[MS-XMLSD]:
Microsoft XML Schema (Part 2: Datatypes) Standards Support Document

Revision Summary

Date	Revision History	Revision Class	Comments
09/08/2010	0.1	New	Released new document.
10/13/2010	0.2	Minor	Clarified the meaning of the technical content.
02/10/2011	1.0	No change	Introduced no new technical or language changes.
02/22/2012	2.0	Major	Significantly changed the technical content.
07/25/2012	2.1	Minor	Clarified the meaning of the technical content.

Table of Contents

Release: July 25, 2012

1   Introduction

This document describes the level of support provided by the Microsoft XML Core Services (MSXML) 6.0 for *XML Schema Part 2: Datatypes (Second Edition)* [W3C-XSD], published on 28 October, 2004.

MSXML6 supports *XML Schema Part 2: Datatypes (Second Edition)* [W3C-XSD] using the *Extensible Markup Language (XML) 1.0 (Fourth Edition)* [XML], W3C Recommendation 16 August 2006, edited in place 29 September 2006.

The [W3C-XSD] specification may contain guidance for authors of webpages and browser users, in addition to user agents (browser applications). Statements found in this document apply only to normative requirements in the specification targeted to user agents, not those targeted to authors.

1.1   Glossary

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2   References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

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. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[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-XSD] World Wide Web Consortium, "XML Schema Part 2: Datatypes Second Edition", October 2004, http://www.w3.org/TR/2004/REC-xmlschema-2-20041028

[XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", W3C Recommendation, August 2006, http://www.w3.org/TR/2006/REC-xml-20060816/

1.2.2   Informative References

None.

1.3   Microsoft Implementations

Throughout this document, Microsoft XML Core Services (MSXML) 6.0 is referred to as *MSXML6*.

MSXML6 implements the [W3C-XSD] specification.

The following Microsoft products implement some portion of the [W3C-XSD] specification:

- Windows® Internet Explorer® 9

- Windows® Internet Explorer® 10

In addition, each version of Windows® Internet Explorer® implements multiple document modes, which can vary individually in their support of the standard. The following table lists the document modes available in each version of Internet Explorer.

Browser version	Document modes supported
Internet Explorer 9	Quirks mode IE7 mode IE8 mode IE9 mode
Internet Explorer 10 (Beta)	Quirks mode IE7 mode IE8 mode IE9 mode IE10 Mode

Throughout this document, the document mode appears first followed by the browser version in parentheses. Only those document modes and versions of Internet Explorer for which there is a variation note will be listed. If the document mode is not listed, conformance to the specification can be assumed.

Note "Standards mode" in Windows® Internet Explorer® 7 and "IE7 mode" in Windows® Internet Explorer® 8 refer to the same document mode. "IE7 mode" is the preferred way of referring to this document mode across all versions of the browser.

1.4 Standards Support Requirements

To conform to [W3C-XSD], 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].)

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 specification, which can impair interoperability.

For document mode and browser version notation, see 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-XSD].

- 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-XSD].

2.1.1 [W3C-XSD] Section 2.5.1.2, List datatypes

V0001:

The specification states:

```
When a datatype is derived from a list datatype, the following
constraining facets apply:
• length
• maxLength
• minLength
• enumeration
• pattern
• whiteSpace
For each of length, maxLength, and minLength, the unit of length is measured in
number of list items. The value of whiteSpace is fixed to the value collapse.
```

MSXML6

Other values (`preserve` or `replace`) can be assigned to the **whiteSpace** facet when the base type of the list data type is **string**.

2.1.2 [W3C-XSD] Section 3.2.4, float

V0002:

The specification states:

```
[Definition:]  float is patterned after the IEEE single-precision 32-bit floating
point type [IEEE 754-1985]. The basic ·value space· of float consists of the
values m × 2^e, where m is an integer whose absolute value is less than 2^24,
and e is an integer between -149 and 104, inclusive. In addition to the basic
·value space· described above, the ·value space· of float also contains the
following three special values: positive and negative infinity and not-a-number
(NaN). The ·order-relation· on float is: x < y iff y - x is positive for x and y
in the value space. Positive infinity is greater than all other non-NaN values.
NaN equals itself but is ·incomparable· with (neither greater than nor less
```

Release: July 25, 2012

than) any other value in the ·value space·.

MSXML6

No error is reported if the `NaN` value is compared with positive infinity or negative infinity.

2.1.3   [W3C-XSD] Section 3.2.5, double

V0003:

The specification states:

```
[Definition:]  The double datatype is patterned after the IEEE double-precision
64-bit floating point type [IEEE 754-1985]. The basic ·value space· of double
consists of the values m × 2^e, where m is an integer whose absolute value is
less than 2^53, and e is an integer between -1075 and 970, inclusive. In addition
to the basic ·value space· described above, the ·value space· of double also
contains the following three special values: positive and negative infinity and
not-a-number (NaN). The ·order-relation· on double is: x < y iff y - x is
positive for x and y in the value space. Positive infinity is greater than all
other non-NaN values. NaN equals itself but is ·incomparable· with (neither
greater than nor less than) any other value in the ·value space·.
```

MSXML6

No error is reported if the `NaN` value is compared with positive infinity or negative infinity.

V0004:

The specification states:

```
Any value ·incomparable· with the value used for the four bounding facets
(·minInclusive·, ·maxInclusive·, ·minExclusive·, and ·maxExclusive·) will be
excluded from the resulting restricted ·value space·. In particular, when "NaN"
is used as a facet value for a bounding facet, since no other double values are
·comparable· with it, the result is a ·value space· either having NaN as its only
member (the inclusive cases) or that is empty (the exclusive cases). If any other
value is used for a bounding facet, NaN will be excluded from the resulting
restricted ·value space·; to add NaN back in requires union with the NaN-only
space.
```

MSXML6

If `NaN` is used as a facet value for a bounding facet, when `NaN` is compared to other non-`NaN` doubles, `NaN` is less than any non-`NaN` number.

2.1.4   [W3C-XSD] Section 3.2.6.2, Order relation on duration

V0005:

The specification states:

```
Implementations are free to optimize the computation of the ordering relationship.
For example, the following table can be used to compare durations of a small
```

```
number of months against days.
```

MSXML6

Some comparisons that are indeterminate are considered to be `true`, such as the comparison between 1 month and 30 days.

2.1.5   [W3C-XSD] Section 3.2.6.3, Facet Comparison for durations

V0006:

The specification states:

```
In comparing duration values with minInclusive, minExclusive, maxInclusive
and maxExclusive facet values indeterminate comparisons should be considered
as "false".
```

MSXML6

Some comparisons that are indeterminate are considered to be `true`, such as P30D and P1M.

2.1.6   [W3C-XSD] Section 3.2.7.1, Lexical representation

V0007:

The specification states:

```
The lexical space· of dateTime consists of finite-length sequences of characters of the form:
'-'? yyyy '-' mm '-' dd 'T' hh ':' mm ':' ss ('.' s+)? (zzzzzz)?, where

- '-'? yyyy is a four-or-more digit optionally negative-signed numeral that represents the
year; if more than four digits, leading zeros are prohibited, and '0000' is prohibited (see
the Note above (§3.2.7); also note that a plus sign is not permitted);
- the remaining '-'s are separators between parts of the date portion;
the first mm is a two-digit numeral that represents the month;
- dd is a two-digit numeral that represents the day;
- 'T' is a separator indicating that time-of-day follows;
= hh is a two-digit numeral that represents the hour; '24' is permitted if the minutes and
seconds represented are zero, and the dateTime value so represented is the first instant of
the following day (the hour property of a dateTime object in the ·value space· cannot have a
value greater than 23);
- ':' is a separator between parts of the time-of-day portion;
- the second mm is a two-digit numeral that represents the minute;
- ss is a two-integer-digit numeral that represents the whole seconds;
= '.' s+ (if present) represents the fractional seconds;
- zzzzzz (if present) represents the timezone (as described below).
```

MSXML6, IE9 Mode and IE10 Mode (All Versions)

Leading zeros are accepted if *yyyy* is more than four digits, such as 01987-10-12T00:00:00.

2.1.7   [W3C-XSD] Section 4.1.2.2, Derivation by list

V0008:

The specification states:

```
For each of ·length·, ·maxLength· and ·minLength·, the unit of length is measured
in number of list items. The value of ·whiteSpace· is fixed to the value collapse.
```

MSXML6

The **whiteSpace** facet may have other values in addition to `collapse`.

2.1.8   [W3C-XSD] Section 4.3.1.4, Constraints on length Schema Components

V0009:

The specification states:

```
Schema Component Constraint: length and minLength or maxLength

If length is a member of {facets} then
1 It is an error for minLength to be a member of {facets} unless
1.1 the {value} of minLength <= the {value} of length and
1.2 there is type definition from which this one is derived by one or more restriction steps
in which minLength has the same {value} and length is not specified.
2 It is an error for maxLength to be a member of {facets} unless
2.1 the {value} of length <= the {value} of maxLength and
2.2 there is type definition from which this one is derived by one or more restriction steps
in which maxLength has the same {value} and length is not specified.
```

MSXML6

An error is reported when the **length** facet is a member of the **{facets}** set and either the **minLength** or **maxLength** facet is also a member of **{facets}**.

V0010:

The specification states:

```
Schema Component Constraint: length valid restriction

It is an ·error· if length is among the members of {facets} of {base type
definition} and {value} is not equal to the {value} of the parent length.
```

MSXML6

No error is reported if the **length** facet value is different from the value of the parent **length** facet, unless one of the **length** facet values is `fixed`.

2.2   Clarifications

The following subsections identify clarifications to [W3C-XSD].

2.2.1   [W3C-XSD] Section 3.2.6.2, Order relation on duration

V0011:

The specification states:

```
In general, the ·order-relation· on duration is a partial order since there is
no determinate relationship between certain durations such as one month (P1M) and
30 days (P30D). The ·order-relation· of two duration values x and y is x < y iff
s+x < s+y for each qualified dateTime s in the list below. These values for s
cause the greatest deviations in the addition of dateTimes and durations. Addition
of durations to time instants is defined in Adding durations to dateTimes (§E).
```

MSXML6

Some comparisons that are indeterminate are considered to be true, such as the comparison between 1 month and 30 days.

2.2.2 [W3C-XSD] Section 3.2.11, gYear

C0001:

The specification states:

```
Since the lexical representation allows an optional time zone indicator, gYear
values are partially ordered because it may not be possible to unequivocally
determine the order of two values one of which has a time zone and the other does
not. If gYear values are considered as periods of time, the order relation on
gYear values is the order relation on their starting instants. This is discussed
in Order relation on dateTime (§3.2.7.4). See also Adding durations to dateTimes
(§E). Pairs of gYear values with or without time zone indicators are totally
ordered.
```

MSXML6

Indeterminate comparisons, such 2009Z with 2010, are considered to be determinate.

2.2.3 [W3C-XSD] Section 3.2.12, gMonthDay

C0002:

The specification states:

```
Since the lexical representation allows an optional time zone indicator, gMonthDay
values are partially ordered because it may not be possible to unequivocally
determine the order of two values one of which has a time zone and the other does
not. If gMonthDay values are considered as periods of time, in an arbitrary leap
year, the order relation on gMonthDay values is the order relation on their
starting instants. This is discussed in Order relation on dateTime (§3.2.7.4).
See also Adding durations to dateTimes (§E). Pairs of gMonthDay values with or
without time zone indicators are totally ordered.
```

MSXML6

Indeterminate comparisons are considered to be determinate.

2.2.4 [W3C-XSD] Section 3.2.13, gDay

C0003:

The specification states:

> Since the lexical representation allows an optional time zone indicator, gDay
> values are partially ordered because it may not be possible to unequivocally
> determine the order of two values one of which has a time zone and the other does
> not. If gDay values are considered as periods of time, in an arbitrary month that
> has 31 days, the order relation on gDay values is the order relation on their
> starting instants. This is discussed in Order relation on dateTime (§3.2.7.4).
> See also Adding durations to dateTimes (§E). Pairs of gDay values with or without
> time zone indicators are totally ordered.

MSXML6

Indeterminate comparisons are considered to be determinate.

2.2.5 [W3C-XSD] Section 3.2.14, gMonth

C0004:

The specification states:

> Since the lexical representation allows an optional time zone indicator, gMonth
> values are partially ordered because it may not be possible to unequivocally
> determine the order of two values one of which has a time zone and the other
> does not. If gMonth values are considered as periods of time, the order relation
> on gMonth is the order relation on their starting instants. This is discussed in
> Order relation on dateTime (§3.2.7.4). See also Adding durations to dateTimes (§E).
> Pairs of gMonth values with or without time zone indicators are totally ordered.

MSXML6

Indeterminate comparisons are considered to be determinate.

2.3 Error Handling

There are no additional error handling considerations.

2.4 Security

There are no additional security considerations.

Release: July 25, 2012

3 Change Tracking

This section identifies changes that were made to the [MS-XMLSD] protocol document between the February 2012 and July 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.

- An extensive rewrite, addition, or deletion of major portions of content.

- The removal of a document from the documentation set.

- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.

- Content updated.

- Content removed.

- New product behavior note added.

- Product behavior note updated.

- Product behavior note removed.

- New protocol syntax added.

- Protocol syntax updated.

- Protocol syntax removed.

- New content added due to protocol revision.

- Content updated due to protocol revision.

- Content removed due to protocol revision.

- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.

- Protocol syntax removed due to protocol revision.

- New content added for template compliance.

- Content updated for template compliance.

- Content removed for template compliance.

- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1 Introduction	Updated document to remove beta tagging.	N	Content updated.

4 Index