How to build msound, using MATLAB

To build msound using the LCC compiler (included with MATLAB,) simply run the provided `compile_msound.m` function. This will build the msound MEX-file using default configuration, i.e. Windows MultiMedia Extensions (WMME) audio API support, only. (The actual audio i/o will be implemented using the PortAudio library v19 snapshot from November 10th 2009, located in the `portaudio` subdirectory.)

In order to add support for Steinberg Audio Stream Input/Output (ASIO) API, a C++ compiler (e.g. the Microsoft Visual Studio / Microsoft Visual C++) and Steinberg's ASIO Software Development Kit (SDK) are required. Microsoft has a free "Express Edition" of Visual C++, which is available at http://www.microsoft.com/express. You may either use the web-based installer or download the CD/DVD-image. In order to have all the required features, the Visual C++ compiler does require Microsoft's Windows SDK to be installed, too. It is available at http://www.microsoft.com/downloads/details.aspx?FamilyID=c17ba869-9671-4330-a63e-1fd44e0e2505. The ASIO SDK (version 2.2) may be downloaded at http://www.steinberg.net/en/company/3rd_party_developer.html, and is also free of charge but requires creation of a developer account. The downloaded zip-file needs to be unzipped to the `ASIOSDK2` sub directory.

Now, MATLAB needs to be told to use the Visual C++ compiler for building MEX-files. This is achieved by typing `mex -setup` at its commando prompt. MATLAB will ask if you like to locate installed compilers; the answer is `y` (yes.) Now you have to pick the Microsoft Visual C++ from the list of available compilers; in most cases this'll be answer 2.
If the Visual C++ compiler doesn't show up, either your MATLAB or the Visual C++ compiler is too old, i.e. the MATLAB version at hand doesn't support the installed version of Visual C++. In this case compiling msound via MATLAB doesn't work, but you may try compiling msound using the Visual Studio integrated development environment (IDE), see below.
After your compiler choice has been verified by pressing `enter`, MATLAB should ready to build msound. So simply execute the provided `compile_msound_asio.m` function.

How to build msound, using Microsoft Visual Studio 2003

Building msound via the Visual Studio IDE is some what more complicated and error prone than using the procedure as described above, so you'll be better off if you stick to compiling via MATLAB. If you still decide to use the Visual Studio IDE, be aware that this approach has the same requirements (Visual Studio and ASIO SDK) as the one described above. (Please note that the included pictures are out of date. I didn't have the time to update them to the current msound version, but you'll get the idea, right?)

This guide is based on the Microsoft Visual Studio 2003, since this is the author's Visual C++ (MSVC++) version of choice because it does include the Windows Platform SDK, already. In addition the MSVC++ 2003 project files can be easily imported into newer versions of the MSVC++.

The msound source code package includes two Visual Studio project files, one for building an old style `msound.dll` MEX-file and another one for the newer `msound.mexw32` file. Both project files are part of a common solution `msound_msvc2003.sln` and are able to build with or without ASIO support. So to get started simply open the solution-file, and select your project and configuration of choice.

The provided project files may require you to adapt some path specification according to your own system setup. In order to configure the project to use your local MATLAB external interfaces SDK, you'll have to open the project properties window and select "all configurations" in the upper left drop down box. Then adjust the compiler's include path by selecting the C/C++ group, as shown in Figure 1 and updating the "Additional include directories" entry.

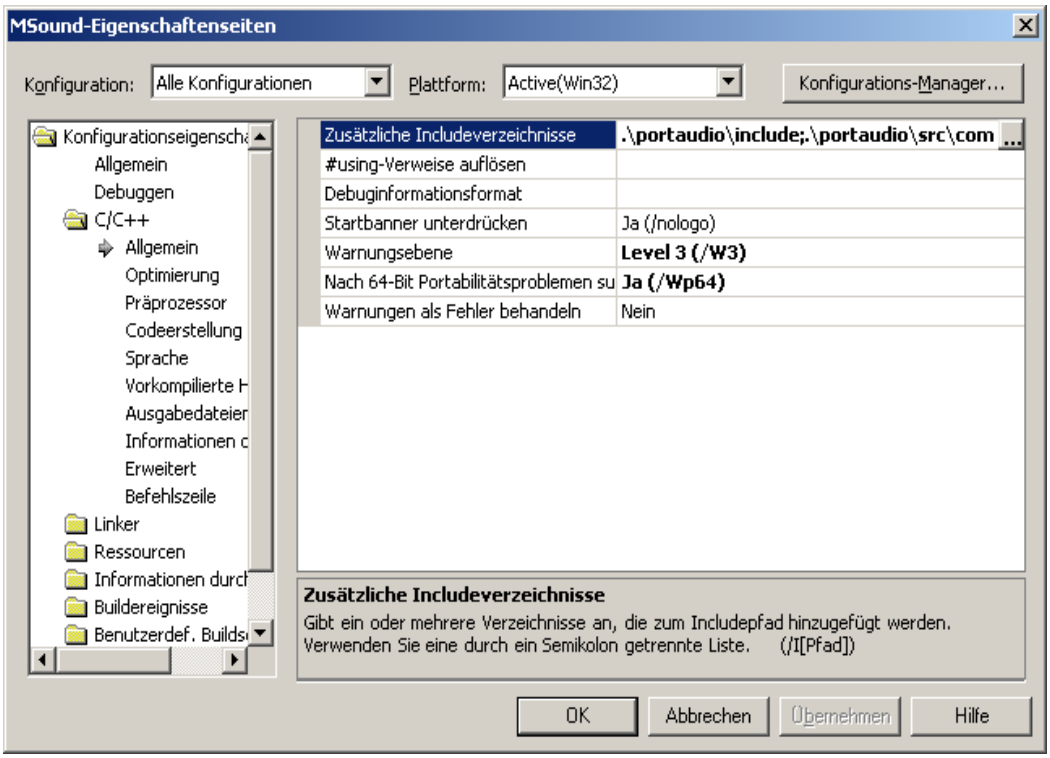

Figure 1: Adjust the compiler's include path to match MATLAB's external interface include directory.

The second section to be modified is the linker's section, as shown in Figure 2. Select and adjust the "Additional library directories" entry. Be sure to specify the correct path, leading to

the MATLAB external interface libraries, suitable for your compiler, since MATLAB usually comes with libraries for several different compilers.

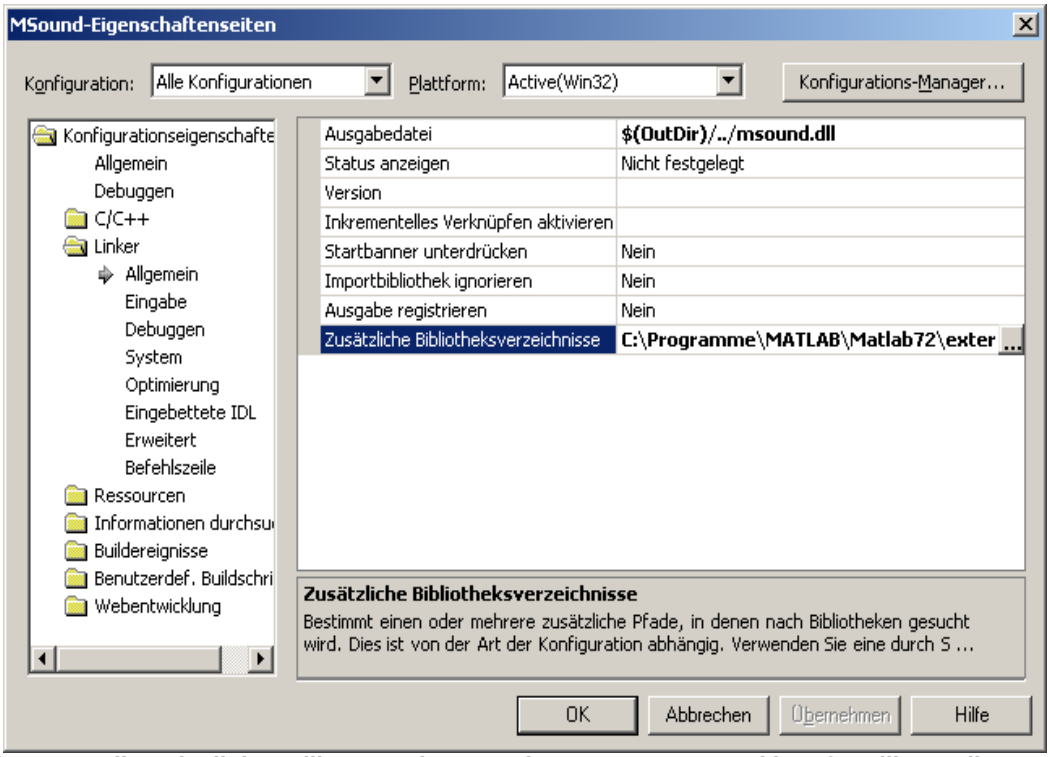

Figure 2: Adjust the linker's library path to match MATLAB's external interface library directory.

Hit the “OK” button to close the project properties window. Then select the msound configuration you want to build, Figure 3. Note there are also configurations to build msound without ASIO support. So if you don’t have the ASIO SDK you’ll have to use one of the configurations, marked by “(without ASIO)”.

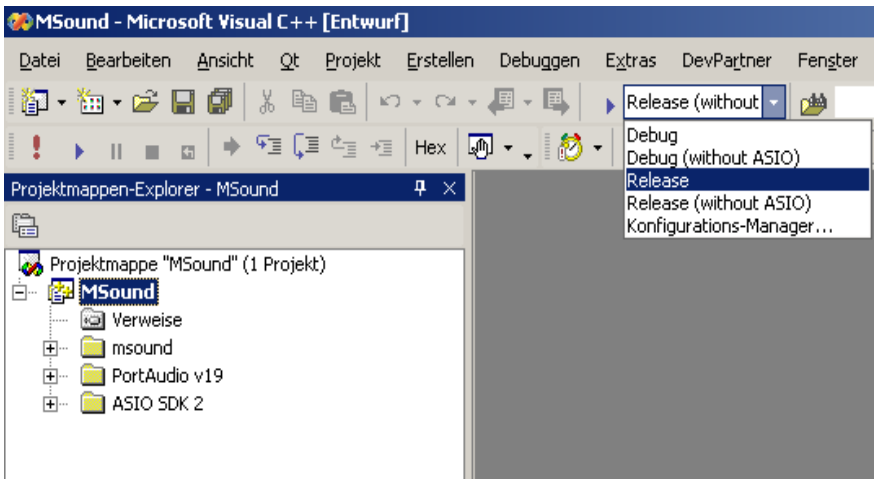

Figure 3: Select the msound configuration to be build.

Now, you may use the build menu to compile the msound project.