>

<tr><td colspan="0">zero</td></tr>

<tr><td colspan="1">one</td></tr>

<tr><td colspan="2">two</td></tr>

<tr><td colspan="3">three</td></tr>
```

```
</table>

<table border="1">

<tr><th>column 1</th><th>column 2</th><th>column 3</th></tr>

<tr><th>default</th></tr>

<tr><th colspan="0">zero</th></tr>

<tr><th colspan="1">one</th></tr>

<tr><th colspan="2">two</th></tr>

<tr><th colspan="3">three</th></tr>

</table>
```

Expected results:

The third row (`<td rowspan="0">zero</td>, <th colspan="0">zero</th>`) should span all three columns.

Actual results for all document modes (all versions):

The third row spans only one column.

Conclusion:

The value of the **colspan** attribute for the table cell is set to `0` (zero). A value of `0` implies that the table cell should span all columns.

Based on the actual results, Internet Explorer fails this test case for all document modes (all versions).

See Also

Section 2.1.17

3.4 HTML 4.01 Test: Specify an image with an OBJECT element

Test case: http://www.w3.org/MarkUp/Test/HTML401/current/tests/13_3-BF-01.html

The specification states:

```
To include images, authors may use the OBJECT element or the IMG element.
```

The test case states:

Verify the codetype and classid attributes for OBJECT element.

Object code:

```
<p>A JPG image and PNG image, displayed via the OBJECT element, shall appear

after this sentence.</p>
```

```
<p>

<object classid="image.jpg" codetype="image/jpeg" ></object>

<p>

<object classid="image.png" codetype="image/png"></object>
```

Expected results:

A .jpg image followed by a .png image should appear after the opening sentence.

Actual results for all document modes (all versions):

An **OBJECT** element placeholder is rendered for each object instead of the expected images.

Conclusion:

For the first **OBJECT** element, the value of the **classid** attribute is set to `image.jpg` and the value of the **codetype** attribute is set to `image/jpeg`. This specifies that an image object of type jpeg should be displayed.

For the second **OBJECT** element, the value of the **classid** attribute is set to `image.png` and the value of the **codetype** attribute is set to `image/png`. This specifies that an image object of type png should be displayed.

Based on the actual results for both image types, Internet Explorer fails this test case for all document modes (all versions).

See Also

Section 2.1.25

3.5 HTML 4.01 Test: AREA elements are ignored when a MAP element contains mixed content

Test case: http://www.w3.org/MarkUp/Test/HTML401/current/tests/13_6_1-BF-02.html

The specification states:

```
When a MAP element contains mixed content (both AREA elements and block-level
content), user agents must ignore the AREA elements.
```

The test case states:

Verify when a MAP element contains mixed content, AREA elements are ignored.

Object code:

```
<img src="map1.gif" usemap="#map1" alt="map1">


<map name="map1">

<area alt="circle" shape="circle" coords="20,20,12" href="circle.html">
```

```
<area alt="circle2" shape="circle" coords="12,64,7" nohref>

<area alt="poly" shape="poly"

coords="13,44,32,37,49,48,48,37,63,49,43,67,30,55,23,62,25,49"

href="poly.html">

<a id="area1" shape="rect" coords="0, 0, 30, 30"

href="rect.html">rect.html</a>

</map>
```

Expected results:

Because the **MAP** element contains block-level content (the **A** element), the three **AREA** elements should be ignored. There should be no areas of the image that are rendered as hot (can be selected or clicked).

Actual results for all document modes (all versions):

The AREA elements that are specified with an href attribute and value are rendered within the image as hot (can be selected or clicked).

Conclusion:

The **MAP** element contains an **A** element (block-level content). This implies that all **AREA** elements contained within the **MAP** element should be ignored.

Based on the actual results, Internet Explorer fails this test case for all document modes (all versions).

See Also

Section 2.1.31

4 Change Tracking

This section identifies changes that were made to the [MS-HTML401] 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.

5 Index

A

Access keys 56
Adding structure to forms: the FIELDSET and LEGEND elements 55
Adding style to HTML 44
Anchors with the id attribute (section 2.1.22 20, section 2.2.16 41)
Attributes
  borderColor 16
  char 18
  charset 10
  codebase 21
  colspan 16
  declare 22
  lang 10
  rowspan 16
  standby 22
  tabindex 55
  valuetype 24

C

Cell margins 40
Change tracking 70
Character encodings 10
Character entity references for symbols - mathematical symbols - and Greek letters 63
Client-side image maps: the MAP and AREA elements (section 2.1.31 26, section 2.2.22 43)
Colors 9
Column groups: the COLGROUP and COL elements 14
Control types created with INPUT 52
Controlling line breaks 12
Controls 50

D

Disabled controls 58
Document relationships: the LINK element 41

E

Elements
  APPLET (section 2.1.29 25, section 2.2.21 43)
  AREA (section 2.1.31 26, section 2.2.22 43)
  BASE (section 2.1.23 21, section 2.1.24 21)
  BASEFONT 29
  BDO 11
  BLOCKQUOTE (section 2.1.9 11, section 2.2.6 37)
  BUTTON (section 2.1.40 31, section 2.2.38 53)
  CAPTION 13
  COL 14
  COLGROUP 14
  FIELDSET 55
  FONT 29
  FORM (section 2.1.38 30, section 2.2.35 51)
  FRAME 48
  IFRAME (section 2.1.37 30, section 2.2.33 50)
  INPUT (section 2.1.39 31, section 2.2.36 52, section 2.2.51 60)
  INS (section 2.1.11 12, section 2.2.9 37)
  ISINDEX 54
  LABEL 54
  LEGEND 55
  LINK 41
  MAP (section 2.1.31 26, section 2.2.22 43)
  NOFRAMES 50
  NOSCRIPT 63
  OBJECT (section 2.1.25 22, section 2.2.19 42, section 2.2.49 59)
  OPTGROUP 53
  OPTION 53
  PARAM (section 2.1.27 24, section 2.2.20 42)
  PRE 37
  Q (section 2.1.9 11, section 2.2.6 37)
  SCRIPT (section 2.2.52 60, section 2.2.53 61)
  SELECT 53
  STYLE 44
  TABLE 12
  TBODY (section 2.1.14 14, section 2.2.11 38)
  TD (section 2.1.17 16, section 2.2.12 38)
  TEXTAREA (section 2.2.40 53, section 2.2.44 55)
  TFOOT (section 2.1.14 14, section 2.2.11 38)
  TH (section 2.1.17 16, section 2.2.12 38)
  THEAD (section 2.1.14 14, section 2.2.11 38)
  TITLE 10
  TR 16

F

Font modifier elements: FONT and BASEFONT 29
Form content types (section 2.1.43 33, section 2.2.51 60)
Form submission method 59
Frame target names 36

G

Generic inclusion: the OBJECT element (section 2.1.25 22, section 2.2.19 42)
Glossary 6

H

Header style information: the STYLE element 44
Horizontal and vertical alignment 18
Hyphenation 37

I

Including an applet: the APPLET element (section 2.1.29 25, section 2.2.21 43)
Informative references 6

Inheritance of language codes 10
Inline frames: the IFRAME element (section 2.1.37 30, section 2.2.33 50)
Internationalization and links 40
Interpretation of language codes 11
Intrinsic events 62
Introduction 6
Introduction to frames 48
Introduction to tables 38

L

Language codes 34
Lengths 10
Link types 35
Linking to style sheets with HTTP headers 47
Links and external style sheets 42

M

Marking document changes: The INS and DEL elements (section 2.1.11 12, section 2.2.9 37)
Media descriptors 36

N

Nested links are illegal 20
Normative references 6
Notes on embedded documents 26

O

Object declarations and instantiations 25
Object initialization: the PARAM element (section 2.1.27 24, section 2.2.20 42)
Overriding the bidirectional algorithm: the BDO element 11

P

Path information: the BASE element 21
Preferred and alternate style sheets (section 2.1.35 29, section 2.2.25 45)
Preformatted text: The PRE element 37
Pre-selected options 32
Processing form data 60

Q

Quotations: The BLOCKQUOTE and Q elements (section 2.1.9 11, section 2.2.6 37)

R

Read-only controls 58
References
  informative 6
  normative 6
Resolving relative URIs 21
Row groups: the THEAD - TFOOT - and TBODY elements (section 2.1.14 14, section 2.2.11 38)
Rules for rendering objects 23

S

Server-side image maps 28
Setting the default style sheet language 29
SGML basic types 33
Specifying anchors and links (section 2.1.19 19, section 2.2.14 40)
Specifying external style sheets 46
Specifying target frame information 49
Specifying the character encoding 9
Specifying the scripting language 61
Successful controls 59
Syntax of anchor names 19

T

Tabbing navigation 55
Table Captions: The CAPTION element 13
Table cells: The TH and TD elements (section 2.1.17 16, section 2.2.12 38)
Table rows: The TR element 16
Target semantics 49
The BUTTON element (section 2.1.40 31, section 2.2.38 53)
The FORM element (section 2.1.38 30, section 2.2.35 51)
The FRAME element 48
The INPUT element (section 2.1.39 31, section 2.2.36 52)
The ISINDEX element 54
The LABEL element 54
The NOFRAMES element 50
The NOSCRIPT element 63
The SCRIPT element 60
The SELECT - OPTGROUP - and OPTION elements (section 2.1.41 32, section 2.2.39 53)
The TABLE element 12
The TEXTAREA element 53
The TITLE element 10
Tracking changes 70

W

Width and height 28