[MS-ISO10646]:

Microsoft Universal Multiple-Octet Coded Character Set (UCS) Standards Support Document

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
3/26/2010	1.0	New	Released new document.
5/26/2010	1.2	None	Introduced no new technical or language changes.
9/8/2010	1.3	Major	Significantly changed the technical content.
2/10/2011	2.0	None	Introduced no new technical or language changes.
2/22/2012	3.0	Major	Significantly changed the technical content.
7/25/2012	3.1	Minor	Clarified the meaning of the technical content.
6/26/2013	4.0	Major	Significantly changed the technical content.
3/31/2014	4.0	None	No changes to the meaning, language, or formatting of the technical content.
1/22/2015	5.0	Major	Updated for new product version.
7/7/2015	5.1	Minor	Clarified the meaning of the technical content.
11/2/2015	5.1	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2016	5.2	Minor	Clarified the meaning of the technical content.
3/22/2016	5.2	None	No changes to the meaning, language, or formatting of the technical content.
7/19/2016	5.3	Minor	Clarified the meaning of the technical content.
11/2/2016	5.3	None	No changes to the meaning, language, or formatting of the technical content.
3/14/2017	5.3	None	No changes to the meaning, language, or formatting of the technical content.
10/3/2017	5.3	None	No changes to the meaning, language, or formatting of the technical content.
2/22/2018	5.3	None	No changes to the meaning, language, or formatting of the technical content.
3/23/2018	5.3	None	No changes to the meaning, language, or formatting of the technical content.
8/28/2018	5.3	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document describes the level of support provided by Microsoft web browsers for *the ISO/IEC 10646:2003 Information technology -- Universal Multiple-Octet Coded Character Set (UCS)* [ISO-10646], published December 2003.

The [ISO-10646] specification may contain guidance for authors of webpages and browser users, in addition to user agents (browser applications). Statements found in this document apply only to normative requirements in the specification targeted to user agents, not those targeted to authors.

1.1 Glossary

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

1.2 References

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

1.2.1 Normative References

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

[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

1.2.2 Informative References

None.

1.3 Microsoft Implementations

The following Microsoft web browsers implement some portion of [ISO-10646]:

- Windows Internet Explorer 7
- Windows Internet Explorer 8
- Windows Internet Explorer 9
- Windows Internet Explorer 10
- Internet Explorer 11
- Internet Explorer 11 for Windows 10
- Microsoft Edge

Each browser version may implement multiple document rendering modes. The modes vary from one another in support of the standard. The following table lists the document modes supported by each browser version.

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
Internet Explorer 11	Quirks Mode IE7 Mode IE8 Mode IE9 Mode IE10 Mode IE11 Mode
Internet Explorer 11 for Windows 10	Quirks Mode IE7 Mode IE8 Mode IE9 Mode IE10 Mode IE11 Mode
Microsoft Edge	EdgeHTML Mode

For each variation presented in this document there is a list of the document modes and browser versions that exhibit the behavior described by the variation. All combinations of modes and versions that are not listed conform to the specification. For example, the following list for a variation indicates that the variation exists in three document modes in all browser versions that support these modes:

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

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 [ISO-10646], 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 [ISO-10646] and whether they are considered normative or informative.

Sections	Normative/Informative
1-6	Informative
7-33	Normative
Annexes A-D	Normative
Annexes F-U	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 all variations and clarifications for the Microsoft implementation of [ISO-10646].

- Section 2.1 describes normative variations from the MUST requirements of the specification.
- Section 2.2 describes clarifications of the MAY and SHOULD requirements.
- Section 2.3 considers error handling aspects of the implementation.
- Section 2.4 considers security aspects of the implementation.

2.1 Normative Variations

The following subsections describe normative variations from the MUST requirements of [ISO-10646].

2.1.1 [ISO10646] Section 19, Mirrored Characters in a Bidirectional Context

V0001:

The specification states:

```
This character mirroring is not limited to paired characters and shall be applied
to all characters belonging to that class.
```

All Document Modes (All Versions)

Characters for which [ISO-10646] represents the mirrored glyph as a separate code point are mirrored. For characters with no code point for the mirrored glyph, no mirroring is performed. For example, because the character 0028 LEFT PARENTHESIS has the mirrored glyph at code point 0029 RIGHT PARENTHESIS, it is mirrored.

2.1.2 [ISO10646] Section B.1, List of all combining characters

V0002:

The specification contains a list of combining characters that spans several amendments.

All Document Modes (All Versions)

Combining characters in the following ranges are not recognized.

Core Specification

- 0D82-0D83
- 1712-1773 (TAGALOG, HANUNOO, BUHID, TAGBANWA)
- 1920-193B (LIMBU)
- 1D165-1D1AD (MUSICAL)

Amendment 1

- 19B0-19C9 (NEW TAI LUE)
- 1A17-1A1B (BUGINESE)

- A802-A827 (SYLOTI)
- 10A01-10A3A (KHAROSHTHI)
- 1D242-1D244 (GREEK MUSICAL)

Amendment 2

- 07EB-07F3 (NKO)
- 1B00-1B73 (BALINESE)

Amendment 3

- 1B80-1BAA (SUDANESE)
- 1C24-1C37 (LEPCHA)
- A880-A8C4 (SAURASHTRA)
- A926-A92D (KAYAH)
- A947-A953 (REJANG)
- 101FD (PHAISTOS)

Amendment 4

- 0616-061A (ARABIC)
- 1067-108F (MYANMAR)
- A66F-A67D (CYRILLIC)
- AA29-AA4D (CHAM)

The entirety of amendment 5 is not supported.

2.1.3 [ISO10646] Section D.4, Mapping from UCS-4 form to UTF-8 form

V0003:

The specification states:

```
Table D.4 defines in mathematical notation the mapping from the UCS-4 coded
representation form to the UTF-8 coded representation form.
```

All Document Modes (All Versions)

Characters encoded as UTF-8 that have values beyond the range of what can be represented by UTF-16 (up to `0x10FFFF`) have each byte decoded as a separate character.

2.2 Clarifications

The following subsections describe clarifications of the MAY and SHOULD requirements of [ISO-10646].

2.2.1 [ISO10646] Section 14, Implementation Levels

C0001:

The specification states:

```
ISO/IEC 10646 specifies three levels of implementation. Combining characters are
described in clause 25 and listed in annex B.

14.1 Implementation level 1
When implementation level 1 is used, a CC-dataelement shall not contain coded
representations of combining characters (see clause B.1) nor of characters from the
HANGUL JAMO block (see clause 26.1). When implementation level 1 is used the
uniquespelling rule shall apply (see clause 26.2).

14.2 Implementation level 2
When implementation level 2 is used, a CC-dataelement shall not contain coded
representations of characters listed in clause B.2. When implementation level 2 is
used the unique-spelling rule shall apply (see clause 26.2).

14.3 Implementation level 3
When implementation level 3 is used, a CC-dataelement may contain coded
representations of any characters.
```

All Document Modes (All Versions)

Coded representations of characters not allowed in implementation levels 1 or 2 (for example, `0x0483`) are displayed. Therefore, Windows Internet Explorer is considered to be at implementation level 3.

2.2.2 [ISO10646] Section C.6, Unpaired RC-elements: Interpretation by receiving devices

C0002:

The specification states:

```
According to clause C.1 an unpaired RC-element (see clause 4.34) is not in
conformance with the requirements of UTF-16. If a receiving device that has adopted
the UTF-16 form receives an unpaired RC-element because of error conditions either:
* in an originating device, or
* in the interchange between an originating and the receiving device, or
* in the receiving device itself,
then it shall interpret that unpaired RC-element in the same way that it interprets
a character that is outside the adopted subset that has been identified for the
device (see sub-clause 2.3c).
```

All Document Modes (All Versions)

Unpaired RC elements are replaced with the character `0xFFFD`.

2.2.3 [ISO10646] Section D.7, Incorrect sequences of octets: Interpretation by receiving devices

C0003:

The specification states:

```
According to D.2 an octet in the range 00 to 7F or C0 to FB is the first octet of a
UTF-8 sequence, and is followed by the appropriate number (from 0 to 5) of
```

```
continuing octets in the range 80 to BF. Furthermore, octets whose value is FE or
FF are not used; thus they are invalid in UTF-8.

If a CC-data-element includes either:
* a first octet that is not immediately followed by the correct number of
continuing octets, or
* one or more continuing octets that are not required to complete a sequence of
first and continuing octets, or
* an invalid octet,
then according to D.2 such a sequence of octets is not in conformance with the
requirements of UTF-8. It is known as a malformed sequence. If a receiving device
that has adopted the UTF-8 form
receives a malformed sequence, because of error conditions either:
* in an originating device, or
* in the interchange between an originating and a receiving device, or
* in the receiving device itself,
then it shall interpret that malformed sequence in the same way that it interprets
a character that is outside the adopted subset that has been identified for the
device (see sub-clause 2.3c).
```

All Document Modes (All Versions)

Incorrect octets are replaced with the character `0xFFFD`.

2.3 Error Handling

There are no additional error handling considerations.

2.4 Security

There are no additional security considerations.

3 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

4 Index