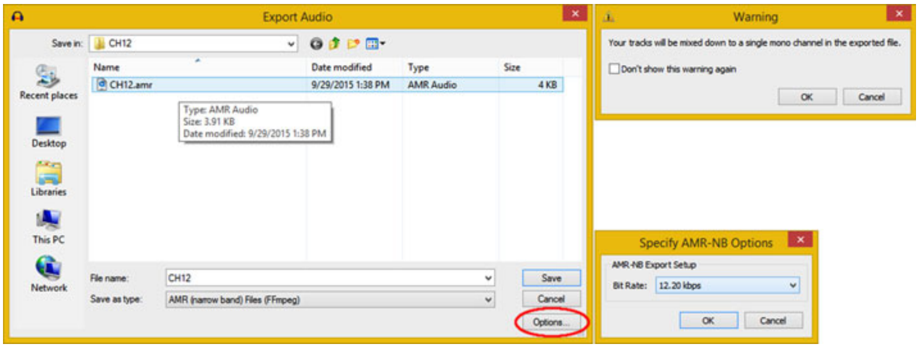
Save this file in your CH12 directory, or whatever your digital audio assets folder is named, and then click the **Options** button to open the **Specify AMR-NB Options** dialog (see Figure 12-8).



Save in: CH12

Name	Date modified	Type	Size
CH12.amr	9/29/2015 1:38 PM	AMR Audio	4 KB

Type: AMR Audio
Size: 3.91 KB
Date modified: 9/29/2015 1:38 PM

File name: CH12
Save as type: AMR (narrow band) Files (FFmpeg)

Your tracks will be mixed down to a single mono channel in the exported file.
☐ Don't show this warning again

AMR-NB Export Setup
Bit Rate: 12.20 kbps

Figure 12-8. *Export Audio in AMR format; set Bit Rate to 12.20*

I chose to use the 12.20 kbps **Bit Rate** setting to initially get the maximum quality results possible with this codec. Later I can try other settings for comparison purposes and put together a results table to see which is best.

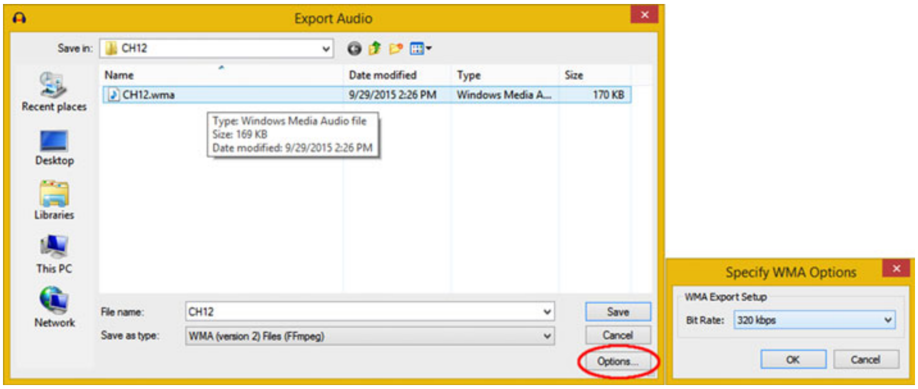
Click **Save** and you should get a **Warning** dialog (see Figure 12-8) informing you that your Stereo Track will be converted to a Mono Track. As you can see on the left of Figure 12-8, this is the smallest data footprint that you've obtained thus far, using only 4 KB of data.

Interestingly, when you play this AMR audio data sample, it still sounds a lot like the audio contained inside the other supported codec formats that you have generated thus far. The CH12.amr file size is 4 KB.

This represents more than a 99% data footprint reduction, some of which is from making a stereo data sample monoaural. To figure this out, 4 KB ÷ 428 KB = 0.00935. This is 0.94% of an original uncompressed file size; 100% – 1% = 99% file size reduction. If you invert 0.00934579, you get 107, which represents a 10,700% reduction in data footprint.

Exporting Audio for Windows: WMA Audio Format

Just in case you are delivering content on Windows OS, let's cover **WMA** or **Windows Media Audio**, which is supported by the Windows Media Player. The last three formats that you are exporting in this chapter use the FFMPEG library that you installed in Chapter 1. The WMA format also uses the library, shown in a drop-down selector using an **(FFmpeg)** denotation. Follow a **File ➤ Export Audio** work process to invoke the Export Audio dialog and select **WMA (Version 2) Files (FFmpeg)** from the **Save as type** drop-down menu. As usual, name this file **CH12**, which is named CH12.wma. Save it into your audio assets folder (mine is a CH12 directory), and then click the **Options** button to open the **Specify WMA Options** dialog (see Figure 12-9). I chose to initially set my **Bit Rate** setting at the highest, 320 kbps. Click **OK** to set **Quality**. Click **Save** to export your CH12.wma file.



Save in: CH12

Name	Date modified	Type	Size
CH12.wma	9/29/2015 2:26 PM	Windows Media A...	170 KB

Type: Windows Media Audio file
Size: 169 KB
Date modified: 9/29/2015 2:26 PM

File name: CH12
Save as type: WMA (version 2) Files (FFmpeg)

WMA Export Setup
Bit Rate: 320 kbps

Figure 12-9. *Export Audio in WMA format, at Quality of 320 kbps*

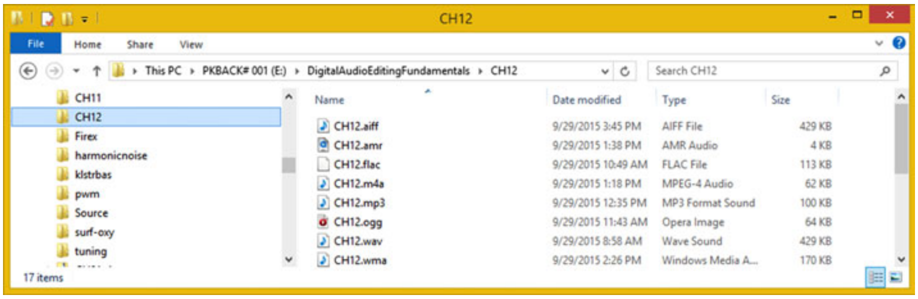
The CH12.wma file size is 170 KB, representing 40% of the uncompressed data footprint and giving you a 60% data reduction.

To figure this out, 170 ÷ 428 = 0.3972, or 40% of your original, uncompressed file size; 100% – 40% = 60% file size reduction. Invert the 0.39719626 and you get 2.517647, or a not-so-impressive 252% data footprint reduction.

Now that you've seen that your WMA file size has the least impressive data footprint reduction thus far, let's fully compare all eight of the digital audio file formats that you generated. For my OS, it is the Windows Explorer file manager.

The Audio Codec Results: Spanning 4 KB to 400 KB

As you can see in Figure 12-10, I opened my CH12 folder in an operating system file management utility to get a comparison viewpoint of all of these digital audio files in one location.



This PC › PKBACK# 001 (E:) › DigitalAudioEditingFundamentals › CH12

Name	Date modified	Type	Size
CH12.aiff	9/29/2015 3:45 PM	AIFF File	429 KB
CH12.amr	9/29/2015 1:38 PM	AMR Audio	4 KB
CH12.flac	9/29/2015 10:49 AM	FLAC File	113 KB
CH12.m4a	9/29/2015 1:18 PM	MPEG-4 Audio	62 KB
CH12.mp3	9/29/2015 12:35 PM	MP3 Format Sound	100 KB
CH12.ogg	9/29/2015 11:43 AM	Opera Image	64 KB
CH12.wav	9/29/2015 8:58 AM	Wave Sound	429 KB
CH12.wma	9/29/2015 2:26 PM	Windows Media A...	170 KB

Figure 12-10. *Eight different exported digital audio formats*

The AMR-NB codec, which is optimized for voice and uses a Mono Track, is the clear winner at 4 KB—more than one hundred times less data used to reproduce the voice-over.

For Stereo Track usage, MPEG-4 AAC and the open sourced Ogg Vorbis codec virtually tie for second place at around 64 KB. They both have widespread support in Kindle, Android, and HTML5, just as AMR-NB does, and are appropriate for use in other types of digital audio, such as music and sound effects.

Tied for third place at around 100 KB each are the MP3, which is lossy, and the FLAC, which is lossless. Between the two, I would opt for using FLAC, although the playback support for MP3 is more widespread.

Finally, in last place is Windows Media Audio (WMA) with a 170 KB data footprint. You can use for Windows Media Player, but be advised that these other digital audio data formats also work in Windows applications. I'd opt for MPEG-4 or Ogg Vorbis.

I am not including the baseline 428 KB AIFF (Mac OS X), or WAV (Windows) digital audio formats in this analysis, but if a storage data footprint isn't an issue, they can be used with results as impressive as FLAC, as they give you a 100% quality reproduction of your digital audio sample data.

Summary

In this chapter, you looked at digital audio data footprint optimization concepts, principles, and techniques regarding eight primary digital audio formats, supported across all four of the key open new media content publishing platforms (HTML5, Android, Kindle, Java) that can decompress your digital audio assets created with Audacity 2.1.1.

You looked at how you should match the specifications for your digital audio assets to the hardware capabilities of your target hardware devices. For high-quality audio, this means using 24-bit 96 kHz audio for HD Audio devices or 16-bit 48 kHz audio to cover all mainstream audio hardware devices.

You learned how to use Audacity to export audio into six of the most widely utilized digital audio codec formats used in HTML5, Kindle, Java, and Android applications development today.

You learned how to calculate a data footprint reduction percentage, and applied this to the six different data formats to see a range of reductions from 60% to more than 99%, spanning FLAC, OGG, MP3, MPEG4, WMA, and AMR digital audio codecs.

In the next chapter, you learn about programming code to implement digital audio inside some of the most popular open source publishing platforms, including HTML5, Kindle, Android Studio, Java, and JavaFX.

CHAPTER 13

The Interactivity of Digital Audio: Programming

Now that you have learned how to create professional digital audio assets using the powerful features in Audacity 2.1 and you know how to export these assets to the most widely used audio file formats in popular content publishing platforms, it's time to take a look at application programming platforms. I am covering this in its own chapter in case you want to take your digital audio compositing career to the next level. In this chapter you will learn about the internal programming language used for Audacity, called **Nyquist**, as well as external programming languages that support digital audio, such as C# (.NET), Objective C (iOS), Java (Android Studio, Linux), JavaFX, JavaScript, HTML5, and CSS3 (WebKit browsers, and HTML5 operating systems).

This is important information to know if you plan to use digital audio samples in programming projects or in open software development platforms, or if you have any interest in learning more about adding programming.

The platforms run a majority of the consumer electronics devices, and include Java (Android Studio and WebKit), JavaFX (Android, iOS, Windows, Linux, Mac OS X, Solaris) and JavaScript with CSS3 and HTML5 scripting (WebKit browsers).

This chapter is not going to teach you programming, for that would take an number of books (and coding experience), but it will expose you to what's possible if you extend the journey you are on from digital audio compositing to new media software development. Everything covered in the chapter is free for commercial use. You can go and download Android Studio (IntelliJ), Java and JavaFX (NetBeans), HTML5 (NetBeans), and Nyquist (NyquistIDE).

Let's start with Nyquist, the programming language for Audacity, and then cover open platform languages such as Java (Android and Kindle), JavaScript (HTML5), and the JavaFX new media engine, which is now a part of Java 7, 8, and 9.

Audacity's Scripting Language: Nyquist 3

As you have seen already, Audacity 2.1 supports an internal scripting language for automating digital audio work processes. A scripting language is traditionally referred to in the computer industry as a **batch processing language**. This term comes from the old mainframe days

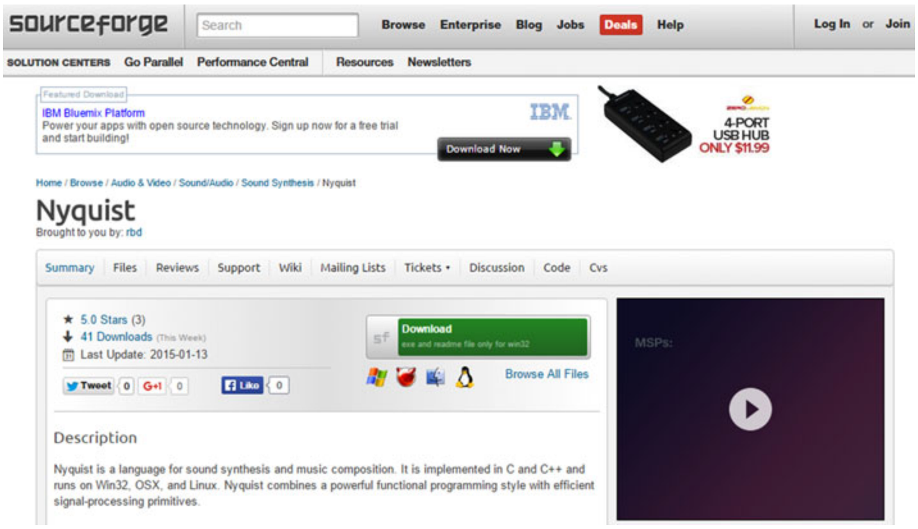
W. Jackson, *Digital Audio Editing Fundamentals*, DOI 10.1007/978-1-4842-1648-4_13

when data was input during the day by employees, and then “batch processed” at night by computers while the employees got some sleep. I’ll first cover how to install Nyquist and then we’ll look at other languages that support digital audio compositing.

Downloading Nyquist: SourceForge.net Repository

You can download the NyquistIDE at SourceForge’s open source software repository at `http://sourceforge.net/projects/nyquist/`.

Figure 13-1 shows Nyquist’s project page on SourceForge.


sourceforge
Search
Browse Enterprise Blog Jobs Deals Help
Log In or Join
SOLUTION CENTERS Go Parallel Performance Central Resources Newsletters
Featured Download
IBM Bluemix Platform
IBM
Download Now
4-PORT USB HUB ONLY $11.99
Home / Browse / Audio & Video / Sound/Audio / Sound Synthesis / Nyquist
Summary Files Reviews Support Wiki Mailing Lists Tickets Discussion Code Cvs
Tweet 0 G+1 0 Like 0
Download
exe and readme file only for win32
Browse All Files
MSPs:

Power your apps with open source technology. Sign up now for a free trial and start building!

Nyquist
Brought to you by: rbd

5.0 Stars (3)
41 Downloads (This Week)
Last Update: 2015-01-13

Description

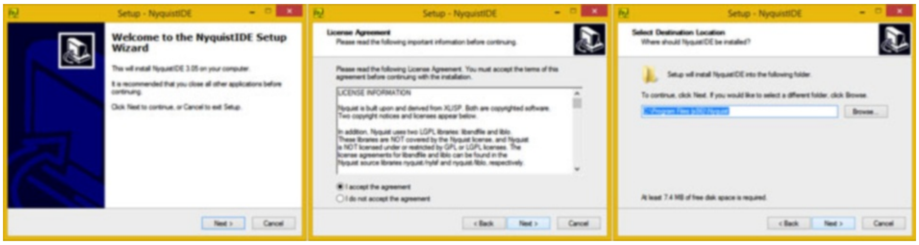
Nyquist is a language for sound synthesis and music composition. It is implemented in C and C++ and runs on Win32, OSX, and Linux. Nyquist combines a powerful functional programming style with efficient signal-processing primitives.

***Figure 13-1.** Go to SourceForge.net and download the NyquistIDE*

After you install the NyquistIDE, let’s look at the Nyquist scripting language and the integrated development environment (IDE) optimized for writing a Nyquist digital audio script.

Installing NyquistIDE: A Nyquist Integrated Editor

Download the Setup NyquistIDE Runtime version 3.05, which for Windows is named `setupnyqiderun305.exe`, to your hard disk drive. I have included this file in the ZIP archive for the book on the Apress repository, along with Nyquist documentation, and the half-dozen Nyquist plug-ins that you looked at in Chapter 11 as well. Right-click the EXE file and then select **Run As Administrator** from the context-sensitive menu. You get the **Welcome to NyquistIDE Setup Wizard** dialog, as shown in Figure 13-2. Next, accept the **License Agreement**, click **Next**, and accept the default destination location for the installation.



Setup - NyquistIDE

Welcome to the NyquistIDE Setup Wizard

This will install NyquistIDE 3.05 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Next > Cancel

Setup - NyquistIDE

License Agreement
Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

LICENSE INFORMATION

Nyquist is built upon and derived from XLISP. Both are copyrighted software. Two copyright notices and licenses appear below.

In addition, Nyquist uses two LGPL libraries: libsndfile and liblo. These libraries are NOT covered by the Nyquist license, and Nyquist is NOT licensed under or restricted by GPL or LGPL licenses. The license agreements for libsndfile and liblo can be found in the Nyquist source libraries nyquist/nylsf and nyquist/liblo, respectively.

[✓ I accept the agreement]
I do not accept the agreement

< Back Next > Cancel

Setup - NyquistIDE

Select Destination Location
Where should NyquistIDE be installed?

Setup will install NyquistIDE into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

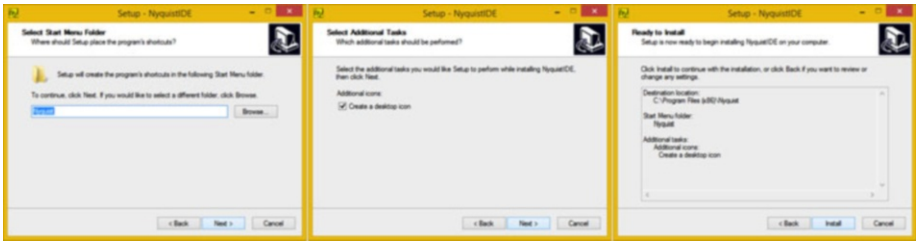
C:\Program Files (x86)\Nyquist Browse...

At least 7.4 MB of free disk space is required.

< Back Next > Cancel

Figure 13-2. Start an install, accept license, and select folder

Click **Next** and then select your **Start Menu Folder**. Click **Next** to select the **Create a desktop icon** option. Click **Next** to get to the **Ready to Install** dialog (see Figure 13-3). Click the **Install** button to start the installation process.



Setup - NyquistIDE

Select Start Menu Folder
Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Nyquist Browse...

< Back Next > Cancel

Setup - NyquistIDE

Select Additional Tasks
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing NyquistIDE, then click Next.

Additional icons:
☑ Create a desktop icon

< Back Next > Cancel

Setup - NyquistIDE

Ready to Install
Setup is now ready to begin installing NyquistIDE on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
C:\Program Files (x86)\Nyquist

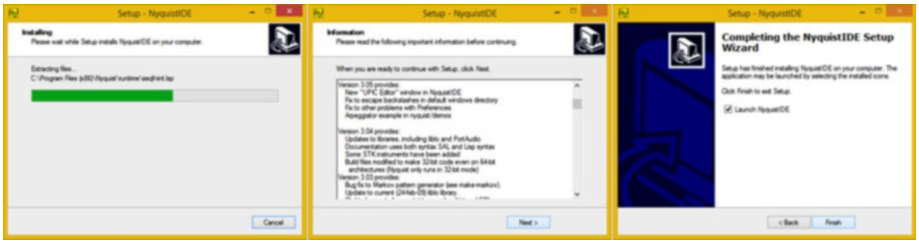
Start Menu folder:
Nyquist

Additional tasks:
Additional icons:
Create a desktop icon

< Back Install Cancel

Figure 13-3. Select Start Menu Folder, desktop icon, and Install

As you can see in Figure 13-4, you'll get the **Extracting** files progress bar, which tells you which Nyquist files are being installed to your hard disk drive. After this process has finished, you get an **Information** dialog, which informs you what each version of Nyquist 3.0 added, the current version being 3.05. After you have reviewed this, click **Next** and make sure that your **Launch NyquistIDE** check box is selected.



Setup - NyquistIDE

Installing
Please wait while Setup installs NyquistIDE on your computer.

Extracting files...
C:\Program Files (x86)\Nyquist\runtime\seqfnint.lsp

Cancel

Setup - NyquistIDE

Information
Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

Version 3.05 provides:
New "UPIC Editor" window in NyquistIDE
Fix to escape backslashes in default windows directory
Fix to other problems with Preferences
Arpeggiator example in nyquist/demos

Version 3.04 provides:
Updates to libraries, including liblo and PortAudio
Documentation uses both syntax SAL and Lisp syntax
Some STK instruments have been added
Build files modified to make 32-bit code even on 64-bit architectures (Nyquist only runs in 32-bit mode)
Version 3.03 provides:
Bug fix to Markov pattern generator (see make-markov).
Update to current (24-Feb-09) liblo library

Next >

Setup - NyquistIDE

Completing the NyquistIDE Setup Wizard

Setup has finished installing NyquistIDE on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

☑ Launch NyquistIDE

< Back Finish

Figure 13-4. Extract Files, Read Information and Launch Nyquist