Guide to the *csound~* Sources: Davis Pyon

csound~ (v1.0.7) – by Davis Pyon

These are the instructions for building the csound~ MaxMSP external on Windows or Mac OS X. The build process is different on each platform. For the latest news and downloads concerning csound~, visit http://www.davixology.com/csound~.html.

Dependencies

You will need to install Csound 5 (http://sourceforge.net/projects/csound).

You will need the MaxMSP SDK (http://cycling74.com/products/sdk/).

Mac OS X

To build csound~, you will need **Xcode 2.4** or later. Install Xcode from the Mac OS X Install DVD or get it from http://developer.apple.com/technologies/xcode.html.

Windows

To build csound~, you will need **Visual C++ 2005** or later. The free Express Edition of Visual C++ can be obtained at http://www.microsoft.com/express/Downloads/.

Download the **pexports** command line tool from: http://www.emmestech.com/software/pexports-0.43/download_pexports.html.

Download **win32 pthreads** at http://sourceware.org/pthreads-win32/.

Download **libsndfile** at http://www.mega-nerd.com/libsndfile/#Download.

Building csound~

Mac OS X

After installing Csound 5 and the MaxMSP SDK, make sure the /Library/Frameworks folder contains:

CsoundLib.Framework or **CsoundLib64.Framework**
MaxAPI.framework
MaxAudioAPI.framework

CsoundLib.Framework is installed automatically by Csound5.pkg or CsoundLib.pkg These packages are found in the Csound 5 disk images for Mac OS X.

Also, make sure that **libsndfile.1.dylib** (a symbolic link to the actual libsndfile dynamic library) exists in /usr/local/lib.

Open **csound~.xcodeproj** in the **src** folder.

Under Groups & Files → Targets, double-click either **csound~ float** or **csound~ double**, depending on which version of Csound 5 you installed. Select the Release configuration.

Under General → Search Paths → Header Search Paths, change the paths to match your installations.

Under General → Build Locations, change Build Products Path and Intermediate Build Files Path to reflect your externals folder and build folder locations. The externals folder is typically:

/Applications/Max5/Cycling '74/msp-externals/_my_externals

Press Cmd + Shift + B, or select Build → Build Results from the menu.

Set Active Target to **csound~ float** or **csound~ double** depending on which version of Csound 5 you installed.

Set Active Build Configuration to Release.

Press Cmd + B, or select Build → Build from the menu.

Windows

Update your PATH variable so that you can run lib.exe. It is typically located at:

C:\Program Files\Microsoft Visual Studio 9\VC\bin

To update you PATH variable on Vista or Windows 7:

a) Press Windows + R
b) Type “sysdm.cpl”, then press enter
c) Go to “Advanced” tab
d) Click “Environment Variables...”
e) Under “System Variables”, double click “Path”
f) At the end of the string, enter “;C:\Program Files\Microsoft Virual Studio 9\VC\bin”

Copy pexports.exe to one of the directories listed in your PATH variable.

Open Command Prompt (not Cygwin or Msys). Then type the following commands:

```
$ vcvars32.bat
$ cd C:\Program Files\Csound\bin
$ pexports csound__.dll.5.1 > csound__.def
$ lib /machine:i386 /def:csound__.def
```

replacing "__" with "32" or "64". The numbers stand for bits. So, if you downloaded Csound5 floats version, then it's 32bit. For Csound5 doubles version, it's 64bit. The Csound5 installer file will give a hint in the name with the letter 'd' or 'f'.

Open **csound~.vcproj** in the **src** directory.

Press Alt + F7, or select the menu item Project → Properties.

Go to Configuration Properties → C/C++ → General → Additional Include Directories. Change all include paths to match your installations (Max, Csound5, pthreads, libsndfile).

Go to Configuration Properties → Linker → Input → Additonal Dependencies. Change all lib paths to match your installations (Max, Csound5, pthreads, libsndfile).

If using MaxSDK version 5 or later, go to Configuration Properties → Linker → Input → Ignore Specific Library. Then add **maxcrt.lib**.

Go to Configuration Properties → Build Events → Post-Build Event → Command Line. Change the destination directory to match your MaxMSP installation. You may need to change the permissions on the destination directory or run VC++ as Admin. Or, you can just delete the copy command and copy the resulting .mxe file manually after building.

Press F7 to compile csound~.

If you deleted the copy command, copy **csound~.mxe** from the **src\build** directory to:

C:\Program Files\Cycling '74\Max 5.0\Cycling '74\msp-externals\.

If you get an error like "stdint.h not found", download **stdint.h** from:

http://en.wikipedia.org/wiki/Stdint.h

and move it to:

C:\Program Files\Csound\include

If you get an error like "__int64_t not defined", open **stdint.h** and add the line:

```
typedef __int64 __int64_t;
```

Running csound~

In order to run csound~, you must have MaxMSP 4.6.3 or later installed. After **csound~.mxe** or **csound~.mxo** is built and in the proper location, you can open one of the Max patches contained in **examples** or **help_files** folders. Max patches end with **.maxpat** or **.maxhelp**.

Known Issues

On Windows, **fluidOpcodes.dll** can cause MaxMSP to crash. Typically, you will encounter a crash when clicking on a non-MaxMSP window after stopping a Csound performance, deleting a csound~ instance, or closing a patch that contains csound~. This may be fixed in a future release of Csound 5, but as of version 5.11.1, the bug still exists. To fix, move or delete fluidOpcodes.dll from the Csound 5 **plugins** or **plugins64** directory (e.g. C:\Program

Files\Csound\bin\plugins64).