

# Configure Speech Synthesis Markup Language with Google Text to Speech

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Configuration](#)

[Example 1:](#)

[Example 2:](#)

## Introduction

This document describes how to configure and use Speech Synthesis Markup Language (SSML) with Google Text to Speech (Google TTS).

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

- Webex Contact Center (WxCC) 2.0

### Components Used

The information in this document is based on these software versions:

- WxCC 2.0

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

## Background Information

Speech Synthesis Markup Language allows more customization in your Google Text to Speech audio response. It provides details on pauses, and audio format for acronyms, dates, times, abbreviations, or text that can be censored.

## Configuration

Refer to [Google SSML](#) to find information about all SSML elements on the Google Cloud Portal.

Here are several examples of WxCC Flow configuration with some of the SSML elements:

### <speak>

The Speech Synthesis Markup Language is an XML application. The root element is <speak>. All additional elements must be configured inside of the speak tags. If you put plain text within the speak tag, the caller hears the voice in the default TTS setup which is configured in the WxCC Flow.

The image shows a WxCC Flow configuration for a 'PlayMessage1' activity. On the left, a flow diagram shows a 'NewPhoneCo... Start Flow' event leading to a 'PlayMessage1' activity, which then leads to a 'DisconnectCo...' activity. On the right, the configuration panel for 'PlayMessage1' is open. It includes sections for 'General Settings', 'Prompt', and 'Output Voice'. The 'Text-to-Speech Message' field is highlighted with a red box and contains the SSML code: 

```
<1> <speak> Speak is the root element </speak>
```

### <speak-as>

This element lets you indicate information about the type of text construct that is contained within the element. It also helps specify the level of detail provided in the contained text.

The <sayas> element has the required attribute, interpret-as, which determines how the value is spoken. Optional attributes format and detail can be used related to the particular interpret-as value.

## Example 1:

<spek-as> element with interpret-as='currency' as the required attribute and language='en-US' as the optional attribute. In this example the caller hears - Your current balance is Fifty-three dollars and twenty-one cents.

The screenshot shows a workflow editor with a 'NewPhoneCo...' Start Flow activity leading to a 'PlayMessage1' activity. The configuration panel for 'PlayMessage1' is open, showing the following text-to-speech message:

```
<spek>
Your current balance is
1
<say-as interpret-as='currency' language='en-US'>$53.21</say-as>
|
</spek>
```

## Example 2:

<spek-as> element with interpret-as="ordinal" as the required attribute. In this setup, the caller hears his or her position in the queue in the ordinal format such as first, second, and so on.

The screenshot shows a workflow editor with a 'QueueContact1' activity leading to a 'GetQueueInfo1' activity, which then leads to a 'PlayMessage3' activity. The configuration panel for 'PlayMessage3' is open, showing the following text-to-speech message:

```
<spek>
You are the
<say-as interpret-as='ordinal'>[[GetQueueInfo1.P1Q1s]]</say-as>
in the queue. Please stand by and next available
agent will be with you shortly.
</spek>
```