Harmonizer User Manual:

How and where to compile.

The C executable has 2 main dependencies:

GSL (GNU Science Library)	ftp://ftp.gnu.org/gnu/gsl/
LIBXML2	ftp://xmlsoft.org/libxml2/

Please follow the configuration instructions in the GSL and LIBXML2 distributions that you download (see INSTALL and or README). The basic configuration and compilation commands are:

```
$ ./configure
$ make
$ make install
```

If this gives you trouble, add the flag --help to see other compiler options:

```
$ ./configure --help
```

After this, the next step will be to configure the makefile included with your Harmonizer distribution to find and link these libraries.

Notes: In order to compile the GSL on the Appliance, you will need root privileges. These can be obtained by typing:

```
$ sudo su
```

at the command line. In order for the GSL to run properly, shared libraries need to be disabled during the compilation process. If that is not done, the commands:

```
$ LD_LIBRARY_PATH=/usr/local/lib
$  export LD_LIBRARY_PATH
```

at the command line (after obtaining root privileges) will allow the program to compile and run correctly.

For Mac OS X distribution:

The makefile comes equipped with two variables that are relevant to linking in the header and object files. The `-I/usr/include` and `-I/usr/local/include` flags below tell the compiler to look in the specified directories for the header files. The other flag in the GSL_LIBS variable, `-L/usr/local/lib,` tells the compiler where to find the object code associatedd with these libraries. Locate these directories in your particular installations and change the example to the appropriate path.

```
XML_LIBS = -I/usr/include -lm -lxml2
GSL_LIBS = -I/usr/local/include -L/usr/local/lib -lm -lgsl
```

For Appliance distribution:

The makefile in the Appliance folder contains the following flags, which tell the compiler where to look for header files: `-I/usr/local/include/libxml2` and `-I/usr/local/include.` The other type of flag `-L/usr/local/lib,` tells the compiler where to find the object code associated with these one of these libraries. Linker flags signal the compiler to link in the object code for the libraries at hand. Locate these directories in your particular installations and change the example to the appropriate path to compile on your home computer.

```
-I/usr/local/include/libxml2/ -lxml2 -I/usr/local/include
-L/usr/local/lib -lm -lgsl -lgslcblas
```

Once you have compiled import with the command

```
make import
```

place the executable file called "import" into the appropriate "dist" directory in the Harmonizer GUI folder system. (Source/Java/ Harmonizer2/dist). Double click Harmonizer.jar to run Harmonizer. Note: in the appliance, you will need to run Harmonizer from the command line by cd-ing in to the dist directory and typing the command `$ java -jar "Harmonizer.jar".` Your output files will be saved to the directory dist. (Or use relative paths to save your xml file in an arbitrary location.)

Note: import currently attempts to open the xml file in ©Finale Notepad 2012, a free version of Finale available for download online. This can be reconfigured in the source code if desired.

You may also view your outputted xml files in any of the editors on this page:

http://www.makemusic.com/musicxml/community/software

We have found MuseScore to be a useful editor, which displays musicxml scores and also outputs WAV files. It is a bit more difficult to find an open-source editor that will run on the appliance, so often it is easier to run the harmonizer program in the appliance and then copy the resulting xml files out into a different platform. Both Windows and MacOS can run MuseScore.

How to configure and use.

To load the Harmonizer GUI, double click on the Harmonizer.jar file (or run the equivalent command-line code). This will reveal a GUI with four panels: Upload, Determine BPM, Configure, and Output. First, select the WAV file of the melody for which you would like to generate harmonies. This is done by clicking the "Select File" button. Only directories and WAV files should be visible to you, and only WAV files should be selectable. Once you've picked a WAV file, you can test that it uploaded properly by verifying that the name of the file was displayed below the "Select File" button, and by clicking the Play button to listen to it.

Next, turn to the Determine Beats-Per-Minute (BPM) panel (assuming the BPM of the song is not known). The BPM aids in determining the length of time of a given quarter note. Play the file using the Play button referenced earlier, and click the "Click to the Beat" button in rhythm. This will estimate the BPM based on a moving average of the lengths of times between clicks. Then, once the BPM has converged to a particular value or small range of values, input your estimated BPM into the textbox, and continue on. (The value does not need to be perfect—but the closer it is to correct, the more likely the outputted sheet music is to have the proper note lengths.)

Then move on to the Configure panel for some configuration fun. Select the time signature of the song (you are limited to 4/4 or 3/4 time), the key of the WAV file, the key you would like your output file to be in (melody and harmonies), the number of pickup beats (without these, our algorithm won't be able to pick out the downbeat, and you will wind up with an oddly peppy sprung rhythm), the number of parts you would like in your generated harmony, and the length of note you would like your harmony to consist of. Now you are ready to tackle the Output panel.

In the Output panel, write the name of your file (the name under which it will be saved in the output directory), your name in the composer field, and the title you would like to have appear at the top of the generated sheet music. Then, click the "Harmonize" button below.

If the harmony generates successfully, a congratulatory message should appear under the "Harmonize" button letting you know. If it fails, an error message should appear. If, for some reason, some of the input configurations were either incorrectly entered (BPM not in a reasonable range, for example), an error message will pop up, letting you know what went wrong and what needs to change. Have a wonderful Harmonize experience.

To test this out for yourself, try it on jingle_bells.wav in the samples folder. Input these configurations:

Beats Per Minute: 140 (though you are free to try our click calculator to verify that that's a good estimate)
Time Signature: 4/4
Original Key: C
Transpose To: (any key you want)
Pickup Beats: 0
Number of Parts: (Your choice)
Speed: (We recommend whole, but if you're feeling adventurous, half or exact are a party on a page)
Then, after filling out the file name, composer, and title fields, click Harmonize!

You can also try it on somewhere.wav. (Somewhere Over the Rainbow)
Beats Per Minute: 120
Time Signature: 4/4
Original Key: C
Transpose To: (any key you want)
Pickup Beats: 0
Number of Parts: (your choice)
Speed: (Half, whole – anything but dotted half, though it is kind of endearing)

If you don't like the results (sometimes we don't), click harmonize again to generate a different harmony. Our program never replicates exact harmonies.

If you're still feeling adventurous, check out oh_come.wav (Oh Come All Ye Faithful).
Beats Per Minute: 108
Time Signature: 4/4
Original Key: G
Transpose To: (any key you want)
Pickup Beats: 0
Number of Parts: (Your choice)
Speed: (Try half, whole)