

## [MS-POINTERLOCK]:

# Microsoft Edge / Internet Explorer Pointer Lock 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](mailto:iplg@microsoft.com).
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **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](http://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.

**Support.** For questions and support, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

## Revision Summary

Date	Revision History	Revision Class	Comments
3/14/2017	1.0	New	Released new document.
10/3/2017	1.0	None	No changes to the meaning, language, or formatting of the technical content.
2/22/2018	1.0	None	No changes to the meaning, language, or formatting of the technical content.
3/23/2018	1.0	None	No changes to the meaning, language, or formatting of the technical content.

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
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
<b>2</b>	<b>Standards Support Statements.....</b>	<b>7</b>
2.1	Normative Variations .....	7
2.1.1	[W3C-POINTERLOCK] Section 4. pointerlockchange and pointerlockerror Events ..	7
2.1.2	[W3C-POINTERLOCK] Section 5.1 Methods .....	8
2.1.3	[W3C-POINTERLOCK] Section 6. Extensions to the Document Interface.....	8
2.1.4	[W3C-POINTERLOCK] Section 6.1 Attributes .....	9
2.1.5	[W3C-POINTERLOCK] Section 8. Extensions to the MouseEventInit Dictionary.....	9
2.1.6	[W3C-POINTERLOCK] Section 8.1 Dictionary MouseEventInit Members .....	10
2.2	Clarifications .....	10
2.3	Extensions .....	10
2.4	Error Handling .....	11
2.5	Security .....	11
<b>3</b>	<b>Change Tracking.....</b>	<b>12</b>
<b>4</b>	<b>Index.....</b>	<b>13</b>

# 1 Introduction

This document describes the level of support provided by Microsoft Edge for the Pointer Lock W3C specification [\[W3C-POINTERLOCK\]](#), published 27 October 2016.

The [\[W3C-POINTERLOCK\]](#) specification defines an API that provides scripted access to raw mouse movement data while locking the target of mouse events to a single element and removing the cursor from view. This is an essential input mode for certain classes of applications, especially first person perspective 3D applications and 3D modeling software.

## 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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.

[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>

[W3C-POINTERLOCK] World Wide Web Consortium, "Pointer Lock", W3C Recommendation 27 October 2016, <https://go.microsoft.com/fwlink/p/?linkid=835118>

### 1.2.2 Informative References

None.

## 1.3 Microsoft Implementations

The following Microsoft web browsers implement some portion of the [\[W3C-POINTERLOCK\]](#) specification:

- 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 11	Quirks Mode

Browser Version	Document Modes Supported
	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)*

## 1.4 Standards Support Requirements

To conform to [\[W3C-POINTERLOCK\]](#), 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 [\[W3C-POINTERLOCK\]](#) and whether they are considered normative or informative.

Sections	Normative/Informative
1-3	Informative
4-9	Normative
10-12	Informative
Appendix A	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,

Notation	Explanation
	SHOULD, or RECOMMENDED. (See <a href="#">RFC2119</a> .) 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, clarifications, and extensions for the Microsoft implementation of [\[W3C-POINTERLOCK\]](#).

- 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](#) describes extensions to the requirements.
- Section [2.4](#) considers error handling aspects of the implementation.
- Section [2.5](#) considers security aspects of the implementation.

### 2.1 Normative Variations

The following subsections describe normative variations from the MUST requirements of [\[W3C-POINTERLOCK\]](#).

#### 2.1.1 [W3C-POINTERLOCK] Section 4. pointerlockchange and pointerlockerror Events

V0001: If the same document is already locked when the `requestPointerLock` method is called, no `pointerlockchange` event is sent

The specification states:

##### 4. pointerlockchange and pointerlockerror Events

Two events are used to communicate pointer lock state change or an error in changing state. They are named `pointerlockchange` and `pointerlockerror`. If pointer lock is entered or exited for any reason a `pointerlockchange` event must be sent.

##### 5.1 Methods

```
requestPointerLock
...
If any element (including this one) in the same document is already locked (or
pending lock) the pointer lock target must be updated to this element and a
pointerlockchange event sent.
```

##### 6. Extensions to the Document Interface

```
partial interface Document {
  attribute EventHandler onpointerlockchange;
  ...
};
```

##### 6.1 Attributes

```
onpointerlockchange of type EventHandler
  An event handler [HTML51] for pointerlockchange events.[HTML51]
```

### **EdgeHTML Mode (All versions)**

If the same document is already locked when the `requestPointerLock` method is called, no `pointerlockchange` event is sent. The event is only sent for the initial lock.

## 2.1.2 [W3C-POINTERLOCK] Section 5.1 Methods

V0001: If the same document is already locked when the `requestPointerLock` method is called, no `pointerlockchange` event is sent

The specification states:

### 4. `pointerlockchange` and `pointerlockerror` Events

Two events are used to communicate pointer lock state change or an error in changing state. They are named `pointerlockchange` and `pointerlockerror`. If pointer lock is entered or exited for any reason a `pointerlockchange` event must be sent.

### 5.1 Methods

```
requestPointerLock
...
If any element (including this one) in the same document is already locked (or
pending lock) the pointer lock target must be updated to this element and a
pointerlockchange event sent.
```

### 6. Extensions to the Document Interface

```
partial interface Document {
  attribute EventHandler onpointerlockchange;
  ...
};
```

### 6.1 Attributes

```
onpointerlockchange of type EventHandler
  An event handler [HTML51] for pointerlockchange events.[HTML51]
```

## **EdgeHTML Mode (All versions)**

If the same document is already locked when the `requestPointerLock` method is called, no `pointerlockchange` event is sent. The event is only sent for the initial lock.

## 2.1.3 [W3C-POINTERLOCK] Section 6. Extensions to the Document Interface

V0001: If the same document is already locked when the `requestPointerLock` method is called, no `pointerlockchange` event is sent

The specification states:

### 4. `pointerlockchange` and `pointerlockerror` Events

Two events are used to communicate pointer lock state change or an error in changing state. They are named `pointerlockchange` and `pointerlockerror`. If pointer lock is entered or exited for any reason a `pointerlockchange` event must be sent.

### 5.1 Methods

```
requestPointerLock
...
If any element (including this one) in the same document is already locked (or
pending lock) the pointer lock target must be updated to this element and a
pointerlockchange event sent.
```

### 6. Extensions to the Document Interface



```
partial interface Document {
    attribute EventHandler onpointerlockchange;
    ...
};
```

#### 6.1 Attributes

```
onpointerlockchange of type EventHandler
    An event handler [HTML51] for pointerlockchange events.[HTML51]
```

### **EdgeHTML Mode (All versions)**

If the same document is already locked when the `requestPointerLock` method is called, no `pointerlockchange` event is sent. The event is only sent for the initial lock.

#### **2.1.4 [W3C-POINTERLOCK] Section 6.1 Attributes**

V0001: If the same document is already locked when the `requestPointerLock` method is called, no `pointerlockchange` event is sent

The specification states:

##### 4. `pointerlockchange` and `pointerlockerror` Events

Two events are used to communicate pointer lock state change or an error in changing state. They are named `pointerlockchange` and `pointerlockerror`. If pointer lock is entered or exited for any reason a `pointerlockchange` event must be sent.

##### 5.1 Methods

```
requestPointerLock
    ...
    If any element (including this one) in the same document is already locked (or pending lock) the pointer lock target must be updated to this element and a pointerlockchange event sent.
```

##### 6. Extensions to the Document Interface

```
partial interface Document {
    attribute EventHandler onpointerlockchange;
    ...
};
```

#### 6.1 Attributes

```
onpointerlockchange of type EventHandler
    An event handler [HTML51] for pointerlockchange events.[HTML51]
```

### **EdgeHTML Mode (All versions)**

If the same document is already locked when the `requestPointerLock` method is called, no `pointerlockchange` event is sent. The event is only sent for the initial lock.

#### **2.1.5 [W3C-POINTERLOCK] Section 8. Extensions to the MouseEventInit Dictionary**

V0005: The extensions to `MouseEventInit` are not supported

The specification states:

#### 8. Extensions to the MouseEventInit Dictionary

User agents must extend the MouseEventInit dictionary [UIEVENTS] with two members movementX and movementY used to initialize respective members of MouseEvent.

```
partial dictionary MouseEventInit {  
    long movementX = 0;  
    long movementY = 0;  
};
```

##### 8.1 Dictionary MouseEventInit Members

movementX of type long, defaulting to 0  
movementY of type long, defaulting to 0

### **EdgeHTML Mode (All versions)**

The extensions to MouseEventInit are not supported.

## **2.1.6 [W3C-POINTERLOCK] Section 8.1 Dictionary MouseEventInit Members**

V0005: The extensions to MouseEventInit are not supported

The specification states:

#### 8. Extensions to the MouseEventInit Dictionary

User agents must extend the MouseEventInit dictionary [UIEVENTS] with two members movementX and movementY used to initialize respective members of MouseEvent.

```
partial dictionary MouseEventInit {  
    long movementX = 0;  
    long movementY = 0;  
};
```

##### 8.1 Dictionary MouseEventInit Members

movementX of type long, defaulting to 0  
movementY of type long, defaulting to 0

### **EdgeHTML Mode (All versions)**

The extensions to MouseEventInit are not supported.

## **2.2 Clarifications**

There are no clarifications of the MAY and SHOULD requirements of [\[W3C-POINTERLOCK\]](#).

## **2.3 Extensions**

There are no extensions to the requirements of [\[W3C-POINTERLOCK\]](#).

## **2.4 Error Handling**

There are no additional error handling considerations.

## **2.5 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](#) 12

### G

[Glossary](#) 4

### I

[Informative references](#) 4

[Introduction](#) 4

### N

[Normative references](#) 4

### R

References

[informative](#) 4

[normative](#) 4

### T

[Tracking changes](#) 12