[MS-CORSXF]:

Internet Explorer Standards Support Cross-Origin Resource Sharing for XDomainRequest, Images, and Fonts 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
7/16/2014	1.0	New	Released new document.
1/22/2015	2.0	Major	Updated for new product version.
7/7/2015	2.1	Minor	Clarified the meaning of the technical content.
11/2/2015	2.1	None	No changes to the meaning, language, or formatting of the technical content.
3/22/2016	2.1	None	No changes to the meaning, language, or formatting of the technical content.
11/2/2016	2.1	None	No changes to the meaning, language, or formatting of the technical content.
3/14/2017	2.1	None	No changes to the meaning, language, or formatting of the technical content.
10/3/2017	2.1	None	No changes to the meaning, language, or formatting of the technical content.
2/22/2018	2.1	None	No changes to the meaning, language, or formatting of the technical content.
3/23/2018	2.1	None	No changes to the meaning, language, or formatting of the technical content.
8/28/2018	2.1	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 *Cross-Origin Resource Sharing* [CORS] specification, published 16 January 2014. The [CORS] specification defines a mechanism to enable client-side cross-origin requests. It is a building block for other specifications, so-called CORS API specifications, which define how the [CORS] specification is used.

This document describes support for aspects of [CORS] pertinent to [XDomainRequest], images [HTML5], and fonts [CSS-FontsLevel3].

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.

[CORS] World Wide Web Consortium, "Cross-Origin Resource Sharing", W3C Recommendation 16 January 2014, http://www.w3.org/TR/2014/REC-cors-20140116/

[CSS-FontsLevel3] World Wide Web Consortium, "CSS Fonts Module Level 3", W3C Candidate Recommendation 3 October 2013, http://www.w3.org/TR/css-fonts-3/

[HTML5] Berjon, R., Faulkner, S., Leithead, T., Navara, E., et al., Eds., "HTML5 -- A vocabulary and associated APIs for HTML and XHTML", http://www.w3.org/TR/html5/

[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

[XDomainRequest] Microsoft Corporation, "XDomainRequest object", http://msdn.microsoft.com/library/ie/cc288060

1.2.2 Informative References

None.

1.3 Microsoft Implementations

The following Microsoft web browser versions implement some portion of the [CORS] specification for [XDomainRequest]:

- Windows Internet Explorer 8
- Windows Internet Explorer 9

- Windows Internet Explorer 10

The following Microsoft web browsers implement some portion of the [CORS] specification for font fetching [CSS-FontsLevel3] and the `canvas` element [HTML5]:

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

The following Microsoft web browsers implement some portion of the [CORS] specification for the `crossorigin` attribute [HTML5]:

- 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 to another in support of the standard. The following table lists the document modes supported by each browser version.

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

Browser Version	Document Modes Supported
	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)

1.4 Standards Support Requirements

To conform to [CORS] 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 [CORS] and whether they are considered normative or informative.

Sections	Normative/Informative
1	Informative
2 - 3	Normative
4	Informative
5 - 6.2	Normative
6.3 - 6.4	Informative
7 - 7.2	Normative
7.3 - 8	Informative
References	Informative
Acknowledgments	Informative

1.5 Notation

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

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

Notation	Explanation
	specification, which can impair interoperability.

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

2 Standards Support Statements

This section contains all variations and clarifications for the Microsoft implementation of [CORS].

- 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 `XDomainRequest` object [XDomainRequest] is supported in IE8 Mode, IE9 Mode, and IE10 Mode (all versions). The `@font-face` rule [CSS-FontsLevel3] is supported in IE9 Mode, IE10 Mode, IE11 Mode, and EdgeHTML Mode (all versions). The `crossorigin` attribute for the `img element` [HTML5] is supported in IE11 Mode and EdgeHTML Mode (all versions) and cross-origin canvas behavior is partially supported in IE9 Mode and IE10 Mode (all versions) and supported in IE11 Mode and EdgeHTML Mode (all versions). All of these features involve cross-origin resource sharing [CORS] behavior, as required by their normative specifications and/or as implemented by Microsoft.

The following subsections describe normative variations from the MUST requirements of [CORS].

2.1.1 Section 5.2, Access-Control-Allow-Credentials Response Header

V0001

The specification states:

```
The Access-Control-Allow-Credentials header indicates whether the response to request can be
exposed when the omit credentials flag is unset. When part of the response to a preflight
request it indicates that the actual request can include user credentials. ABNF:

Access-Control-Allow-Credentials: "Access-Control-Allow-Credentials" ":" true
                            true: %x74.72.75.65 ; "true", case-sensitive
```

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Not supported.

2.1.2 Section 5.3, Access-Control-Expose-Headers Response Header

V0002

The specification states:

```
The Access-Control-Expose-Headers header indicates which headers are safe to expose to the
API of a CORS API specification. ABNF:

Access-Control-Expose-Headers = "Access-Control-Expose-Headers" ":" #field-name
```

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Not supported.

2.1.3 Section 5.4, Access-Control-Max-Age Response Header

V0003

The specification states:

```
The Access-Control-Max-Age header indicates how long the results of a preflight request can
be cached in a preflight result cache. ABNF:

Access-Control-Max-Age = "Access-Control-Max-Age" ":" delta-seconds
```

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Not supported.

2.1.4 Section 5.5, Access-Control-Allow-Methods Response Header

V0004

The specification states:

```
The Access-Control-Allow-Methods header indicates, as part of the response to a preflight
request, which methods can be used during the actual request.

The `Allow` header is not relevant for the purposes of the CORS protocol. ABNF:

Access-Control-Allow-Methods: "Access-Control-Allow-Methods" ":" #Method
```

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Not supported.

2.1.5 Section 5.6, Access-Control-Allow-Headers Response Header

V0005

The specification states:

```
The Access-Control-Allow-Headers header indicates, as part of the response to a preflight
request, which header field names can be used during the actual request. ABNF:

Access-Control-Allow-Headers: "Access-Control-Allow-Headers" ":" #field-name
```

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Not supported.

2.1.6 Section 5.8, Access-Control-Request-Method Request Header

V0006

The specification states:

```
The Access-Control-Request-Method header indicates which method will be used in the actual
request as part of the preflight request. ABNF:
```

```
Access-Control-Request-Method: "Access-Control-Request-Method" ":" Method
```

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Not supported.

2.1.7 Section 5.9, Access-Control-Request-Headers Request Header

V0007

The specification states:

```
The Access-Control-Request-Headers header indicates which headers will be used in the actual
request as part of the preflight request. ABNF:

Access-Control-Request-Headers: "Access-Control-Request-Headers" ":" #field-name
```

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Not supported.

2.1.8 Section 6.1, Simple Cross-Origin Request, Actual Request, and Redirects

V0008

The specification states:

```
    Resources must use the following set of steps to determine which additional headers to
    use in the response:


1.  If the Origin header is not present terminate this set of steps. The request is outside
    the scope of this specification.



2.  If the value of the Origin header is not a case-sensitive match for any of the values in
    list of origins, do not set any additional headers and terminate this set of steps.


    Note: Always matching is acceptable since the list of origins can be unbounded.


3.  If the resource supports credentials add a single Access-Control-Allow-Origin header,
    with the value of the Origin header as value, and add a single Access-Control-Allow-
    Credentials header with the case-sensitive string "true" as value.


    Otherwise, add a single Access-Control-Allow-Origin header, with either the value of the
    Origin header or the string "*" as value.

    Note: The string "*" cannot be used for a resource that supports credentials.
```

4. If the list of exposed headers is not empty add one or more Access-Control-Expose-Headers headers, with as values the header field names given in the list of exposed headers.

   Note: By not adding the appropriate headers resource can also clear the preflight result cache of all entries where origin is a case-sensitive match for the value of the Origin header and url is a case-sensitive match for the URL of the resource.

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Credentials and exposed headers are not supported.

2.1.9 Section 6.2, Preflight Request

V0009

The specification states:

> In response to a preflight request the resource indicates which methods and headers (other than simple methods and simple headers) it is willing to handle and whether it supports credentials.

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Not supported.

2.1.10 Section 7.1.1, Handling a Response to a Cross-Origin Request

V0010

The specification states:

> User agents must filter out all response headers other than those that are a simple response header or of which the field name is an ASCII case-insensitive match for one of the values of the Access-Control-Expose-Headers headers (if any), before exposing response headers to APIs defined in CORS API specifications.
>
> Note: The getResponseHeader() method of XMLHttpRequest will therefore not expose any header not indicated above.

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Not supported.

2.1.11 Section 7.1.2, Cross-Origin Request Status

V0011

The specification states:

> Each cross-origin request has an associated cross-origin request status that CORS API specifications that enable an API to make cross-origin requests can hook into. It can take at most two distinct values over the course of a cross-origin request. The values are:
>
> preflight complete

```
    The user agent is about to make the actual request.


success

    The resource can be shared.


abort error

    The user aborted the request.


network error

    The resource cannot be shared. Also used when a DNS error, TLS negotiation failure, or
    other type of network error occurs. This does not include HTTP responses that indicate
    some type of error, such as HTTP status code 410.
```

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Preflight is not supported.

2.1.12 Section 7.1.3, Source Origin

V0012

The specification states:

```
The source origin is the initial origin that user agents must use for the Origin header.
It can be modified during the redirect steps.
```

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Redirection modification is not supported.

2.1.13 Section 7.1.4, Simple Cross-Origin Request

V0013

The specification states:

```
↪If the manual redirect flag is unset and the response has an HTTP status code of 301,
302, 303, 307, or 308

Apply the redirect steps.
```

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Manual redirection is not supported.

2.1.14 Section 7.1.5, Cross-Origin Request with Preflight

V0014

The specification states:

> To protect resources against cross-origin requests that could not originate from certain user agents before this specification existed a preflight request is made to ensure that the resource is aware of this specification. The result of this request is stored in a preflight result cache.

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Preflight is not supported.

2.1.15 Section 7.1.6, Preflight Result Cache

V0015

The specification states:

> As mentioned, a cross-origin request with preflight uses a preflight result cache. This cache consists of a set of entries.

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

Not supported.

2.1.16 Section 7.1.7, Generic Cross-Origin Request Algorithms

V0016

The specification states:

> Whenever the make a request steps are applied, fetch the request URL from origin source origin using referrer source as override referrer source with the manual redirect flag set, and the block cookies flag set if the omit credentials flag is set. Use method request method, entity body request entity body, including the author request headers, and include user credentials if the omit credentials flag is unset.
>
> Whenever the redirect steps are applied, follow this set of steps:

1. Let original URL be the request URL.
2. Let request URL be the URL conveyed by the Location header in the redirect response.
3. If the request URL <scheme> is not supported, infinite loop precautions are violated, or the user agent does not wish to make the new request for some other reason, apply the network error steps.
4. If the request URL contains the userinfo production apply the network error steps.
5. If the resource sharing check for the current resource returns fail, apply the network error steps.
6. If the request URL origin is not same origin with the original URL origin, set source origin to a globally unique identifier (becomes "null" when transmitted).
7. Transparently follow the redirect while observing the set of request rules.

> Whenever the abort steps are applied, terminate the algorithm that invoked this set of steps and set the cross-origin request status to abort error.

```
Whenever the network error steps are applied, terminate the algorithm that invoked this set
of steps and set the cross-origin request status to network error.

Note: This has no effect on setting of user credentials. I.e. if the block cookies flag is
unset, cookies will be set by the response.

Whenever the cache and network error steps are applied, follow these steps:
```

1. Remove the entries in the preflight result cache where origin field value is a case-sensitive match for source origin and url field value is a case-sensitive match for request URL.

2. Apply the network error steps acting as if the algorithm that invoked the cache and network error steps invoked the network error steps instead.

```
There is a cache match when there is a cache entry in the preflight result cache for which
the following is true:
```

- The origin field value is a case-sensitive match for source origin.
- The url field value is a case-sensitive match for request URL.
- The credentials field value is true and the omit credentials flag is unset, or it is false and the omit credentials flag is set.

```
There is a method cache match when there is a cache entry for which there is a cache match
and the method field value is a case-sensitive match for the given method.

There is a header cache match when there is a cache entry for which there is a cache match
and the header field value is an ASCII case-insensitive match for the given header field
name.
```

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The redirect origin check, method cache, header cache, preflight cache checks, and updates are not supported.

2.1.17 Section 7.2, Resource Sharing Check

V0017

The specification states:

```
4. If the omit credentials flag is unset and the response includes zero or more than one
Access-Control-Allow-Credentials header values, return fail and terminate this algorithm.

5. If the omit credentials flag is unset and the Access-Control-Allow-Credentials header
value is not a case-sensitive match for "true", return fail and terminate this algorithm.
```

IE8 Mode, IE9 Mode, and IE10 Mode (All Versions)

The omit credentials flag is not supported.

2.2 Clarifications

There are no clarifications of the MAY and SHOULD requirements of [CORS].

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