Pulse-Code Modulation: Windows WAV or Mac AIFF PCM Codecs

Finally, almost all operating systems, including Windows, Mac OS, and Linux-based ones, such as Android, Tizen, Ubuntu, openSUSE, Blackberry, Firefox OS, Opera OS, and Chrome OS, support the **PCM** (pulse-code modulation) codecs, commonly known as the Windows WAVE (WAV) audio format or the Apple AIFF (Audio Interchange File Format). Most of you are familiar with this lossless digital audio format from one of these two popular operating systems. It is lossless because there is zero compression applied. PCM audio is commonly used for CD-ROM content, as well as telephony applications. This is because PCM Wave audio is an uncompressed digital audio format. It has no CPU-intensive compression algorithms applied to the data stream, and thus decoding (CPU overhead) is not an issue for telephony equipment or for CD players.

For this reason, when we start compressing digital audio assets into various file formats in Chapter 12, which covers digital audio data footprint optimization, you will use PCM as the “baseline” file format.

You probably won’t put PCM into Kindle (MOBI), Java (JAR), or Android (APK) distributable files, however, because there are other formats, such as FLAC and MPEG-4 AAC, which give you the same quality, and do it using an order of magnitude less data.

Ultimately, the only way to find out which audio formats supported by Android have the best digital audio result for any given audio data is to actually encode digital audio in all the primary codecs that you know are well supported and efficient. I show you how this is accomplished in Chapter 12.

Summary

In this chapter, you looked at the digital audio data encoding concepts, principles, and file formats used to compress and decompress digital audio assets, as well as to publish and distribute to end users. You also learned how sample resolution, sample frequency, bit rate, streaming, and HD audio can contribute to your digital audio sample’s quality and to its data footprint.

In the next chapter, you learn about **digital audio data footprint optimization** concepts, terms, and principles.

CHAPTER 5

The Cleanup of Digital Audio: Noise Removal

Now that you know about some of the different areas of digital audio, including MIDI, synthesis, digital audio recording/sampling, digital audio streaming, captive digital audio, and digital audio data file formats, the next step in the sampling process is to remove those **background noise** "artifacts" from the data sample that you created in Chapter 3. This is especially important when working with voice-overs and similar vocal-oriented samples, because you want it to be relatively quiet—except when the person is speaking, of course.

In this chapter, you learn about the noise removal algorithms in Audacity, which were recoded and taken to a new professional level in version 2.1. Noise reduction is a timely topic for Audacity 2 users.

Noise Removal: Algorithmic Processing

In this chapter, you delve into the process of **background noise removal**, which is the first step you generally take after sampling something. You'll need extra waveform noise as an input to the noise reduction (noise removal) algorithm; this is something that you want to do before you trim the unused portions of your sample. You learn about trimming in the next chapter.

Background noise can come from a significant number of different sources, which is why professional recording studios have rooms with sound-absorbing foam on the walls and that use extremely expensive microphones on stands behind circular pop-screens that prevent vocal pops, like you have at the end of the word "fundamentals" in the current sample. The noise in my sample was recorded on a consumer-quality microphone.

Noise Reduction: Defining the Background Noise

If you play the `CH3.aud` digital audio sample created in Chapter 3, and you listen carefully, you can hear background noise.

This is especially evident at the beginning and the end of the data sample. If you do not have Audacity open, launch it and open the **CH3.aud** project file now, so that you can learn

---

Electronic supplementary material The online version of this chapter (doi:10.1007/978-1-4842-1648-4_5) contains supplementary material, which is available to authorized users.

W. Jackson, *Digital Audio Editing Fundamentals*, DOI 10.1007/978-1-4842-1648-4_5

how to fix this problem using the Audacity **noise reduction algorithm**. It's unprofessional to have background noise in the digital audio assets in any of your applications.

Now you begin to get experience in using your Audacity **Effect** menu to apply digital audio sample processing effects; in this case, to remove background noise.

The work process for removing noise in Audacity—perfect for use with voice-over tracks—involves using the vertical bar tool to select a portion of the audio sample that only has noise in it. The easiest way to do this with surgical precision starts with the **Zoom tool**. It is accessed by selecting the **magnifying glass icon**, which is next to the record button. Click the vocal waveform one or two times to zoom into the sample data so that you can see the voice data in the waveform, as shown in Figure 5-1.

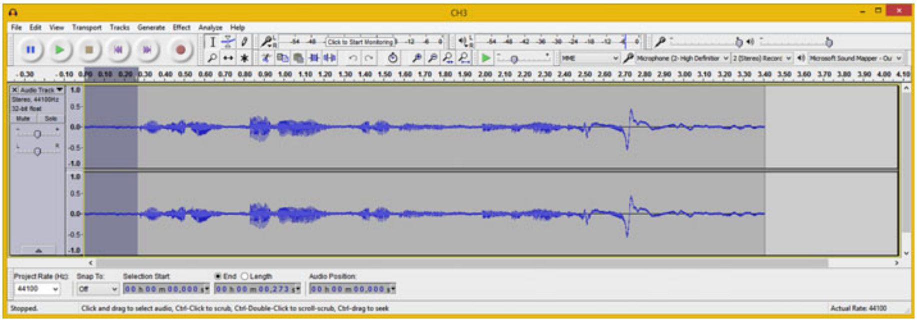

Figure 5-1. *Select an area with noise, using the vertical bar icon*

The noise reduction algorithm uses the noise data that you select; the most logical sample is the data before the first word starts. The noise reduction algorithm can use it to develop algorithmic noise data "profiles" and then remove similar noise across your entire data sample.

After you define noise in the algorithm, the next step processes the entire waveform and removes similar noise data from your audio sample.

As you can see in Figure 5-1, I've zoomed in and selected a section of the sample data that has noise in it, using the vertical bar (waveform selection) tool, which is located right next to the record button.

Be careful not to select any portion of your sound wave that contains vocal content.

Next, in the **Effect** menu, select the **Noise Reduction** algorithm option, as shown in Figure 5-2.

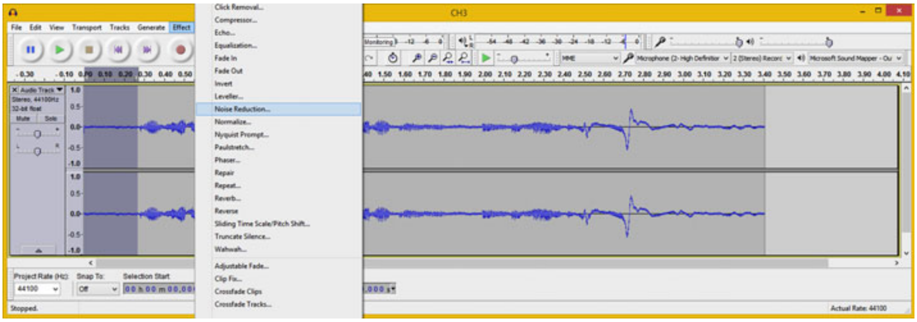

Figure 5-2. *Select the Effect menu and Noise Reduction submenu*

This opens the **Noise Reduction** dialog (see Figure 5-3), where you click the **Get Noise Profile** button. I selected only a quarter of a second of noise data, even though Step 1 of the dialog advises you to select a few seconds of isolated noise data, which may not be possible.

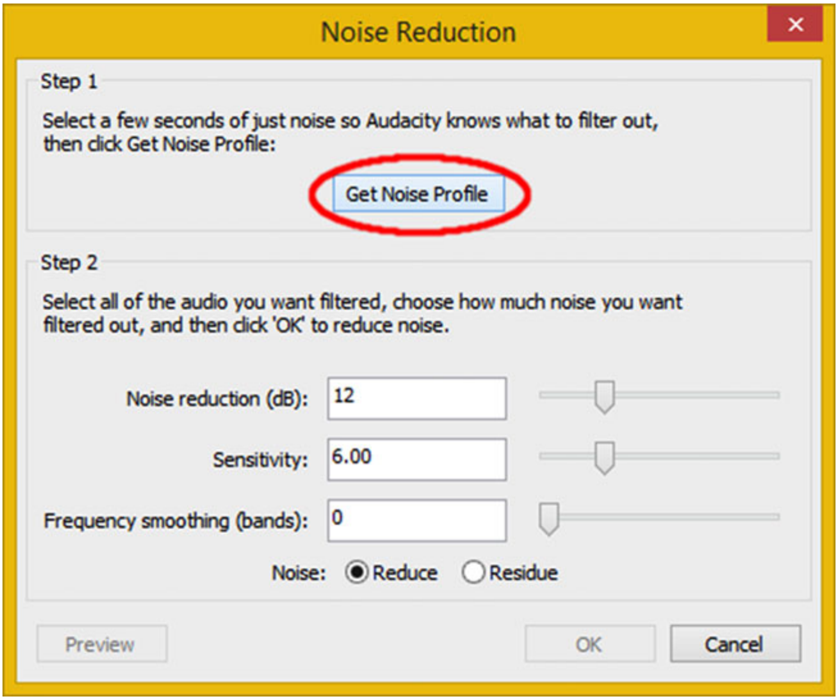

Figure 5-3. *Click the Get Noise Profile button then click on OK*