

The Piltdown Process for Getting Music Into and Out of a Spreadsheet

Thanks to Dr. James C. Nourse for asking the question that prompted me to write this summary.

I won't explain the evolution of my music software suite, except to say that I have steadily tried to get myself and my limited programming skills out of the loop, using other people's software as much as possible. Over time, this goal has become more achievable, my music-writing has become more efficient, and that efficiency has been largely offset by advancing age.

Here is the current hardware/software system:

There are two computers, connected by a local wireless network. Let's call these Big and Small. Big is a Gateway desktop with four processors and plenty of disk space, including 6 TB of external storage. It runs Windows 10. Small is a Lenovo Yoga 900, also running Windows 10.

Microsoft Office and Java live on Small. I write the music in my own algebraic language, which is logically, though not visually, similar to normal music notation. I write the score using Excel, with 'text' formatting of all cells, because that gives me many nice editing features.

I have written a Java program called MugixX (currently version 4) to translate the score from Excel to MIDI, the standard representation for music data. Java has methods for generating MIDI files. MugixX now uses Apache Tika, a relatively new package that can extract text from any file format, including Excel. Many of the software pieces I've mentioned above are free.

I also have to mention NetBeans, the free Integrated Development Environment (IDE). MugixX v.4 consists of 46 Java classes, comprising 12,985 lines of source code and comments, plus the Tika library. Even though this is a smallish package, there is no way I could develop and maintain it without NetBeans. Thanks to Dr. Richard Keller of NASA for politely advising me that my earnest commitment to a folder of ratty text files had become, with the passing years, somewhat antiquated. Was I not aware, for example, of things like spell-checkers?

Once I have the MIDI file, I ship it over the network to Big. Getting this wireless connection was another recent advance that has greatly increased overall efficiency.

I next use the van Basco Karaoke Player to play the MIDI file. The Karaoke Player is free, but the next step is where "money comes innuendo," as Groucho Marx said: Karaoke Player sends, via LoopBe1 from nerds.de, MIDI data to my synthesizer, Native Instruments Kontakt 5.

Kontakt 5 contains gigabytes of digital files that capture the complex sounds of diverse instruments, as recorded in studios by expert engineers using the best equipment. It can also import files in the .gig format, dating back to the all-too-brief lifetime of Gigastudio, my first sample-based synthesizer. I have recently added free .gig instruments from Flame Studios and Gigasamples.com.

Kontakt 5 is by far the largest and most important part of the software suite. It is reasonably priced, but not free. Here is an example of the current bleeding edge in this arena: a real, one-of-a-kind instrument built so that Native Instruments could record it and market the sound library to their delighted customers (including me):

<http://www.native-instruments.com/en/products/komplete/keys/una-corda/>

The output of Kontakt 5 can be recorded as a .wav file or played through my speakers for the instant benefit of family and pets. If I like the sound, I will record it. At this point I am back in the freeware world, and I use two programs, Audacity and LAME, to edit the digital audio data, normalize it, trim it, and compress it into .mp3 format.

Here is a summary of the software I am currently using. Except as noted*, I am indebted to other people from around the world for developing, publishing, and maintaining this software and its documentation.

- Microsoft Office
- Java
- Apache Tika
- NetBeans
- MugixX v.4 (* my only contribution)
- van Basco Karaoke Player
- LoopBe1
- Native Instruments Kontakt 5
- Audacity
- LAME

From a high level, the process as I experience it has the following three phases:

1. Writing the score. This is a long, tortuous, and torturous process which occurs on Small, although I can play a keyboard (currently a Roland A-49) attached to Big to try out ideas, audition instruments, etc.
2. Playing -> correcting -> playing -> correcting -> ... The number of cycles between Big and Small is highly variable, though iteration is much faster and easier now with the wireless network in place.
3. Recording, post-processing, compressing (and sometimes posting). This occurs on Big and is super-fast compared to 1 and 2.

There is also an outer loop of MugixX debugging and modification. At this point, the program is stable enough that the outer loop is negligible. I barely do enough Java programming to stay competent. I really should document MugixX, though. Someday.

BTW, why use a Karaoke player? Because it can play synchronized lyrics, displayed on the screen as text, as well as sending MIDI data to Kontakt 5 via LoopBe1. I hijacked this feature to display text messages tracking exactly where in the score (measure and beat) the current sound is coming from, thus speeding up error-fixes. Regardless of how cute and clever this may seem, it is just a kluge betraying a shameful lack of system integration.