[MS-INDEXEDDB-2]:

Microsoft Edge Indexed Database API 2.0 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
6/5/2018	1.0	New	Released new document.
8/28/2018	1.0	None	No changes to the meaning, language, or formatting of the technical content.

Copyright © 2018 Microsoft Corporation

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 [W3C-INDEXEDDB-2] Section 2.4. Keys 6
2.1.2 [W3C-INDEXEDDB-2] Section 2.5. Key Path 7
2.1.3 [W3C-INDEXEDDB-2] Section 2.11. Key Generators 9
2.1.4 [W3C-INDEXEDDB-2] Section 4.3. The IDBFactory interface 10
2.1.5 [W3C-INDEXEDDB-2] Section 4.4. The IDBDatabase interface 11
2.1.6 [W3C-INDEXEDDB-2] Section 4.5. The IDBObjectStore interface 12
2.1.7 [W3C-INDEXEDDB-2] Section 4.6. The IDBIndex interface 15
2.1.8 [W3C-INDEXEDDB-2] Section 4.7. The IDBKeyRange interface 16
2.1.9 [W3C-INDEXEDDB-2] Section 4.8. The IDBCursor interface 17
2.1.10 [W3C-INDEXEDDB-2] Section 4.9. The IDBTransaction interface 17
2.2 Clarifications 18
2.2.1 [W3C-INDEXEDDB-2] Section 2.1.1. Database Connection 18
2.3 Extensions 18
2.4 Error Handling 18
2.5 Security 18

3 Change Tracking 19

4 Index 20

Copyright © 2018 Microsoft Corporation

1 Introduction

This document describes the level of support provided by Microsoft Edge for the *Indexed Database API 2.0* specification [W3C-INDEXEDDB-2], published 30 January 2018. The [W3C-INDEXEDDB-2] specification defines APIs for a database of records holding simple values and hierarchical objects. Each record consists of a key and some value. Records are accessed by key or by an indexed attribute.

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-INDEXEDDB-2] World Wide Web Consortium, "Indexed Database API 2.0", W3C Recommendation, 30 January 2018, https://www.w3.org/TR/2018/REC-IndexedDB-2-20180130/

1.2.2 Informative References

None.

1.3 Microsoft Implementations

The following Microsoft web browsers implement some portion of the [W3C-INDEXEDDB-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)

Copyright © 2018 Microsoft Corporation

1.4 Standards Support Requirements

To conform to [W3C-INDEXEDDB-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 [W3C-INDEXEDDB-2] and whether they are considered normative or informative.

Sections	Normative/Informative
1	Informative
2-7	Normative
8	Informative
9	Normative
10, 11	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.

Copyright © 2018 Microsoft Corporation

2 Standards Support Statements

This section contains all variations, clarifications, and extensions for the Microsoft implementation of [W3C-INDEXEDDB-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 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-INDEXEDDB-2].

2.1.1 [W3C-INDEXEDDB-2] Section 2.4. Keys

V0002: Invalid keys do not throw the correct exception

The specification states:

```
2.4. Keys

    In order to efficiently retrieve records stored in an indexed database, each record
    is organized according to its key.

    A key has an associated type which is one of: number, date, string, binary, or array.
        ...
        The following ECMAScript types are valid keys:
            ▪ Number primitive values, except NaN. This includes Infinity and -Infinity.
            ▪ Date objects, except where the [[DateValue]] internal slot is NaN.
            ▪ String primitive values.
            ▪ ArrayBuffer objects (or views on buffers such as Uint8Array).
            ▪ Array objects, where every item is defined, is itself a valid key, and does
            not directly or indirectly contain itself. This includes empty arrays. Arrays
            can contain other arrays.
```

EdgeHTML Mode

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:

```
2.4. Keys

    In order to efficiently retrieve records stored in an indexed database, each record
    is organized according to its key.

    A key has an associated type which is one of: number, date, string, binary, or array.
```


Copyright © 2018 Microsoft Corporation

```
...
The following ECMAScript types are valid keys:
    ▪ Number primitive values, except NaN. This includes Infinity and -Infinity.
    ▪ Date objects, except where the [[DateValue]] internal slot is NaN.
    ▪ String primitive values.
    ▪ ArrayBuffer objects (or views on buffers such as Uint8Array).
    ▪ Array objects, where every item is defined, is itself a valid key, and does
    not directly or indirectly contain itself. This includes empty arrays. Arrays
    can contain other arrays.
```

EdgeHTML Mode

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()`.

V0039: Binary keys are not supported

The specification states:

```
2.4. Keys

    In order to efficiently retrieve records stored in an indexed database, each record
    is organized according to its key.

    A key has an associated type which is one of: number, date, string, binary, or array.
```

EdgeHTML Mode

Binary keys are not supported.

V0001: An Array cannot be used as a key

The specification states:

```
2.4. Keys

    In order to efficiently retrieve records stored in an indexed database, each record
    is organized according to its key.

    A key has an associated type which is one of: number, date, string, binary, or array.
    ...
    The following ECMAScript types are valid keys:
        …
        ▪ Array objects, where every item is defined, is itself a valid key, and does not
        directly or indirectly contain itself. This includes empty arrays. Arrays can
        contain other arrays.
```

EdgeHTML Mode

An `Array` cannot be used as a key.

Copyright © 2018 Microsoft Corporation

2.1.2 [W3C-INDEXEDDB-2] Section 2.5. Key Path

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

The specification states:

```
... Key Path

    A key path is a [DOMString or sequence<DOMString>/string or list of strings] 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/string] matching the IdentifierName
        production from the ECMAScript Language Specification [ECMA-262].

        • A [DOMString/string] consisting of two or more identifiers separated by periods
        ([ASCII character code 46/U+002E FULL STOP]).

        • A non-empty [sequence<DOMString> containing only DOMStrings/list containing
        only strings] conforming to the above requirements.
```

EdgeHTML Mode

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

V0006: Some types of key paths are considered valid values and do not throw exceptions

The specification states:

```
... Key Path

    A key path is a [DOMString or sequence<DOMString>/string or list of strings] 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/string] matching the IdentifierName
        production from the ECMAScript Language Specification [ECMA-262].

        • A [DOMString/string] consisting of two or more identifiers separated by periods
        ([ASCII character code 46/U+002E FULL STOP]).

        • A non-empty [sequence<DOMString> containing only DOMStrings/list containing
        only strings] conforming to the above requirements.
```

EdgeHTML Mode

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

- 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

V0004: A sequence<DOMString> key path is not supported and will not behave correctly

The specification states:

Copyright © 2018 Microsoft Corporation

```
... Key Path

    A key path is a [DOMString or sequence<DOMString>/string or list of strings] that
    defines how to extract a key from a value. A valid key path is one of:
        ...
        • A non-empty [sequence<DOMString> containing only DOMStrings/list containing
        only strings] conforming to the above requirements.
```

EdgeHTML Mode

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

2.1.3 [W3C-INDEXEDDB-2] Section 2.11. Key Generators

V0010: An InvalidAccessError, not a ConstraintError, is thrown when the key generator reaches the maximum value

The specification states:

```
2.11. Key Generators

    When a object store is created it can be specified to use a key generator. A key
    generator is used to generate keys for records inserted into an object store if not
    otherwise specified.
        …
        When the current number of a key generator reaches above the value 253
        (9007199254740992) any subsequent attempts to use the key generator to generate a
        new key will result in a "ConstraintError" DOMException. ...
```

EdgeHTML Mode

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:

```
2.11. Key Generators

    When a object store is created it can be specified to use a key generator. A key
    generator is used to generate keys for records inserted into an object store if not
    otherwise specified.
        …
            ...
            Only specified keys of type number can affect the current number of the key
            generator. Keys of type date, array (regardless of the other keys they
            contain), binary, or string (regardless of whether they could be parsed as
            numbers) have no effect on the current number of the key generator. Keys of
            type number with value less than 1 do not affect the current number since
            they are always lower than the current number.
```

EdgeHTML Mode

Key generators specified as arrays throw a `DataError`.

Copyright © 2018 Microsoft Corporation

2.1.4 [W3C-INDEXEDDB-2] Section 4.3. The IDBFactory interface

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

The specification states:

```
4.3. The IDBFactory interface
    Database objects are accessed through methods on the IDBFactory interface. ...
    …
        ...
        interface IDBFactory {
            [NewObject] IDBOpenDBRequest open(DOMString name,
                optional [EnforceRange] unsigned long long version);
            …
        };
    ...
    The open(name, version) method, when invoked, must run these steps:
        1. If the value of version is 0 (zero), the implementation MUST throw a TypeError.
        …
        2. Queue a task to run these steps:
            1. If result is an error, set request's result to undefined, set request's
               error to result, set request's done flag, and fire an event named error at
               request with its bubbles and cancelable attributes initialized to true.
```

EdgeHTML Mode

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

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

The specification states:

```
4.3. The IDBFactory interface
    Database objects are accessed through methods on the IDBFactory interface. ...
    …
        ...
        interface IDBFactory {
            …
            short cmp (any first, any second);
        };
        ...
        result = indexedDB . cmp(key1, key2)

            Compares two values as keys. Returns -1 if key1 precedes key2, 1 if key2
            precedes key1, and 0 if the keys are equal.

            Throws a "DataError" DOMException if either input is not a valid key.
```

EdgeHTML Mode

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:

Copyright © 2018 Microsoft Corporation

```
4.3 The IDBFactory interface
    Database objects are accessed through methods on the IDBFactory interface. A single
    object implementing this interface is present in the global scope of environments
    that support Indexed DB operations.
    …
    The deleteDatabase(name) method, when invoked, must run these steps:
        …
        4. Run these steps in parallel:
           …
           2. Queue a task to run these steps:
              ...
              2. Otherwise, set request's result to undefined, set request's done flag,
                 and fire a version change event named success at request with result and
                 null.
```

EdgeHTML Mode

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

2.1.5 [W3C-INDEXEDDB-2] Section 4.4. The IDBDatabase interface

V0021: The keyPath parameter cannot be specified as an array

The specification states:

```
4.4. The IDBDatabase interface

    The IDBDatabase interface represents a connection to a database.
    …
        …
        interface IDBDatabase : EventTarget {
            …
            [NewObject] IDBObjectStore createObjectStore(DOMString name,
                optional IDBObjectStoreParameters options);
            ...
        };
    ...
    The createObjectStore(name, options) method, when invoked, must run these steps:
        …
        4. Let keyPath be options's keyPath member if it is not undefined or null, or
           null otherwise.
```

EdgeHTML Mode

The `keyPath` parameter cannot be specified as an array.

V0040: The onclose event is not supported

The specification states:

```
4.4. The IDBDatabase interface

    The IDBDatabase interface represents a connection to a
    database.[Exposed=(Window,Worker)]
        ...
        interface IDBDatabase : EventTarget {
            ...
            attribute EventHandler onclose;
```


Copyright © 2018 Microsoft Corporation

```
            ...
        };
```

EdgeHTML Mode

The `onclose` event is not supported.

V0019: The onversionchange event is not supported

The specification states:

```
… …
    ...
        interface IDBDatabase : EventTarget {
            …
            attribute EventHandler onversionchange;
        };
```

EdgeHTML Mode

The `onversionchange` event is not supported.

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

The specification states:

```
4.4. The IDBDatabase interface

    The IDBDatabase interface represents a connection to a database.
    …
        …
        interface IDBDatabase : EventTarget {
            …
            [NewObject] IDBTransaction transaction((DOMString or sequence<DOMString>)
            storeNames,
                optional IDBTransactionMode mode = "readonly");
            ...
        };
    ...
    The transaction(storeNames, mode) method, when invoked, must run these steps:
        …
        6. If mode is not "readonly" or "readwrite", throw a TypeError.
```

EdgeHTML Mode

The transaction function throws an `InvalidAccessError`, not a `TypeError`.

2.1.6 [W3C-INDEXEDDB-2] Section 4.5. The IDBObjectStore interface

V0028: An index with an empty key path cannot be created

The specification states:

Copyright © 2018 Microsoft Corporation

```
4.5. The IDBObjectStore interface

    The IDBObjectStore interface represents an object store handle.

        interface IDBObjectStore {
            ...
            [NewObject] IDBIndex   createIndex (DOMString name,
              (DOMString or sequence<DOMString>) keyPath,
               optional IDBIndexParameters optionalParameters);
            ...
        };

        index = store . createIndex(name, keyPath [, options])

            Creates a new index in store with the given name, keyPath and options and
            returns a new IDBIndex. ...
```

EdgeHTML Mode

An index with an empty key path cannot be created.

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

The specification states:

```
4.5. The IDBObjectStore interface

    The IDBObjectStore interface represents an object store handle.

        interface IDBObjectStore {
            ...
            [NewObject] IDBIndex   createIndex (DOMString name,
              (DOMString or sequence<DOMString>) keyPath,
               optional IDBIndexParameters optionalParameters);
            ...
        };

        index = store . createIndex(name, keyPath [, options])

            Creates a new index in store with the given name, keyPath and options and
            returns a new IDBIndex. ...
```

EdgeHTML Mode

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

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

The specification states:

```
interface IDBObjectStore {
            ...
            readonly    attribute any            keyPath;
            ...
        };
```

EdgeHTML Mode

Copyright © 2018 Microsoft Corporation

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

readonly attribute DOMString keyPath;

V0041: The getKey, getAll, getAllKeys, and openKeyCursor functions are not supported

The specification states:

```
4.5. The IDBObjectStore interface

    The IDBObjectStore interface represents an object store handle.
        ...
        interface IDBObjectStore {
            …
            [NewObject] IDBRequest getKey(any query);
            [NewObject] IDBRequest getAll(optional any query,
                optional [EnforceRange] unsigned long count);
            [NewObject] IDBRequest getAllKeys(optional any query,
                optional [EnforceRange] unsigned long count);
            ...
            [NewObject] IDBRequest openKeyCursor(optional any query,
                optional IDBCursorDirection direction = "next");
            ...
        };
```

EdgeHTML Mode

The `getKey`, `getAll`, `getAllKeys`, and `openKeyCursor` functions are not supported.

V0037: The autoIncrement attribute is not supported

The specification states:

```
interface IDBObjectStore {
            ...
            readonly    attribute boolean        autoIncrement;
            ...
        };
```

EdgeHTML Mode

The `autoIncrement` attribute is not supported.

V0027: The multiEntry parameter is not supported

The specification states:

```
4.5. The IDBObjectStore interface

    The IDBObjectStore interface represents an object store handle.

        ...
        interface IDBObjectStore {
            ...
            [NewObject] IDBIndex   createIndex (DOMString name,
```


Copyright © 2018 Microsoft Corporation

```
            (DOMString or sequence<DOMString>) keyPath,
             optional IDBIndexParameters optionalParameters);
        ...
    };

    dictionary IDBIndexParameters {
        …
        boolean multiEntry = false;
    };
```

EdgeHTML Mode

The `multiEntry` parameter is not supported.

V0038: The keyPath argument of the createIndex function is Domstring, but should be (DOMString or sequence<DOMString>)

The specification states:

```
4.5. The IDBObjectStore interface

    The IDBObjectStore interface represents an object store handle.
        ...
        interface IDBObjectStore {
            ...
            [NewObject] IDBIndex createIndex(DOMString name,
              (DOMString or sequence<DOMString>) keyPath,
              optional IDBIndexParameters options);
            ...
        };
```

EdgeHTML Mode

The `keyPath` argument of the `createIndex` function is `Domstring`, but should be (`DOMString` or `sequence<DOMString>`):

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

2.1.7 [W3C-INDEXEDDB-2] Section 4.6. The IDBIndex interface

V0031: The multiEntry attribute is not supported

The specification states:

```
4.6. The IDBIndex interface

    The IDBIndex interface represents an index handle.
        ...
        interface IDBIndex {
            …
            readonly attribute boolean multiEntry;
            …
        };
```


Copyright © 2018 Microsoft Corporation

EdgeHTML Mode

The `multiEntry` attribute is not supported.

V0045: The getAll and getAllKeys functions are not supported

The specification states:

```
4.6. The IDBIndex interface

    The IDBIndex interface represents an index handle.

        interface IDBIndex {
            ...
            [NewObject] IDBRequest getAll(optional any query,
                optional [EnforceRange] unsigned long count);
            [NewObject] IDBRequest getAllKeys(optional any query,
              optional [EnforceRange] unsigned long count);
            ...
        };
```

EdgeHTML Mode

The `getAll` and `getAllKeys` functions are not supported.

2.1.8 [W3C-INDEXEDDB-2] Section 4.7. The IDBKeyRange interface

V0008: The bound function throws an "Invalid argument" exception, not a DataError exception

The specification states:

```
4.7. The IDBKeyRange interface

    The IDBKeyRange interface represents a key range.
        ...
        interface IDBKeyRange
            ...
            [NewObject] static IDBKeyRange bound(any lower,
                any upper,
                optional boolean lowerOpen = false,
                optional boolean upperOpen = false);
        );
    ...
    The bound(lower, upper, lowerOpen, upperOpen) method, when invoked, must run these
    steps:
        ...
        4. If upperKey is invalid, throw a "DataError" DOMException.
        5. If lowerKey is greater than upperKey, throw a "DataError" DOMException.
```

EdgeHTML Mode

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

V0042: The _includes function is not supported

Copyright © 2018 Microsoft Corporation

The specification states:

```
4.7. The IDBKeyRange interface

    The IDBKeyRange interface represents a key range.
        ...
        interface IDBKeyRange {
            ...
            boolean _includes(any key);
            ...
        };
```

EdgeHTML Mode

The `_includes` function is not supported.

2.1.9 [W3C-INDEXEDDB-2] Section 4.8. The IDBCursor interface

V0043: The continuePrimaryKey function is not supported

The specification states:

```
4.7. The IDBKeyRange interface

    The IDBKeyRange interface represents a key range.[Exposed=(Window,Worker)]
        ...
        interface IDBCursor {
            ...
            void continuePrimaryKey(any key, any primaryKey);
            ...
        };
```

EdgeHTML Mode

The `continuePrimaryKey` function is not supported.

2.1.10 [W3C-INDEXEDDB-2] Section 4.9. The IDBTransaction interface

V0044: The objectStoreNames attribute is not supported

The specification states:

```
4.9. The IDBTransaction interface

    transaction objects implement the following interface:
        …
        interface IDBTransaction : EventTarget {
            readonly attribute DOMStringList objectStoreNames;
            ...
        };
```

EdgeHTML Mode

The objectStoreNames attribute is not supported.

Copyright © 2018 Microsoft Corporation

2.2 Clarifications

The following subsections describe clarifications of the MAY and SHOULD requirements of [W3C-INDEXEDDB-2].

2.2.1 [W3C-INDEXEDDB-2] Section 2.1.1. Database Connection

C0001: A connection is not closed for exceptional circumstances

The specification states:

```
2.1.1. Database Connection
    ...
    A connection may be closed by a user agent in exceptional circumstances, for example
    due to loss of access to the file system, a permission change, or clearing of the
    origin's storage. If this occurs the user agent must run the steps to close a
    database connection with the connection and with the forced flag set.
```

EdgeHTML Mode

A connection is not closed for exceptional circumstances.

2.3 Extensions

There are no extensions to the requirements of [W3C-INDEXEDDB-2].

2.4 Error Handling

There are no additional error handling considerations.

2.5 Security

There are no additional security considerations.

Copyright © 2018 Microsoft Corporation

3 Change Tracking

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

Copyright © 2018 Microsoft Corporation

4 Index

C

Change tracking 19

G

Glossary 4

I

Informative references 4
Introduction 4

N

Normative references 4

R

References
  informative 4
  normative 4

T

Tracking changes 19

Copyright © 2018 Microsoft Corporation