info@readspeaker.com
www.readspeaker.com

rSpeak SSML Documentation

rSpeak SSML Documentation 1
Introduction 2
SSML Standards 3
rSpeak SSML support and exceptions 4
Interpretation of Elements and Attributes 4
SSML Elements 6
audio element - for insertion of sound files 6
break element - for insertion of breaks 7
desc element 7
emphasis element 7
lang element 7
lexicon element 8
lookup element 8
mark element 8
meta element 8
metadata element 8
p element 9
phoneme element - for phonetic transcriptions 9
prosody element 10

s element 10
say-as element 11
speak element 11
sub element 12
token element 13
SSML Example **14**

Introduction

SSML Standards

The rSpeak Text-To-Speech supports both SSML 1.0 and SSML 1.1 as defined by the following standards:

Speech Synthesis Markup Language (SSML) Version 1.0, W3C Recommendation 7 September 2004 https://www.w3.org/TR/2004/REC-speech-synthesis-20040907/

Speech Synthesis Markup Language (SSML) Version 1.1, W3C Recommendation 7 September 2010
https://www.w3.org/TR/2010/REC-speech-synthesis11-20100907/

A specific version may be chosen explicitly by setting the version attribute of <speak> document element.

rSpeak SSML support and exceptions

SSML is supported according to the SSML 1.0 and SSML 1.1 specifications above, with some exceptions as described in the document of supported elements below.

Generally, the TTs engine will not require SSML input to strictly adhere to any specific version of SSML, i.e. SSML 1.0 and 1.1 elements and values can be mixed in the same document. In some cases where the specifications are in conflict, the version attribute of the speak element will determine how certain constructs are interpreted.

Note that specifying language in SSML (the xml:lang attribute) is mandatory. The TTS expects one or two part language codes per SSML/BCP47, for example “en” or “en-GB”. Language matching is case insensitive and both dash and underscore are accepted as separators, for example, “en-GB”, “en_gb”, “en_GB” are all accepted and mean the same thing. Which languages can be used depends on which languages are installed and licensed. The TTS will prefer full language matches, but will accept partial matches if no full match is available; i.e., if a document tries to use Australian english (en-AU) but no such voice is available, then it will choose a voice that speaks any variant of English at all, if available.

Interpretation of Elements and Attributes

There are some exceptions, when the rSpeak TTS does not fully support SSML standards. There are also cases when SSML standards do not impose restrictions and leave interpretation decisions to SSML processors. Such cases are described in the following section.

SSML Elements

audio element - for insertion of sound files

With the **audio** element it is possible to insert specific sounds in the speech output.

Notation: `<audio src="file:FILENAME">DESCRIPTION</audio>`

Example: `<audio src="file:laugh">haha</audio>`

An **audio** element with src attribute consisting of **file:** followed by a name without a file suffix (such as *file:laugh*), is used to insert rSpeak's prerecorded paralinguistic sounds (laughter, coughs etc). Lists of available sounds for each voice are found in a separate document.

Depending on current rSpeak settings (see separate documentation for the rSpeak SDK or the product through which the rSpeak engine is used), it may also be permitted to use a **file:** URL that refers to an audio file using an absolute and/or relative path. PCM sound files of .au and .wav formats are supported, (including a-law and μ-law). The PCM file audio will automatically be resampled to match the current sampling rate of the rSpeak engine.

If the named audio resource can not be played for any reason, then any contents of the audio elements (for example “haha” in the example above) will be read instead, as a fallback.

The supported attributes are **speed**, **soundLevel** and **src** (required). Trimming attributes (SSML 1.1) are currently not supported. Other URL types than **file:** are not supported.

break element - for insertion of breaks

The break element inserts a break. Both the **time** and **strength** attributes are supported. If both are used, the time attribute will set the duration of the break.

Values of the **strength** attribute:

- None
- x-weak
- weak
- medium
- strong
- x-strong

Example: <break strength="strong" />

Value of the time attribute can be set either with **s** (seconds) or **ms** (milliseconds).

Example: <break time=”700ms” />

desc element

Ignored

emphasis element

Ignored

lang element

With the lang element it is possible to switch language. Changing the language also changes the voice, if there is a voice available. If the language is not supported the text inside the lang element will be ignored. The attribute onlangfailure is treated as ignoretext.

Example: English, <lang xml:lang=de”>Deutsch</lang>, English.

lexicon element

Ignored

lookup element

Ignored

mark element

The mark element specifies a named event which is triggered by the TTS engine when that location in the text is encountered in the generated audio stream. (What effect this event has is application specific, but it doesn't affect the audion being generated.)

The mark event must have a name attribute. The given name doesn't have any meaning to the TS engine, but is included in the generated event.

Note taht built-in normalization rules might, in some particular contexts such as date and currency expressions, cause adjacent words and numbers to be reordered. The TTS engine will generally try to preserve the association between marks and adjacent words in such cases, meaning that the mark events are not necessarily triggered in the exact order in which they occur in the SSML input but rather in a way that is more true to the reading order.

Example: <mark name="item1"/>First item, <mark name="item2"/>second item.

meta element

Ignored

metadata element

Ignored

p element

A p element represents a paragraph. rSpeak adds a sentence break before and after the element.

phoneme element - for phonetic transcriptions

The phoneme element can be used to correct the pronunciation of a word. Supported phonetic alphabets are “ipa” and “x-rspeak”. “ipa” is default. See separate document list of phonemes installed by each voice package.

Notation: <phoneme alphabet="ALPHABET" ph='TRANSCRIPTION'>WORD</phoneme>

Example: <phoneme alphabet="x-rspeak" ph='"k A: r'>car</phoneme>

Example: <phoneme alphabet="ipa" ph='ˈk ɑː ɻ'>car</phoneme>

prosody element

The prosody element allows control of the pitch, speaking rate and/or volume of the speech output.

Attributes:

- **contour** - ignored
- **duration** - ignored
- **pitch** - **“+/-Xst”** (semitones), **“+/-X%”, “x-low”, “low”, “medium”, “high”, “x-high”** or **“default”**
- **range** - ignored
- **rate** - values **“+/-X%”, “x-slow”, “slow”, “medium”, “fast”, “x-fast”,** or **“default”.**
- **volume** - **“+/-X%”, “+/-X.XdB”, “silent”, “x-soft”, “soft”, “medium”, “loud”, “x-loud”** or **“default”.**

Example: <prosody volume="+10.5dB" rate="x-slow" pitch="-5st">This will be read with increased volume, decreased rate and x-high pitch</prosody>

s element

An s element represents a sentence and rSpeak adds sentence pauses before and after the element. Currently sentence breaks are added inside s elements, in feature versions no sentence breaks will be added.

say-as element

The say-as element together with the **interpret-as** attribute can be used to guide the TTS's normalization when the interpretation needs to be corrected.

Supported interpret-as values:

- **characters** - spells out letters, reads digits one by one, and expands non-alphabetical characters
- **date** - Read digits as date
- **spell-out** - spells out letters
- **cardinal** - reads as a cardinal number
- **ordinal** - reads as an ordinal number
- **digits** - reads digits one by one with natural pauses
- **fraction** - reads as fractions
- **year** - reads as a year
- **telephone** - reads as a telephone number
- **url** - reads as an url
- **unit** - expands abbreviated weights and measures

Example: <say-as interpret-as="spell-out">AMEP</say-as> (will be read as "A M E P")

Example: <say-as interpret-as="ordinal">3</say-as> (will be read as "third")

speak element

The speal element is required as the root element of an SSML document, as per SSML 1.0/1.1.

The attribute version should always be set and must be “1.0” and “1.1”. Generally, the TTS engine will attempt to support SSML 1.1 constructs even in SSML 1.0 documents and vice versa, but the version set here will affect the processing of the document in some cases.

The attribute xml:lang is required and should specify a language supported by an installed voice, for example “en-GB” for British English, or “en” to specify English without specifying the variant of English.

The optional attribute onlangfailure is currently ignored; for the behaviour when encountering unavailable/unknown language codes, see the lang element.

SSML 1.1 trimming attributes (startmark, endmark) are currently not supported.

Example: <speak version="1.1" xml:lang="en-AU"> This is Australian English, and will be spoken by an Australian English voice if available. </speak>

Example: <speak version="1.1" xml:lang="en"> This is an unspecified variant of English, and might be spoken by any English voice. </speak>

sub element

The sub element can be used so that the TTS reads the content in the alias attribute instead of the content in the element.

Example: _{WHO}

token element

The token element can be used to disambiguate heteronyms.

Notation: <token role='rspeaktags:POS-TAG'>WORD</token>

Example: <token role='rspeaktags:NN'>content</token>

The set of available part of speech tags is language specific, but typically these are supported:

- NN - noun
- JJ - adjective
- VB - verb
- AB - adverb
- PM - proper noun (name)

NB! Part of speech tagging is only effective when the word is in the rSpeak's lexicon and has the same part-of-speech tags as in the lexicon. If a heteronym has the same part of speech for both pronunciations they may be numbered, eg. NN2.

SSML Example

This is an SSML document demonstrating a few of the elements. Note that language and voice elements will require that you have those languages installed for the desired effect.

```
<speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
xml:lang="en-au"><voice name="Jack"
gender="male">
<p>
Audio element:
<audio src="file:laugh">hahaha</audio>
</p>
<p>
Break element:
<s>500 ms <break time="500ms" /> pause</s>
<s>x-weak <break strength="x-weak" /> pause</s>
</p>
<p>
Lang and voice element:
Australian English, <lang xml:lang="en_us">American English, male voice <voice
gender="female">and a
female voice</voice></lang>, and Australian English.
</p>
<p>
Phoneme:
X-rspeak transcription <phoneme alphabet="x-rspeak" ph='I g . "z A: m . p @
l'>example</phoneme>
IPA transcription <phoneme alphabet="ipa" ph="ɪ g . ˈz ɑː m . p ə
l">example</phoneme>, variant <phoneme
alphabet="ipa" ph="ɪgˈzɑːmpəl">example</phoneme>
</p>
<p>
Prosody element:
<prosody volume="+10.5dB" rate="x-slow" pitch="high">Increased volume,
decreased rate and high
pitch</prosody>
</p>
<p>
Sub element:
<sub alias="World Health Organization">WHO</sub>
</p>
```

Token element:
<token role='rspeaktags:NN'>content</token> <token
role='rspeaktags:JJ'>content</token>

</voice></speak>