

[MS-INDEXDB]:

Microsoft Edge / Internet Explorer Indexed Database API 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.
- **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.
- **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.

Revision Summary

Date	Revision History	Revision Class	Comments
7/7/2015	1.0	New	Released new document.
11/2/2015	1.1	Minor	Clarified the meaning of the technical content.
12/7/2015	1.2	Minor	Clarified the meaning of the technical content.
3/22/2016	1.2	None	No changes to the meaning, language, or formatting of the technical content.
11/2/2016	1.2	None	No changes to the meaning, language, or formatting of the technical content.
3/14/2017	1.2	None	No changes to the meaning, language, or formatting of the technical content.
10/3/2017	1.2	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.....	7
2.1	Normative Variations	7
2.1.1	[W3C-INDEXDB] Section 3.1.3 Keys	7
2.1.2	[W3C-INDEXDB] Section 3.1.5 Key Path	8
2.1.3	[W3C-INDEXDB] Section 3.1.6 Index	9
2.1.4	[W3C-INDEXDB] Section 3.1.9 Key Range	9
2.1.5	[W3C-INDEXDB] Section 3.1.12 Options Object.....	9
2.1.6	[W3C-INDEXDB] Section 3.1.13 Key Generators.....	10
2.1.7	[W3C-INDEXDB] Section 3.2.1 The IDBRequest Interface	10
2.1.8	[W3C-INDEXDB] Section 3.2.3 Opening a database	11
2.1.9	[W3C-INDEXDB] Section 3.2.4 Database	12
2.1.10	[W3C-INDEXDB] Section 3.2.5 Object Store.....	14
2.1.11	[W3C-INDEXDB] Section 3.2.6 Index	18
2.1.12	[W3C-INDEXDB] Section 3.2.7 Cursor	18
2.2	Clarifications	20
2.3	Error Handling	20
2.4	Security	20
3	Change Tracking.....	21
4	Index.....	22

1 Introduction

This document describes the level of support provided by Microsoft web browsers for the W3C *Indexed Database API* specification [\[W3C-INDEXTDB\]](#), published 08 January 2015. The [\[W3C-INDEXTDB\]](#) specification defines APIs for a database of records holding simple values and hierarchical objects.

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.

[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-INDEXTDB] World Wide Web Consortium, "Indexed Database API", W3C Recommendation 08 January 2015, <http://www.w3.org/TR/2015/REC-IndexedDB-20150108/>

1.2.2 Informative References

None.

1.3 Microsoft Implementations

The following Microsoft web browser versions implement some portion of the [\[W3C-INDEXTDB\]](#) specification:

- 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 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 10	Quirks Mode IE7 Mode

Browser Version	Document Modes Supported
	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)

1.4 Standards Support Requirements

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

Sections	Normative/Informative
1	Informative
2,3	Normative
4,5	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 [\[W3C-INDEXDB\]](#).

- 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 [\[W3C-INDEXDB\]](#).

2.1.1 [W3C-INDEXDB] Section 3.1.3 Keys

V0001: An Array cannot be used as a key.

The specification states:

```
An Array is only a valid key if every item in the array is defined and is a valid key
(i.e. sparse arrays can not be valid keys) and if the Array doesn't directly or
indirectly contain itself. Any non-numeric properties on an Array are ignored, and thus
do not affect whether the Array is a valid key.
```

IE11 Mode and EdgeHTML Mode (All versions)

An Array cannot be used as a key.

V0002: Invalid keys do not throw the correct exception

The specification states:

```
If the value is of type Number, it is only a valid key if it is not NaN. If the value is
of type Date it is only a valid key if its [[PrimitiveValue]] internal property, as
defined by [ECMA-262], is not NaN. Conforming user agents MUST support all valid keys as
keys.
```

IE11 Mode and EdgeHTML Mode (All versions)

A key defined as `null`, `function() {}`, `window`, `{length:0, constructor:Array}`, Array object, or String object throws incorrect errors (`DataCloneError`).

V0003: Invalid keys are incorrectly considered valid

The specification states:

Conforming user agents MUST support all valid keys as keys.

IE11 Mode and EdgeHTML Mode (All versions)

The following keys are incorrectly considered valid and do not throw errors: `new String()`, `new Number()`, `new Date(NaN)`, `new Date(Infinity)`, `/foo/`, and `new RegExp()`.

2.1.2 [W3C-INDEXDB] Section 3.1.5 Key Path

V0004: A sequence<DOMString> key path is not supported

The specification states:

A non-empty sequence<DOMString> containing only DOMStrings conforming to the above requirements.

IE11 Mode and EdgeHTML Mode (All versions)

A sequence<DOMString> key path is not supported and will not behave correctly.

V0005: An empty array used as a key path throws an error

The specification states:

A key path is a DOMString or sequence<DOMString> that defines how to extract a key from a value. A valid key path is one of:

- An empty DOMString.
- An identifier, which is a DOMString matching the IdentifierName production from the ECMAScript Language Specification [ECMA-262].
- A DOMString consisting of two or more identifiers separated by periods (ASCII character code 46).
- A non-empty sequence<DOMString> containing only DOMStrings conforming to the above requirements.

IE11 Mode and EdgeHTML Mode (All versions)

An `InvalidAccessError` is thrown when an empty string or an empty array is used for the key path.

V0006: Invalid key do not throw an exception

The specification states:

A key path is a DOMString or sequence<DOMString> that defines how to extract a key from a value. A valid key path is one of:

- An empty DOMString.
- An identifier, which is a DOMString matching the IdentifierName production from the ECMAScript Language Specification [ECMA-262].
- A DOMString consisting of two or more identifiers separated by periods (ASCII character code 46).
- A non-empty sequence<DOMString> containing only DOMStrings conforming to the above

requirements.

IE11 Mode and EdgeHTML Mode (All versions)

The following types of key paths do not throw exceptions and are considered valid values:

- strings containing non-valid identifier characters (e.g. (,) comma, (*) asterisks, (") quotation mark, (%) percent sign, (/) solidus, (&) ampersand, (!) exclamation mark, (^) circumflex accent)
- identifiers starting with a number

2.1.3 [W3C-INDEXDB] Section 3.1.6 Index

V0007: The `multiEntry` flag is not supported

The specification states:

```
Each index also has a |multiEntry flag.
```

IE11 Mode and EdgeHTML Mode (All versions)

The `multiEntry` flag is not supported.

2.1.4 [W3C-INDEXDB] Section 3.1.9 Key Range

V0008: The `bound` function does not throw the correct exception

The specification states:

```
If either the lower parameter or upper parameter is not valid key, or the lower key is greater than the upper key, or the lower key and upper key match and either of the bounds are open, the implementation MUST throw a DOMException of type DataError.
```

IE11 Mode and EdgeHTML Mode (All versions)

The `bound` function throws an "Invalid argument" exception, not a `DataError` exception.

2.1.5 [W3C-INDEXDB] Section 3.1.12 Options Object

V0009: The `multiEntry` parameter is not supported

The specification states:

```
multiEntry of type boolean, defaulting to false.
```

IE11 Mode and EdgeHTML Mode (All versions)

The `multiEntry` parameter is not supported.

2.1.6 [W3C-INDEXDB] Section 3.1.13 Key Generators

V0010: Incorrect error thrown

The specification states:

When the current number of a key generator reaches above the value 2^{53} (9007199254740992) any attempts to use the key generator to generate a new key will result in a `ConstraintError`. It is still possible to insert records into the object store by specifying an explicit key, however the only way to use a key generator again for the object store is to delete the object store and create a new one.

IE11 Mode and EdgeHTML Mode (All versions)

An `InvalidAccessError`, not a `ConstraintError`, is thrown when the key generator reaches the maximum value.

V0011: Key generators specified as arrays throw a `DataError`

The specification states:

Only specified keys values which are `Number` values affect the current number of the key generator. `Dates` and `Arrays` which contain `Numbers` do not affect the current number of the key generator. Nor do `DOMString` values which could be parsed as numbers. Negative `Numbers` do not affect the current number since they are always lower than the current number.

IE11 Mode and EdgeHTML Mode (All versions)

Key generators specified as arrays throw a `DataError`.

2.1.7 [W3C-INDEXDB] Section 3.2.1 The `IDBRequest` Interface

V0012: The `source` attribute is incorrectly specified as type `any`.

The specification states:

```
interface IDBRequest : EventTarget {
  readonly attribute any result;
  readonly attribute DOMError error;
  readonly attribute (IDBObjectStore or IDBIndex or IDBCursor)? source;
  readonly attribute IDBTransaction transaction;
  readonly attribute IDBRequestReadyState readyState;
  attribute EventHandler onsuccess;
  attribute EventHandler onerror;
};
```

IE11 Mode (All versions)

The `source` attribute is incorrectly specified as type `any`.

V0013: The `readyState` attribute is defined as type `DOMString`, not `IDBRequestReadyState`.

The specification states:

```
interface IDBRequest : EventTarget {
  readonly attribute any result;
  readonly attribute DOMError error;
  readonly attribute (IDBObjectStore or IDBIndex or IDBCursor)? source;
  readonly attribute IDBTransaction transaction;
  readonly attribute DOMString readyState;
  attribute EventHandler onsuccess;
  attribute EventHandler onerror;
};
```

IE11 Mode (All versions)

The `readyState` attribute is defined as type `DOMString`, not `IDBRequestReadyState`.

2.1.8 [W3C-INDEXDB] Section 3.2.3 Opening a database

V0014: The `WorkerUtils` interface does not implement the `IDBEnvironment` interface

The specification states:

```
WorkerUtils implements IDBEnvironment;
```

All instances of the `WorkerUtils` type are defined to also implement the `IDBEnvironment` interface.

IE11 Mode and EdgeHTML Mode (All versions)

The `WorkerUtils` interface does not implement the `IDBEnvironment` interface. Instead it implements the `indexedDB` attribute.

V0015: The `cmp` function throws an "Invalid argument" exception, not a `DataError` exception

The specification states:

If either `first` or `second` is not a valid key, the implementation MUST throw a `DOMException` of type `DataError`.

IE11 Mode and EdgeHTML Mode (All versions)

The `cmp` function throws an "Invalid argument" exception, not a `DataError` exception.

V0016: The `deleteDatabase` success event is of type `Event`, not `IDBVersionChangeEvent`

The specification states:

If the steps above are successful, the implementation MUST set the result of the request to undefined and fire a success event at the request. The event MUST implement the `IDBVersionChangeEvent` interface and have `oldVersion` set to database version and have the

newVersion property set to null.

IE11 Mode and EdgeHTML Mode (All versions)

The deleteDatabase success event is of type Event, not IDBVersionChangeEvent.

V0017: Invalid values passed to the open function throw an InvalidAccessError, not a TypeError

The specification states:

If an error is returned from the steps above, the implementation MUST set the error attribute of the request to a DOMError whose name is the same as the error returned, and dispatch an event at the request. The event MUST use the Event interface and have its [type] set to "error". The event does bubble but is not cancelable. The propagation path of the event is just the request.

IE11 Mode and EdgeHTML Mode (All versions)

Invalid values passed to the open function throw an InvalidAccessError, not a TypeError.

2.1.9 [W3C-INDEXDB] Section 3.2.4 Database

V0018: The version attribute is not the correct type

The specification states:

```
interface IDBDatabase : EventTarget {
  readonly attribute DOMString      name;
  readonly attribute unsigned long long version;
  readonly attribute DOMStringList  objectStoreNames;
  IDBObjectStore createObjectStore (DOMString name, optional IDBObjectStoreParameters
  optionalParameters);
  void          deleteObjectStore (DOMString name);
  IDBTransaction transaction ((DOMString or sequence<DOMString>) storeNames, optional
  IDBTransactionMode mode = "readonly");
  void          close ();
                attribute EventHandler      onabort;
                attribute EventHandler      onerror;
                attribute EventHandler      onversionchange;
};
```

IE11 Mode (All versions)

The version attribute is incorrectly defined as:

```
readonly attribute DOMString version;
```

V0019: The onversionchange event is not supported

The specification states:

```
interface IDBDatabase : EventTarget {
  readonly attribute DOMString      name;
```

```

    readonly    attribute unsigned long long version;
    readonly    attribute DOMStringList      objectStoreNames;
    IDBObjectStore createObjectStore (DOMString name, optional IDBObjectStoreParameters
optionalParameters);
    void        deleteObjectStore (DOMString name);
    IDBTransaction transaction ((DOMString or sequence<DOMString>) storeNames, optional
IDBTransactionMode mode = "readonly");
    void        close ();
                attribute EventHandler      onabort;
                attribute EventHandler      onerror;
                attribute EventHandler      onversionchange;
};

```

IE11 Mode and EdgeHTML Mode (All versions)

The onversionchange event is not supported.

V0020: A DOMException of type InvalidAccessError, not SyntaxError, is thrown.

The specification states:

If the optionalParameters argument is specified and has a keyPath property which is not undefined or null, then set keyPath to the value of this property. If keyPath is not a valid key path, the implementation MUST throw a DOMException of type SyntaxError.

IE11 Mode and EdgeHTML Mode (All versions)

A DOMException of type InvalidAccessError, not SyntaxError, is thrown.

V0021: The keyPath parameter cannot be specified as an array

The specification states:

If the optionalParameters argument is specified and has a keyPath property which is not undefined or null, then set keyPath to the value of this property.

IE11 Mode and EdgeHTML Mode (All versions)

The keyPath parameter cannot be specified as an array.

V0022: The transaction function throws an InvalidAccessError, not a TypeError

The specification states:

If the value for the mode parameter is not "readonly" or "readwrite", the implementation MUST throw a TypeError.

IE11 Mode and EdgeHTML Mode (All versions)

The transaction function throws an InvalidAccessError, not a TypeError.

2.1.10 [W3C-INDEXDB] Section 3.2.5 Object Store

V0023: The keyPath attribute is defined as type DOMString, not as any

The specification states:

```
interface IDBObjectStore {
  readonly attribute DOMString      name;
  readonly attribute any            keyPath;
  readonly attribute DOMStringList  indexNames;
  readonly attribute IDBTransaction transaction;
  readonly attribute boolean        autoIncrement;
  IDBRequest put (any value, optional any key);
  IDBRequest add (any value, optional any key);
  IDBRequest delete (any key);
  IDBRequest get (any key);
  IDBRequest clear ();
  IDBRequest openCursor (optional any range, optional IDBCursorDirection direction =
  "next");
  IDBIndex  createIndex (DOMString name, (DOMString or sequence<DOMString>) keyPath,
  optional IDBIndexParameters optionalParameters);
  IDBIndex  index (DOMString name);
  void      deleteIndex (DOMString indexName);
  IDBRequest count (optional any key);
};
```

IE11 Mode and EdgeHTML Mode (All versions)

The keyPath attribute is defined as type DOMString, not as any:

```
readonly attribute DOMString keyPath;
```

V0037: The autoIncrement attribute is not supported

The specification states:

```
interface IDBObjectStore {
  readonly attribute DOMString      name;
  readonly attribute any            keyPath;
  readonly attribute DOMStringList  indexNames;
  readonly attribute IDBTransaction transaction;
  readonly attribute boolean        autoIncrement;
  IDBRequest put (any value, optional any key);
  IDBRequest add (any value, optional any key);
  IDBRequest delete (any key);
  IDBRequest get (any key);
  IDBRequest clear ();
  IDBRequest openCursor (optional any range, optional IDBCursorDirection direction =
  "next");
  IDBIndex  createIndex (DOMString name, (DOMString or sequence<DOMString>) keyPath,
  optional IDBIndexParameters optionalParameters);
  IDBIndex  index (DOMString name);
  void      deleteIndex (DOMString indexName);
  IDBRequest count (optional any key);
};
```

IE11 Mode and EdgeHTML Mode (All versions)

The autoIncrement attribute is not supported.

V0024: The direction argument of the openCursor function is the wrong type

The specification states:

```
interface IDBObjectStore {
  readonly attribute DOMString      name;
  readonly attribute any             keyPath;
  readonly attribute DOMStringList  indexNames;
  readonly attribute IDBTransaction transaction;
  readonly attribute boolean        autoIncrement;
  IDBRequest put (any value, optional any key);
  IDBRequest add (any value, optional any key);
  IDBRequest delete (any key);
  IDBRequest get (any key);
  IDBRequest clear ();
  IDBRequest openCursor (optional any range, optional IDBCursorDirection direction =
  "next");
  IDBIndex  createIndex (DOMString name, (DOMString or sequence<DOMString>) keyPath,
  optional IDBIndexParameters optionalParameters);
  IDBIndex  index (DOMString name);
  void      deleteIndex (DOMString indexName);
  IDBRequest count (optional any key);
};
```

IE11 Mode (All versions)

The direction argument of the openCursor function is defined as type DOMString, not type IDBCursorDirection:

```
IDBRequest openCursor(optional any range = 0,
  optional DOMString direction = "next");
```

V0025: The keyPath and optionalParameters arguments of the createIndex function are of the wrong types

The specification states:

```
interface IDBObjectStore {
  readonly attribute DOMString      name;
  readonly attribute any             keyPath;
  readonly attribute DOMStringList  indexNames;
  readonly attribute IDBTransaction transaction;
  readonly attribute boolean        autoIncrement;
  IDBRequest put (any value, optional any key);
  IDBRequest add (any value, optional any key);
  IDBRequest delete (any key);
  IDBRequest get (any key);
  IDBRequest clear ();
  IDBRequest openCursor (optional any range, optional IDBCursorDirection direction =
  "next");
  IDBIndex  createIndex (DOMString name, (DOMString or sequence<DOMString>) keyPath,
  optional IDBIndexParameters optionalParameters);
  IDBIndex  index (DOMString name);
  void      deleteIndex (DOMString indexName);
  IDBRequest count (optional any key);
};
```

IE11 Mode (All versions)

The `keyPath` and `optionalParameters` arguments of the `createIndex` function are of the wrong types. They should be (`DOMString` or `sequence<DOMString>`) and `IDBCursorDirection` respectively.

```
IDBIndex createIndex (DOMString name,  
    DOMString keyPath, optional any optionalParameters = 0);
```

V0026: A key path of type `Array` is not supported

The specification states:

If the key path is a `sequence<DOMString>`, the value will be a new `Array`, populated by appending `Strings` equal to each `DOMString` in the sequence.

IE11 Mode and EdgeHTML Mode (All versions)

A key path of type `Array` is not supported.

V0027: The `multiEntry` parameter is not supported

The specification states:

The `options` object whose attributes are optional parameters to this function. `unique` specifies whether the index's unique flag is set. `multiEntry` specifies whether the index's `multiEntry` flag is set.

IE11 Mode and EdgeHTML Mode (All versions)

The `multiEntry` parameter is not supported.

V0028: Cannot create an index with an empty key path

The specification states:

Otherwise, the implementation MUST create a new index in the object store and return an `IDBIndex` object representing it. Set the created index's name to `name` and key path to `keyPath`. Set the created index's `unique` and `multiEntry` flags to the values of the `unique` and `multiEntry` properties in the `optionalParameters` argument.

IE11 Mode and EdgeHTML Mode (All versions)

Cannot create an index with an empty key path.

V0029: An index created using an empty string incorrectly return a null index name

The specification states:

Parameter	Type	Nullable	Optional	Description
<code>name</code>	<code>DOMString</code>	X	X	The name of a new index

IE11 Mode and EdgeHTML Mode (All versions)

An index created using an empty string incorrectly returns a `null` index name.

V0030: The `range` argument of `openCursor` does not properly handle the undefined value.

The specification states:

```
interface IDBObjectStore {
  readonly attribute DOMString      name;
  readonly attribute any             keyPath;
  readonly attribute DOMStringList  indexNames;
  readonly attribute IDBTransaction transaction;
  readonly attribute boolean        autoIncrement;
  IDBRequest put (any value, optional any key);
  IDBRequest add (any value, optional any key);
  IDBRequest delete (any key);
  IDBRequest get (any key);
  IDBRequest clear ();
  IDBRequest openCursor (optional any range, optional IDBCursorDirection direction =
    "next");
  IDBIndex  createIndex (DOMString name, (DOMString or sequence<DOMString>) keyPath,
    optional IDBIndexParameters optionalParameters);
  IDBIndex  index (DOMString name);
  void      deleteIndex (DOMString indexName);
  IDBRequest count (optional any key);
};
```

IE11 Mode (All versions)

The `range` argument of `openCursor` does not properly handle the undefined value.

V0038: The `keyPath` argument of the `createIndex` function is of the wrong type

The specification states:

```
interface IDBObjectStore {
  readonly attribute DOMString      name;
  readonly attribute any             keyPath;
  readonly attribute DOMStringList  indexNames;
  readonly attribute IDBTransaction transaction;
  readonly attribute boolean        autoIncrement;
  IDBRequest put (any value, optional any key);
  IDBRequest add (any value, optional any key);
  IDBRequest delete (any key);
  IDBRequest get (any key);
  IDBRequest clear ();
  IDBRequest openCursor (optional any range, optional IDBCursorDirection direction =
    "next");
  IDBIndex  createIndex (DOMString name, (DOMString or sequence<DOMString>) keyPath,
    optional IDBIndexParameters optionalParameters);
  IDBIndex  index (DOMString name);
  void      deleteIndex (DOMString indexName);
  IDBRequest count (optional any key);
};
```

EdgeHTML Mode (All versions)

The `keyPath` argument of the `createIndex` function is of the wrong type. It should be (`DOMString` or `sequence<DOMString>`).

```
IDBIndex createIndex (DOMString name,  
    DOMString keyPath, optional IDBIndexParameters optionalParameters = 0);
```

2.1.11 [W3C-INDEXDB] Section 3.2.6 Index

V0031: The `multiEntry` attribute is not supported

The specification states:

```
multiEntry of type Boolean, readonly  
    On getting, provide the multiEntry flag of this index.
```

IE11 Mode and EdgeHTML Mode (All versions)

The `multiEntry` attribute is not supported.

V0032: The range arguments of `openCursor` and `openKeyCursor` do not properly handle the undefined value

The specification states:

```
interface IDBIndex {  
    readonly attribute DOMString name;  
    readonly attribute IDBObjectStore objectStore;  
    readonly attribute any keyPath;  
    readonly attribute boolean multiEntry;  
    readonly attribute boolean unique;  
    IDBRequest openCursor (optional any range, optional IDBCursorDirection direction =  
        "next");  
    IDBRequest openKeyCursor (optional any range, optional IDBCursorDirection direction =  
        "next");  
    IDBRequest get (any key);  
    IDBRequest getKey (any key);  
    IDBRequest count (optional any key);  
};
```

IE11 Mode (All versions)

The range arguments of `openCursor` and `openKeyCursor` do not properly handle the undefined value.

2.1.12 [W3C-INDEXDB] Section 3.2.7 Cursor

V0033: The source attribute is defined as the wrong type

The specification states:

```
interface IDBCursor {  
    readonly attribute (IDBObjectStore or IDBIndex) source;
```

```

    readonly attribute IDBCursorDirection direction;
    readonly attribute any key;
    readonly attribute any primaryKey;
    IDBRequest update (any value);
    void advance ([EnforceRange] unsigned long count);
    void continue (optional any key);
    IDBRequest delete ();
};

```

IE11 Mode (All versions)

The source attribute is defined as type any, not as (IDBObjectStore or IDBIndex).

V0034: The direction attribute is defined as type DOMString, not as IDBCursorDirection

The specification states:

```

interface IDBCursor {
    readonly attribute (IDBObjectStore or IDBIndex) source;
    readonly attribute IDBCursorDirection direction;
    readonly attribute any key;
    readonly attribute any primaryKey;
    IDBRequest update (any value);
    void advance ([EnforceRange] unsigned long count);
    void continue (optional any key);
    IDBRequest delete ();
};

```

IE11 Mode (All versions)

The direction attribute is defined as type DOMString, not as IDBCursorDirection.

V0035: The count argument of advance is not the correct type

The specification states:

```

interface IDBCursor {
    readonly attribute (IDBObjectStore or IDBIndex) source;
    readonly attribute IDBCursorDirection direction;
    readonly attribute any key;
    readonly attribute any primaryKey;
    IDBRequest update (any value);
    void advance (long count);
    void continue (optional any key);
    IDBRequest delete ();
};

```

IE11 Mode (All versions)

The count argument of the advance function is of type long, not unsigned long, and the [EnforceRange] flag is not set.

```
void advance(long count);
```

V0036: Passing an invalid argument to `advance` throws an `InvalidAccessError`

The specification states:

```
advance
    If the value for count is 0 (zero), the implementation MUST throw a TypeError.
```

IE11 Mode and EdgeHTML Mode (All versions)

Passing an invalid argument to `advance` throws an `InlineCode[InvalidAccessError]`, not a `[InlineCode[TypeError]]`.

2.2 Clarifications

The following subsections identify clarifications relative to [\[W3C-INDEXDB\]](#).

2.3 Error Handling

There are no additional considerations for error handling.

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](#) 21

G

[Glossary](#) 4

I

[Informative references](#) 4

[Introduction](#) 4

N

[Normative references](#) 4

R

References

[informative](#) 4

[normative](#) 4

T

[Tracking changes](#) 21