

# Ogg Speex: A Practical Application

Tak Auyeung

January 30, 2004

## 1 What is it?

Ogg Speex (or simply Speex) is an audio encoding method for voice. Unlike most other (or should I say *all* other) encoding methods, Speex is patent-free and open sourced. This means no one can tell others not to implement the same method, or to improve on it. This also means you will find plenty of free implementations to encode and decode using Speex.

## 2 What is it used for?

As mentioned in the previous section, Speex is for voice file compression. It is true that we already have Ogg Vorbis as a free standard for *music* compression. Speex is complementary to Vorbis because Speex is best suited for voice (dictation, lectures, etc.) compression, while Vorbis is best suited for music compression.

## 3 How good is it?

With my own lecture recording, I found that quality “2” is close enough to the original WAV file. The result is a stream that averages at less than 8 kbps. Note that kbps (lower case b) is different from kBps. 8 kbps is 8 kilo *bits* per second. That is not much bandwidth, even assuming the use of a 56kbps modem. The practical speed of a 56kbps modem is about 40kbps. In other words, at “quality 2”, one hour of voice recording can be downloaded, using a modem, in 12 minutes or so.

## 4 Getting it to work

### 4.1 The Easy Way (No Streaming)

Speex is a relatively new standard. As a result, it is not easy to find a player that handles it. There are few players for Windows, but you can encode or decode Speex files using some utility programs.

If you prefer a GUI program, got to <http://www.foobar2000.org> and download the standard edition of Foobar2000. This is a rather handy program for playing Speex files. You have to associate the extension `.spx` with Foobar2000 initially.

If you prefer a command line tool that gives you more options, go to <http://www.speex.org/download/speexb1.0.1.zip> and download the files. In Windows XP, Windows ME or later, you can open the zip file without additional software. Extract the contained files to a folder and the installation is complete.

Instead of clicking on the links to Speex files, right-click and use the “save target as” or “save link as” option to save the file. Go to File Explorer and double click on the files with `spx` extension. Windows will ask you how to open a file of this extension. Choose “others”, and navigate to the folder containing the files you extracted earlier. Associate the `spx` extension with the program `speexdec`. Click “Okay” and we are all done. You should hear the playback as soon as you click “Okay”.

## 4.2 The “Home-brew” Streaming Player

The previous method works, but it requires the whole file be downloaded before playback. It’s okay if you either have enough bandwidth to download quickly, or that you are patient enough to download via a slower connection.

With a little bit of virtual elbow grease, you can “stream” the voice file over a slower connection so you don’t have to save the whole file first. To do this, you’ll need to know more about using the command line interface of Windows.

First, download and unzip the package from <http://curl.haxx.se/latest.cgi?curl=win32-nossl>. This package contains an executable file (along with some other documents) called `curl.exe`. This is a *cool* command line program that downloads stuff from the internet.

Before we use `curl` in conjunction with `speexdec`, let’s talk about getting Windows to find these executable files. If you know how to set the `PATH` environment variable, just make sure it includes the paths to the folders containing `curl.exe` and `speexdec.exe`. If you don’t want to mess with `PATH`, a quick-and-dirty method is to unzip the Speex and `curl` packages to the Windows folder. This folder is usually `c:\windows` for Windows 95/98/me and `c:\winnt` for Windows NT/2000/XP. The danger is that you may overwrite existing files called `curl.exe` and `speexdec.exe`. I have yet to find one Windows installation with these filenames.

Once you know Windows knows how to find `curl.exe` and `speexdec.exe`, we can begin the fun. Let’s take the file at <http://www.drtak.org/teaches/ARC/cisp317/0821.spx>. Instead of saving the file, copy the link to the clipboard. Then open a command line interface and type in the following (rather long) command:

```
curl -m 7200 http://www.drtak.org/teaches/ARC/cisp317/0821.spx | speexdec -
```

You can click on the upper-left icon in a command line interface and use “paste” to paste in the URL to save some typing.

This home brew streaming method works, but it does not guarantee a minimum buffer size. In other words, if your network connection is slow, you may encounter some choppy playback. In most systems, the built-in buffer size of a pipe is sufficient to ensure a relatively smooth playback as long as you have enough average bandwidth.

### 4.3 Converting to .WAV

It is easy to convert from .SPX to .WAV. If you want to convert as you download, you can use the following command:

```
curl -m 7200 http://www.drtak.org/teaches/ARC/cisp317/0821.spx | speexdec - 0821.wav
```

If you download the file first (let's say it is called 0801.spx), you can use the following command to convert it:

```
speexdec 0821.spx 0821.wav
```

Once a .WAV file is created, you can use the media player bundled with Windows to play it. On most systems, you simply have to double-click on the .WAV file to start playing back.

The nice part about using a “regular” player to play the .WAV file is that you can repeat, pause, resume and fast-forward.

For those who like a challenge (and understand OS concepts), here's my challenge. In Windows (and Linux, of course), one can create “named pipes”. A named pipe has a path of `\\.\pipe\whatever`. The nice thing about a named pipe that is the producer of the pipe can be a program completely different from the consumer of the pipe. In other words, it'd be nice to be able to use a command line like this:

```
curl -m 7200 http://www.drtak.org/teaches/ARC/cisp317/0821.spx | speexdec - \\.\pipe\0821.w
```

This command streams the voice file to convert from .SPX format to .WAV, but the new “file” is now a named pipe. If the media player accepts named pipes as filenames, then you can specify `\\.\pipe\0821.wav` as a file. This way, you get the cool features of your regular media player software while using the highly compressed Speex encoding for over-the-net transmission.

Of course, whether the default Windows media player will take a named pipe as a file is another story.

## 5 Recording

There are many ways to record voice digitally. For example, one can use an analog mini-cassette or micro-cassette tape recorder, then playback to the line-in or microphone jack of a computer. This method, however, is slow because the digitization takes as much time as the original length of recording.

A better way to do this is direct digitization. In other words, voice is directly digitized by some device. There are two ways to do this.

## 5.1 Computer digital recording

First, a desktop or notebook computer can be used to perform the digitization in real time. This is hardly a problem from the perspective of resources. Most desktop computers have much more storage space and processing power than necessary to digitize voice (and encode in real time). The problem with this approach, however, is that the microphone must be connected to the computer. In the case of a meeting, that may be possible. However, in a lecture, the lecturer may want to pace and not stay next to a desktop or notebook computer. An expensive solution to this problem is to use wireless microphones.

## 5.2 Handheld digital recorders

Second, a handheld digital voice recorder can be used perform life digitization. This is the preferred method for lectures because newer digital voice recorders are very light and easy to carry. Please read <http://www.drtak.org/likes/steno/> for the review of such a device. Surprisingly, digital voice recorders are now less expensive than a good wireless microphone set up.

## 5.3 Encoding

Most digital recorders, be it an application on a computer or a handheld device, do not encode in Speex. Hopefully, in the future, Speex will be better supported.

For the time being, most computer-based recorders record in some variant of the WAV format. If the WAV format is PCM16, `speexenc` (the Speex encoder software) can handle it. For other WAV variants, you may need a sound file converter to convert it to PCM16 format first. Do not be alarmed by the large size of the PCM16 WAV file, it will be about 30 times the size of the equivalent Speex file.

If you use Linux, you can use `sndfile-convert` to translate from one file format to another. This utility program is rather versatile.

Once you have a file that is in PCM16 WAV format, you can use `speexenc` to encode it to Speex format. Although `speexenc` has much flexibility in terms of controlling the outcome, the easiest method to control the output is via the `--quality` option. The default is “quality 8”, but that produces a file that averages at more than 15 kbps.

From experiments, a setting of “quality 2” is sufficient for lecture purposes. This yields an average bit rate of less than 8kbps.