

SSML document structure and events

Article • 09/24/2024

The Speech Synthesis Markup Language (SSML) with input text determines the structure, content, and other characteristics of the text to speech output. For example, you can use SSML to define a paragraph, a sentence, a break or a pause, or silence. You can wrap text with event tags such as bookmark or viseme that can be processed later by your application.

Refer to the sections below for details about how to structure elements in the SSML document.

Document structure

The Speech service implementation of SSML is based on the World Wide Web Consortium's [Speech Synthesis Markup Language Version 1.0](#). The elements supported by the Speech can differ from the W3C standard.

Each SSML document is created with SSML elements or tags. These elements are used to adjust the voice, style, pitch, prosody, volume, and more.

Here's a subset of the basic structure and syntax of an SSML document:

XML

```
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
xmlns:mstts="https://www.w3.org/2001/mstts" xml:lang="string">
  <mstts:backgroundaudio src="string" volume="string" fadein="string"
fadeout="string"/>
  <voice name="string" effect="string">
    <audio src="string"></audio>
    <bookmark mark="string"/>
    <break strength="string" time="string" />
    <emphasis level="value"></emphasis>
    <lang xml:lang="string"></lang>
    <lexicon uri="string"/>
    <math xmlns="http://www.w3.org/1998/Math/MathML"></math>
    <mstts:audioduration value="string"/>
    <mstts:ttseembedding speakerProfileId="string"></mstts:ttseembedding>
    <mstts:express-as style="string" styledegree="value" role="string">
</mstts:express-as>
    <mstts:silence type="string" value="string"/>
    <mstts:viseme type="string"/>
```

```

    <p></p>
    <phoneme alphabet="string" ph="string"></phoneme>
    <prosody pitch="value" contour="value" range="value" rate="value"
volume="value"></prosody>
    <s></s>
    <say-as interpret-as="string" format="string" detail="string"></say-as>
    <sub alias="string"></sub>
  </voice>
</speak>

```

Some examples of contents that are allowed in each element are described in the following list:

- **audio**: The body of the `audio` element can contain plain text or SSML markup that's spoken if the audio file is unavailable or unplayable. The `audio` element can also contain text and the following elements: `audio`, `break`, `p`, `s`, `phoneme`, `prosody`, `say-as`, and `sub`.
- **bookmark**: This element can't contain text or any other elements.
- **break**: This element can't contain text or any other elements.
- **emphasis**: This element can contain text and the following elements: `audio`, `break`, `emphasis`, `lang`, `phoneme`, `prosody`, `say-as`, and `sub`.
- **lang**: This element can contain all other elements except `mstts:backgroundaudio`, `voice`, and `speak`.
- **lexicon**: This element can't contain text or any other elements.
- **math**: This element can only contain text and MathML elements.
- **mstts:audioduration**: This element can't contain text or any other elements.
- **mstts:backgroundaudio**: This element can't contain text or any other elements.
- **mstts:embedding**: This element can contain text and the following elements: `audio`, `break`, `emphasis`, `lang`, `phoneme`, `prosody`, `say-as`, and `sub`.
- **mstts:express-as**: This element can contain text and the following elements: `audio`, `break`, `emphasis`, `lang`, `phoneme`, `prosody`, `say-as`, and `sub`.
- **mstts:silence**: This element can't contain text or any other elements.
- **mstts:viseme**: This element can't contain text or any other elements.
- **p**: This element can contain text and the following elements: `audio`, `break`, `phoneme`, `prosody`, `say-as`, `sub`, `mstts:express-as`, and `s`.
- **phoneme**: This element can only contain text and no other elements.
- **prosody**: This element can contain text and the following elements: `audio`, `break`, `p`, `phoneme`, `prosody`, `say-as`, `sub`, and `s`.

- `s`: This element can contain text and the following elements: `audio`, `break`, `phoneme`, `prosody`, `say-as`, `mstts:express-as`, and `sub`.
- `say-as`: This element can only contain text and no other elements.
- `sub`: This element can only contain text and no other elements.
- `speak`: The root element of an SSML document. This element can contain the following elements: `mstts:backgroundaudio` and `voice`.
- `voice`: This element can contain all other elements except `mstts:backgroundaudio` and `speak`.

The Speech service automatically handles punctuation as appropriate, such as pausing after a period, or using the correct intonation when a sentence ends with a question mark.

Special characters

To use the characters `&`, `<`, and `>` within the SSML element's value or text, you must use the entity format. Specifically you must use `&` in place of `&`, use `<` in place of `<`, and use `>` in place of `>`. Otherwise the SSML isn't parsed correctly.

For example, specify `green & yellow` instead of `green & yellow`. The following SSML is parsed as expected:

XML

```
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis" xml:lang="en-US">
  <voice name="en-US-AvaNeural">
    My favorite colors are green &amp; yellow.
  </voice>
</speak>
```

Special characters such as quotation marks, apostrophes, and brackets, must be escaped. For more information, see [Extensible Markup Language \(XML\) 1.0: Appendix D](#).

Double or single quotation marks must enclose the attribute values. For example, `<prosody volume="90">` and `<prosody volume='90'>` are well-formed, valid elements, but `<prosody volume=90>` isn't recognized.


Speak root element

The `speak` element contains information such as version, language, and the markup vocabulary definition. The `speak` element is the root element that's required for all SSML documents. You must specify the default language within the `speak` element, whether or not the language is adjusted elsewhere such as within the [lang](#) element.

Here's the syntax for the `speak` element:

XML

```
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
xml:lang="string"></speak>
```

 Expand table

| Attribute | Description | Required or optional |
|-----------|--|----------------------|
| version | Indicates the version of the SSML specification used to interpret the document markup. The current version is "1.0". | Required |
| xml:lang | The language of the root document. The value can contain a language code such as <code>en</code> (English), or a locale such as <code>en-US</code> (English - United States). | Required |
| xmlns | The URI to the document that defines the markup vocabulary (the element types and attribute names) of the SSML document. The current URI is "http://www.w3.org/2001/10/synthesis". | Required |

The `speak` element must contain at least one [voice element](#).

speak examples

The supported values for attributes of the `speak` element were [described previously](#).

Single voice example

This example uses the `en-US-AvaNeura1` voice. For more examples, see [voice examples](#).

XML

```
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis" xml:lang="en-US">
```

```
<voice name="en-US-AvaNeural">
  This is the text that is spoken.
</voice>
</speak>
```

Add a break

Use the `break` element to override the default behavior of breaks or pauses between words. Otherwise the Speech service automatically inserts pauses.

Usage of the `break` element's attributes are described in the following table.

 Expand table

| Attribute | Description | Required or optional |
|-----------|--|----------------------|
| strength | The relative duration of a pause by using one of the following values: <ul style="list-style-type: none">x-weakweakmedium (default)strongx-strong | Optional |
| time | The absolute duration of a pause in seconds (such as <code>2s</code>) or milliseconds (such as <code>500ms</code>). Valid values range from 0 to 20000 milliseconds. If you set a value greater than the supported maximum, the service uses <code>20000ms</code> . If the <code>time</code> attribute is set, the <code>strength</code> attribute is ignored. | Optional |

Here are more details about the `strength` attribute.

 Expand table

| Strength | Relative duration |
|----------|-------------------|
| X-weak | 250 ms |
| Weak | 500 ms |
| Medium | 750 ms |
| Strong | 1,000 ms |

| Strength | Relative duration |
|----------|-------------------|
| X-strong | 1,250 ms |

Break examples

The supported values for attributes of the `break` element were [described previously](#). The following three ways all add 750 ms breaks.

XML

```
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis" xml:lang="en-US">
  <voice name="en-US-AvaNeural">
    Welcome <break /> to text to speech.
    Welcome <break strength="medium" /> to text to speech.
    Welcome <break time="750ms" /> to text to speech.
  </voice>
</speak>
```

Add silence

Use the `mstts:silence` element to insert pauses before or after text, or between two adjacent sentences.

One of the differences between `mstts:silence` and `break` is that a `break` element can be inserted anywhere in the text. Silence only works at the beginning or end of input text or at the boundary of two adjacent sentences.

The silence setting is applied to all input text within its enclosing `voice` element. To reset or change the silence setting again, you must use a new `voice` element with either the same voice or a different voice.

Usage of the `mstts:silence` element's attributes are described in the following table.

[Expand table](#)

| Attribute | Description | Required or optional |
|-----------|--|----------------------|
| type | <p>Specifies where and how to add silence. The following silence types are supported:</p> <ul style="list-style-type: none"> Leading – Extra silence at the beginning of the text. The value that you set is added to the natural silence before the start of text. Leading-exact – Silence at the beginning of the text. The value is an absolute silence length. Tailing – Extra silence at the end of text. The value that you set is added to the natural silence after the last word. Tailing-exact – Silence at the end of the text. The value is an absolute silence length. Sentenceboundary – Extra silence between adjacent sentences. The actual silence length for this type includes the natural silence after the last word in the previous sentence, the value you set for this type, and the natural silence before the starting word in the next sentence. Sentenceboundary-exact – Silence between adjacent sentences. The value is an absolute silence length. Comma-exact – Silence at the comma in half-width or full-width format. The value is an absolute silence length. Semicolon-exact – Silence at the semicolon in half-width or full-width format. The value is an absolute silence length. Enumerationcomma-exact – Silence at the enumeration comma in full-width format. The value is an absolute silence length. <p>An absolute silence type (with the <code>-exact</code> suffix) replaces any otherwise natural leading or trailing silence. Absolute silence types take precedence over the corresponding non-absolute type. For example, if you set both <code>Leading</code> and <code>Leading-exact</code> types, the <code>Leading-exact</code> type takes effect. The WordBoundary event takes precedence over punctuation-related silence settings including <code>Comma-exact</code>, <code>Semicolon-exact</code>, or <code>Enumerationcomma-exact</code>. When you use both the <code>WordBoundary</code> event and punctuation-related silence settings, the punctuation-related silence settings don't take effect.</p> | Required |
| value | <p>The duration of a pause in seconds (such as <code>2s</code>) or milliseconds (such as <code>500ms</code>). Valid values range from 0 to 20000 milliseconds. If you set a value greater than the supported maximum, the service uses <code>20000ms</code>.</p> | Required |

mstts silence examples

The supported values for attributes of the `mstts:silence` element were [described previously](#).

In this example, `mstts:silence` is used to add 200 ms of silence between two sentences.

XML

```
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
xmlns:mstts="http://www.w3.org/2001/mstts" xml:lang="en-US">
<voice name="en-US-AvaNeural">
<mstts:silence type="Sentenceboundary" value="200ms"/>
If we're home schooling, the best we can do is roll with what each day brings
and try to have fun along the way.
A good place to start is by trying out the slew of educational apps that are
helping children stay happy and smash their schooling at the same time.
</voice>
</speak>
```

In this example, `mstts:silence` is used to add 50 ms of silence at the comma, 100 ms of silence at the semicolon, and 150 ms of silence at the enumeration comma.

XML

```
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
xmlns:mstts="http://www.w3.org/2001/mstts" xml:lang="zh-CN">
<voice name="zh-CN-YunxiNeural">
<mstts:silence type="comma-exact" value="50ms"/><mstts:silence type="semicolon-
exact" value="100ms"/><mstts:silence type="enumerationcomma-exact"
value="150ms"/>你好呀，云希、晓晓；你好呀。
</voice>
</speak>
```

Specify paragraphs and sentences

The `p` and `s` elements are used to denote paragraphs and sentences, respectively. In the absence of these elements, the Speech service automatically determines the structure of the SSML document.

Paragraph and sentence examples

The following example defines two paragraphs that each contain sentences. In the second paragraph, the Speech service automatically determines the sentence structure, since they

aren't defined in the SSML document.

XML

```
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis" xml:lang="en-US">
  <voice name="en-US-AvaNeural">
    <p>
      <s>Introducing the sentence element.</s>
      <s>Used to mark individual sentences.</s>
    </p>
    <p>
      Another simple paragraph.
      Sentence structure in this paragraph is not explicitly marked.
    </p>
  </voice>
</speak>
```

Bookmark element

You can use the `bookmark` element in SSML to reference a specific location in the text or tag sequence. Then you use the Speech SDK and subscribe to the `BookmarkReached` event to get the offset of each marker in the audio stream. The `bookmark` element isn't spoken. For more information, see [Subscribe to synthesizer events](#).

Usage of the `bookmark` element's attributes are described in the following table.

[Expand table](#)

| Attribute | Description | Required or optional |
|-----------|--|----------------------|
| mark | The reference text of the <code>bookmark</code> element. | Required |

Bookmark examples

The supported values for attributes of the `bookmark` element were [described previously](#).

As an example, you might want to know the time offset of each flower word in the following snippet:

XML

```
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis" xml:lang="en-US">
  <voice name="en-US-AvaNeural">
    We are selling <bookmark mark='flower_1' />roses and <bookmark
mark='flower_2' />daisies.
  </voice>
</speak>
```

Viseme element

A viseme is the visual description of a phoneme in spoken language. It defines the position of the face and mouth while a person is speaking. You can use the `mstts:viseme` element in SSML to request viseme output. For more information, see [Get facial position with viseme](#).

The viseme setting is applied to all input text within its enclosing `voice` element. To reset or change the viseme setting again, you must use a new `voice` element with either the same voice or a different voice.

Usage of the `viseme` element's attributes are described in the following table.

[Expand table](#)

| Attribute | Description | Required or optional |
|-----------|---|----------------------|
| type | The type of viseme output. <ul style="list-style-type: none"><code>redlips_front</code> – lip-sync with viseme ID and audio offset output<code>FacialExpression</code> – blend shapes output | Required |

ⓘ Note

Currently, `redlips_front` only supports neural voices in `en-US` locale, and `FacialExpression` supports neural voices in `en-US` and `zh-CN` locales.

Viseme examples

The supported values for attributes of the `viseme` element were [described previously](#).

This SSML snippet illustrates how to request blend shapes with your synthesized speech.

XML

```
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
xmlns:mstts="http://www.w3.org/2001/mstts" xml:lang="en-US">
  <voice name="en-US-AvaNeural">
    <mstts:viseme type="FacialExpression"/>
    Rainbow has seven colors: Red, orange, yellow, green, blue, indigo, and vi-
    olet.
  </voice>
</speak>
```

Next steps

- [SSML overview](#)
- [Voice and sound with SSML](#)
- [Language support: Voices, locales, languages](#)

Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) | [Get help at Microsoft Q&A](#)