Extracted from:

Build Talking Apps

Develop Voice-First Applications for Alexa

The Pragmatic Bookshelf

Raleigh, North Carolina

The Pragmatic Programmers

Build Talking Apps

Develop Voice-First Applications for Alexa

Craig Walls

edited by Jacquelyn Carter

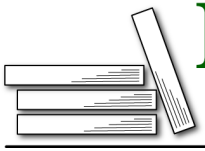
Build Talking Apps

Develop Voice-First Applications for Alexa

Craig Walls

The Pragmatic Bookshelf

Raleigh, North Carolina



For our complete catalog of hands-on, practical, and Pragmatic content for software developers, please visit *https://pragprog.com*.

The team that produced this book includes:

CEO: Dave Rankin
COO: Janet Furlow
Managing Editor: Tammy Coron
Development Editor: Jacquelyn Carter
Copy Editor: Karen Galle
Indexing: Potomac Indexing, LLC
Layout: Gilson Graphics
Founders: Andy Hunt and Dave Thomas

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

ISBN-13: 978-1-68050-725-6
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—April 2022

CHAPTER 6

Embellishing Response Speech

If decades of science fiction have taught us anything, it's that when computers or robots talk, they speak in a mechanical tone and staccato voice. Whether it's the robot from *Lost in Space* or the lovable robot from *Wall-E*, most talking machines in science fiction do not speak in natural human tones.

In the real world, however, Alexa has a much more human voice than her sci-fi counterparts. She speaks almost as naturally as most any person you'll meet (possibly better in some cases). Even so, despite her natural voice, Alexa doesn't often speak with a great deal of variation or emotion. Her tone, although human-like, is at best very matter-of-fact, and at worst might even betray her as a machine.

We want the users of our skills to feel comfortable when they speak with Alexa. The more human she seems the more likely that they will continue to use our skills and continue to rely on Alexa as a relatable assistant that can help them with whatever they need.

In this chapter, we're going to explore the Speech Synthesis Markup Language (SSML), using it to gain control over how Alexa speaks responses. We'll use SSML to change the tone, volume, and pitch of Alexa's voice, change how she pronounces words, and even completely change her voice altogether.

To start, let's have a look at a simple SSML example and see how to test it with Alexa's text-to-speech simulator.

Getting to Know SSML

SSML[1] is an XML-based markup language that describes not only what Alexa should say, but how she should say it. Much the same as how HTML describes

1. https://www.w3.org/TR/speech-synthesis11/

how text should appear on a webpage, SSML describes how text should sound when spoken. We will apply SSML in our skill's responses to make Alexa sound more natural, and expressive, and to add a little pizzazz.

Not All SSMLs Are the Same

As a W3C specification, SSML defines several elements for crafting voice responses for several voice assistants, including Google Assistant, Siri, and, of course, Alexa. But as with many specifications, each SSML implementation is a little different from the others and different from the specification. As such, not all SSML elements defined in the specification will work with Alexa. And Alexa adds a few custom elements that are not part of the specification.

Before we start adding SSML to our skill responses, let's have a look at a simple SSML document on its own, outside of the context of a skill:

```
<speak>
  Hello world!
</speak>
```

As you can see, the root element of any SSML document is the `<speak>` element. But you can leave it out in the text passed to `speak()` and it will be inferred.

Aside from the `<speak>` element, this example doesn't leverage SSML to alter how Alexa speaks. So let's make a small change to alter how Alexa will say "Hello world":

```
<speak>
  <amazon:effect name="whispered">Hello world</amazon:effect>!
</speak>
```

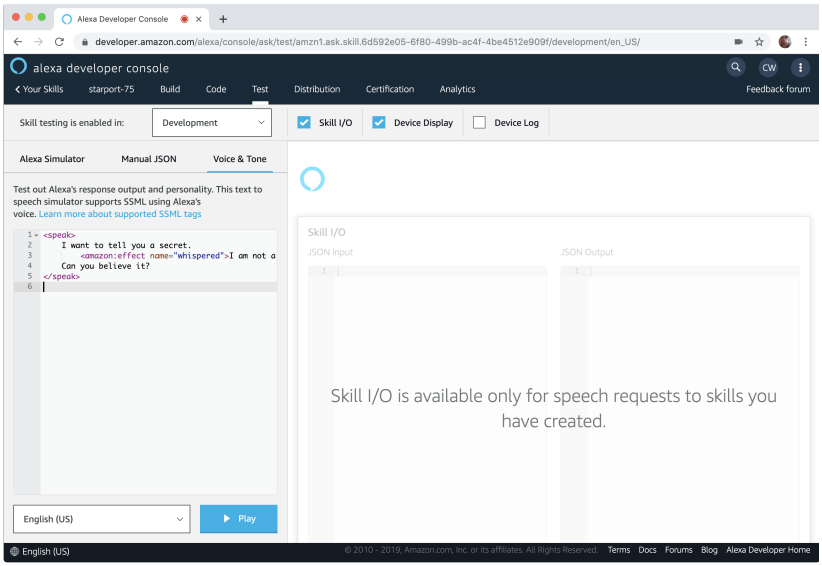
Now, instead of simply saying "Hello world" in her normal voice, Alexa will whisper the greeting, thanks to the `<amazon:effect>` element. But don't just assume that to be true; let's actually *hear* her say it using handy text-to-speech simulator provided in the Alexa Developer Console.

Testing SSML with the Text-to-Speech Simulator

The main subject of this chapter is using SSML to modify how responses returned from a skill will *sound*. Unfortunately, simply reading the words on the pages of this book can't compare with actually *hearing* what effect SSML will have on text. Although it's absolutely possible to write automated tests to assert that a skill's response contains an expected SSML response, there's no way to write a test to verify how it sounds.

The most important tool for testing SSML is your own ears. When Alexa speaks a response, you can deploy and interact with your skill, and listen and decide if it sounds the way you expected it and make adjustments if not. Even so, it'd be even better if there were a way to listen to how Alexa will speak an SSML response before you code it into your intent handler's response.

Fortunately, Amazon has provided a useful (and incredibly fun) tool to listen to snippets of SSML. Under the "Test" tab of the developer console, you may have noticed a sub-tab labeled "Voice & Tone". You can find it by opening any skill in the developer console, clicking on the "Test" tab at the top, and then clicking on "Voice & Tone" in the left-hand panel. You should see something like this:



On the left is Alexa's text-to-speech simulator. It provides a text editor, in which you can write or paste in SSML. Once you're ready to hear the results, make sure your volume is turned up and click the "Play" button at the bottom. Alexa will speak whatever the SSML in the text editor says she should speak.

For example, when you first open the text-to-speech simulator, it is already preloaded with this small bit of SSML:

```
<speak>
    I want to tell you a secret.
        <amazon:effect name="whispered">I am not a real human.
        </amazon:effect>.
    Can you believe it?
</speak>
```

If you leave this unchanged, then Alexa will say, in her normal voice, “I want to tell you a secret.” Then, she will whisper, “I am not a real human.” Finally, she will say, “Can you believe it?” in her normal voice.

Need More Space?

You may find the width of the editor in the text-to-speech simulator to be a bit cramped. Unfortunately, there’s no built-in way to widen it by stretching the left-hand panel.

However, if you are using Chrome, you can install the Alexa Skills Kit Console Chrome Extension[a] and be able to resize the panel by dragging the divider that separates the left panel from the right panel. This extension also lets you save and replay frequently used utterances so that you don’t have to type or say them when testing your skill.

a. https://github.com/jovotech/ask-console-chrome-extension

When it comes to testing SSML, there’s no substitute for actually hearing the results. So you’ll definitely want to have the text-to-speech simulator in reach as you work through the examples in this. Be warned, however: you might find yourself having so much fun tinkering with different SSML incantations in the tool that you’ll lose hours of time.

We’re going to use SSML to add some flair to the Star Port 75 Travel skill’s responses. But before we inject any SSML into our skill’s code, let’s take a tour of some of the ways SSML can change how Alexa speaks. As we work through various SSML examples, use the text-to-speech simulator to try them out. We’ll begin our tour with SSML elements that change the sound of Alexa’s voice.

Changing Alexa’s Voice

Variety is a hallmark of human speech. It’s what keeps us from sounding robotic and canned. When asking a question, we might apply a different inflection than when making a statement. When we’re excited or disappointed about something, we’ll speak with a great deal of emotion, perhaps in a different rate or tone, as compared to our default speech patterns.

Likewise, if Alexa is to be accepted as anything more than a machine, it’s important that her responses exhibit some of the same variety as in human speech, exhibiting variations in tone, rate, volume, excitement, and disappointment. With SSML we can adjust all of these things and even change Alexa’s voice to a completely different voice.

Let's have a look at how to fine-tune how loudly, quickly, and at what tone Alexa speaks.

Adjusting Prosody

In linguistics, prosody is a term used to describe things such as tone, stress, and rhythm of speech. In SSML, prosody can be specified with the <prosody> tag and is specifically concerned with volume, rate, and pitch.

As an example of using the <prosody> tag, let's say you want Alexa to speak a word or phrase at a different volume than normal. The <prosody> tag's volume attribute can help, as shown in the following SSML snippet:

```
<speak>
    Welcome to <prosody volume="x-loud">Star Port 75 Travel</prosody>!
</speak>
```

The volume attribute accepts one of six predefined values, "x-loud", "loud", "medium", "soft", "x-soft", and "silent" (no sound whatsoever). While none of these values will result in a volume that is dramatically different from Alexa's normal volume, they do have a subtle effect on how loud she speaks.

You can also specify a relative volume that is either greater than or less than the current volume:

```
<speak>
    Welcome to <prosody volume="+6dB">Star Port 75 Travel</prosody>!
</speak>
```

In this case, the value "+6dB" is about twice the volume of the current volume. Similarly "-6dB" would be approximately half the volume of the current volume. Take care; volumes much greater than "+6dB" relative to the default volume might result in distortion in the response.

The <prosody> tag can also alter the rate at which Alexa speaks one or more words. For example, consider the following use of <prosody> to have her speak slower than normal:

```
<speak>
    Welcome to <prosody rate="x-slow">Star Port 75</prosody> Travel!
</speak>
```

If you were to paste this into the text-to-speech simulator, Alexa would speak most of the sentence at a normal rate, but would slow down significantly when saying, "Star Port 75."

Like the volume attribute, the rate attribute accepts a handful of predefined values, including "x-slow", "slow", "medium", "fast", and "x-fast". But you can

also specify a relative value as a percentage of the current rate, where “100%” is equal to the current rate. For example, to have her speak “Star Port 75” twice as fast as normal, set the rate attribute to “200%”:

```
<speak>
    Welcome to <prosody rate="200%">Star Port 75</prosody> Travel!
</speak>
```

On the other hand, if you want Alexa to slow down significantly when saying, “Star Port 75,” you can set the rate to “20%”, which is the minimum allowed value:

```
<speak>
    Welcome to <prosody rate="20%">Star Port 75</prosody> Travel!
</speak>
```

One other attribute of prosody that can be controlled is pitch. Pitch specifies how high or low Alexa’s voice is when she speaks. For example, if you want her to speak in a rather low tone, you can set the pitch attribute to “x-low”:

```
<speak>
    Welcome to <prosody pitch="x-low">Star Port 75</prosody> Travel!
</speak>
```

In addition to predefined pitch values—“x-low”, “low”, “medium”, “high”, and “x-high”—you can specify a relative pitch as a positive or negative percentage (where “+0%” is normal pitch):

```
<speak>
    Welcome to <prosody pitch="-33.3%">Star Port 75</prosody> Travel!
</speak>
```

In this case, setting pitch to “-33.3%” is the minimum allowed value and is equivalent to setting it to “x-low”. Similarly, “+50%” is the maximum allowed value and is the same as setting pitch to “x-high”.

You are welcome to mix and match volume, rate, and pitch as you see fit for some interesting effects. The following snippet of SSML, for instance, has Alexa saying “Star Port 75” in an amusingly low, slow, and loud voice:

```
<speak>
  Welcome to
  <prosody pitch="-33.3%"
           rate="20%"
           volume="x-loud">Star Port 75</prosody> Travel!
</speak>
```

You'll definitely want to paste this example into the text-to-speech simulator and give it a try. It should give you some idea of how to alter Alexa's voice to make her sound as if she may have had too much to drink.

While the <prosody> tag gives you near complete control over volume, rate, and pitch, these attributes are often used in combination to apply emphasis to a word or phrase. For simplicity's sake, the <emphasis> tag can be used to control rate and volume in a simpler way when emphasis is the desired outcome:

```
<speak>
  Welcome to
  <emphasis level="strong">Star Port 75</emphasis> Travel!
</speak>
```

Here, a strong emphasis is applied, resulting in a slower rate and increased volume, much as a parent might speak to a child when they're in trouble. On the other hand, setting level to "reduced" will have Alexa speak quicker and at a lower volume, much like a teenager might speak when telling their parent that they've wrecked the family car.

While prosody can add some dramatic effects to the words that Alexa speaks, it has its limits. Meanwhile, language is filled with brief words or phrases that express a great deal of emotion beyond what prosody can handle. Let's see how Alexa can speak with excitement and disappointment with interjections.