

[MS-ESI2]:

Microsoft Edge / Internet Explorer ECMA-402 ECMAScript Internationalization API (2nd Edition) Specification Standards Support Document

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation 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 might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

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.

Revision Summary

Date	Revision History	Revision Class	Comments
12/7/2015	1.0	New	Released new document.
3/22/2016	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/19/2016	1.1	Minor	Clarified the meaning of the technical content.
11/2/2016	1.1	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1	Introduction	4
1.1	Glossary	4
1.2	References	4
1.2.1	Normative References	4
1.2.2	Informative References	4
1.3	Microsoft Implementations	4
1.4	Standards Support Requirements	5
1.5	Notation	5
2	Standards Support Statements	6
2.1	Normative Variations	6
2.1.1	[ECMA-402/2] Section 6.4.1 IsValidTimeZoneName (timeZone)	6
2.1.2	[ECMA-402/2] Section 11.1.1 InitializeNumberFormat (numberFormat, locales, options)	6
2.1.3	[ECMA-402/2] Section 11.3.4 Number Format Functions	7
2.1.4	[ECMA-402/2] Section 12.3.4 DateTime Format Functions	8
2.2	Clarifications	8
2.3	Error Handling	8
2.4	Security	8
3	Change Tracking	9
4	Index	10

1 Introduction

This document describes the level of support provided by Microsoft web browsers for the *ECMAScript 2015 Internationalization API* specification [\[ECMA-402/2\]](#), published June 2015. The [\[ECMA-402/2\]](#) specification defines the application programming interface for ECMAScript objects that support programs that need to adapt to the linguistic and cultural conventions used by different human languages and countries.

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.

[ECMA-402/2] ECMA International, "Standard ECMA-402 2nd Edition: ECMAScript 2015 Internationalization API Specification", <http://www.ecma-international.org/ecma-402/2.0/ECMA-402.pdf>

[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 browser versions implement some portion of the [\[ECMA-402/2\]](#) specification:

- Microsoft Edge

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

Browser Version	Document Modes Supported
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)

1.4 Standards Support Requirements

To conform to [\[ECMA-402/2\]](#), 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 [\[ECMA-402/2\]](#) and whether they are considered normative or informative.

Sections	Normative/Informative
1-5	Informative
6	Normative
7	Informative
8-13	Normative
Annex A, Annex 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 all variations and clarifications for the Microsoft implementation of [ECMA-402/2].

- 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 [\[ECMA-402/2\]](#).

2.1.1 [ECMA-402/2] Section 6.4.1 IsValidTimeZoneName (timeZone)

V0122: The only timeZone supported is UTC

The specification states:

```
6.4.1 IsValidTimeZoneName (timeZone)
```

The IsValidTimeZoneName abstract operation verifies that the timeZone argument (which must be a String value) represents a valid Zone or Link name of the IANA Time Zone Database.

The abstract operation returns true if timeZone, converted to upper case as described in 6.1, is equal to one of the Zone or Link names of the IANA Time Zone Database, converted to upper case as described in 6.1. It returns false otherwise.

EdgeHTML Mode (All versions)

The only timeZone supported is UTC. If no timeZone is provided, UTC is assumed.

2.1.2 [ECMA-402/2] Section 11.1.1 InitializeNumberFormat (numberFormat, locales, options)

V0120: CurrencyCode CLF is not supported.

The specification states:

```
11.1.1 InitializeNumberFormat (numberFormat, locales, options)
```

The abstract operation InitializeNumberFormat accepts the arguments numberFormat (which must be an object), locales, and options. It initializes numberFormat as a NumberFormat object.

The following steps are taken:

...

16. Let *s* be `GetOption(options, "style", "string", « "decimal", "percent", "currency"»`, "decimal").

17. ReturnIfAbrupt(s).
18. Set numberFormat.[[style]] to s.
19. Let c be GetOption(options, "currency", "string", undefined, undefined).
20. ReturnIfAbrupt(c).
21. If c is not undefined, then
 - a. If the result of IsWellFormedCurrencyCode(c), is false, then throw a RangeError exception.
22. If s is "currency" and c is undefined, throw a TypeError exception.
23. If s is "currency", then
 - a. Let c be converting c to upper case as specified in 6.1.
 - b. Set numberFormat.[[currency]] to c.
 - c. Let cDigits be CurrencyDigits(c)
24. Let cd be GetOption(options, "currencyDisplay", "string", «"code", "symbol", "name"», "symbol").
25. ReturnIfAbrupt(cd).
26. If s is "currency", set numberFormat.[[currencyDisplay]] to cd.
27. Let mnid be GetNumberOption(options, "minimumIntegerDigits", 1, 21, 1).
28. ReturnIfAbrupt(mnid).
29. Set numberFormat.[[minimumIntegerDigits]] to mnid.
30. If s is "currency", let mnfdDefault be cDigits; else let mnfdDefault be 0.
31. Let mnfd be GetNumberOption(options, "minimumFractionDigits", 0, 20, mnfdDefault).
32. ReturnIfAbrupt(mnfd).
33. Set numberFormat.[[minimumFractionDigits]] to mnfd.
34. If s is "currency", let mxfdDefault be max(mnfd, cDigits); else if s is "percent", let mxfdDefault be max(mnfd, 0); else let mxfdDefault be max(mnfd, 3).

EdgeHTML Mode (All versions)

CurrencyCode CLF is not supported.

2.1.3 [ECMA-402/2] Section 11.3.4 Number Format Functions

V0121: Numbers are not formatted correctly for locale th-u-nu-arab

The specification states:

11.3.4 Number Format Functions

A NumberFormat format function is an anonymous function that takes one argument value, and performs the following steps:

1. Let nf be the this value.
2. Assert: Type(nf) is Object and nf has an [[initializedNumberFormat]] internal slot whose value is true.
3. If value is not provided, let value be undefined.
4. Let x be ToNumber(value).
5. ReturnIfAbrupt(x).
6. Return FormatNumber(nf, x).

When the FormatNumber abstract operation is called with arguments numberFormat (which must be an object initialized as a NumberFormat) and x (which must be a Number value), it returns a String value representing x according to the effective locale and the formatting options of numberFormat.

EdgeHTML Mode (All versions)

Numbers are not formatted correctly for locale th-u-nu-arab.

2.1.4 [ECMA-402/2] Section 12.3.4 DateTime Format Functions

V0123: Old dates are not formatted correctly

The specification states:

12.3.4 DateTime Format Functions

A DateTime format function is an anonymous function that optionally takes an argument date. ...

...

When the FormatDateTime abstract operation is called with arguments dateTimeFormat (which must be an object initialized as a DateTimeFormat) and x (which must be a Number value), it returns a String value representing x (interpreted as a time value as specified in ES2015, 20.3.1.1) according to the effective locale and the formatting options of dateTimeFormat.

EdgeHTML Mode (All versions)

Old dates are not formatted correctly; for example, June 1 BC.

2.2 Clarifications

The following subsections describe clarifications of the MAY and SHOULD requirements of [ECMA-402/2].

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

C

[Change tracking](#) 9

G

[Glossary](#) 4

I

[Informative references](#) 4

[Introduction](#) 4

L

[locales - options](#) 6

N

[Normative references](#) 4

R

References

[informative](#) 4

[normative](#) 4

T

[Tracking changes](#) 9