

JACK4U Manual

vers. 1.1

monoflow
new media experience

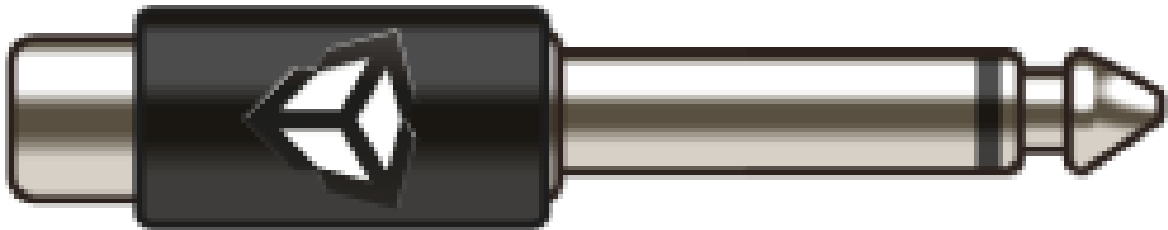
Contents

1	Introduction	1
2	System Requirements	3
3	System Overview	5
3.1	JACK Audio Components	5
3.2	ASIO Driver	5
3.3	Unity with PortAudio	5
4	Installation & Settings	7
4.1	Installation via the JACK4U Editor	7
4.1.1	Installation	7
4.1.2	Settings	9
5	JACK Audio Setup with Qjackctl	11
5.1	Setup	11
5.2	Messages	13
5.3	Connect	14
5.4	Patchbay	14
6	Unity Setup	17
6.1	Set the AudioSettings	17
6.2	JACK4U Editor	17
6.3	JACK4U Connection	19
6.4	Check your AudioListener in the scene	19
7	Run the system	21
7.1	Start the JACK Audio Server	21
7.2	Start your DAW	22
7.3	Start JACK4U Connection	23
8	Known Issues	27
9	FAQ	29

10 History	31
11 Credits	33
12 Links	35
13 Support	37

Chapter 1

Introduction



- **JACK4U** is a tool to create Unity applications that can receive audio stream data from the JACK Audio system on Windows. With JACK Audio it is possible to route your audio from a DAW (Digital Audio Workstation) like Ableton Live to other applications with the lowest latency.
- **JACK4U** provides this audio data to Unity for further processing. It also makes the process of installing and setup a JACK Audio system very easy. With just a few clicks you are ready to use Unity as a VJ audio visualization engine.

Chapter 2

System Requirements

- 32Bit/64Bit **Windows 7+**
- Unity 5.x : works in **Free** version
- Unity 4.x : you need **Pro** version
- **ASIO driver** must be installed on the machine (ASIO4ALL driver comes with the package)
- **JACK Audio Connection Kit** must be installed on the machine (Installer comes with the package)

Chapter 3

System Overview

3.1 JACK Audio Components

JACK Audio is a system for routing audio between applications just as one would plug audio cables in and out from a mixing sound desk. It uses a Server <-> Client architecture to supply the audio connection between your sound device and the ASIO application(s).

- **JACK Audio Server** (Server process 'jackd') is the main component that provides all the resources to route the audio.
- **JackRouter** Once the JACK Audio Server is running, the JackRouter driver is then able to create connections between clients (ASIO applications) and the JACK Audio Server. After installing JACK Audio the new device "JackRouter" will be available in your DAW/Unity when you use the ASIO driver. We have to use this device to route our audio data to the JACK system via ASIO.
- **Qjackctl** The graphical administration tool Qjackctl makes the administration of the JACK Audio Server easy and provides a patch bay where you can route your audio streams (in and out) of your applications.
- JACK Audio uses the **PortAudio** lib on Windows to connect to several audio interface APIs.
- **PortAudio** is an abstraction layer that bundles several Audio Interfaces behind one API. (ASIO, Windows Direct Sound, MME)
- To route the audio from a DAW program like Ableton Live, NI Traktor or PureData we have to use ASIO as audio API.

3.2 ASIO Driver

To use ASIO we need to install a driver. If you have installed an ASIO device you probably have already a driver that you could use. For all others there is a free alternative:

- **ASIO4All** free ASIO driver
- When the ASIO driver is installed you could use ASIO as driver type in your application. Normally you now could choose the "ASIO4All" audio device in your DAW for audio output, but this is not the device we want to use (see JackRouter).

3.3 Unity with PortAudio

- To connect Unity to the JACK Audio System we also use PortAudio

- If we have configured it with the right parameters (ASIO as audio API for example) we could join the party with our Unity app.
- PortAudio provides a callback method where the audio data comes in and we could inject this data into the Unity audio system.
- (JACK4U will do all this steps for you)

Chapter 4

Installation & Settings

The installation steps could all be done by hand (perhaps you have already a running JACK Audio system on your computer) but as it could be difficult for novices and cumbersome to grab all the components from the internet, JACK4U comes with all the components and guides you through the whole process from within Unity.

- First you should import the JACK4U.package from the Assetstore into your project.
- If you use Unity 4 : **Move the "Plugins" folder into the root of your project.**
- In the Assets folder you should have now the JACK4U folder and a Plugins folder where the PortAudio.dll is located.

4.1 Installation via the JACK4U Editor

The JACK4U Editor guides us through the installation process and provides the controls for easy administration. To get a running systems there have to be made a couple of steps:

1. Installation of several components on your machine.
2. Setup of several paths so JACK4U could help you later to control your system in an easy way.

4.1.1 Installation

- First open the JACK4U Editor (Windows/JACK4U/Editor)
- You see an Editor with three tabs at the top. Click on the "Installation" tab und you will see the buttons that will do the installation/registration of the components you need:



Figure 4.1: JACK4U Editor Installation

1. **Install the ASIO4All driver** if you have no ASIO driver on your system.
2. **Install JACK Audio for Windows.** This installs the Jackd server, JackRouter.dll and the administration tool Qjackctl. JACK4U also checks what Windows version you are running (32/64 Bit) and chooses the right version of the software to install.
3. **Register the JackRouter.dll on 64 Bit systems.** As the JACK Audio installer is a 32 Bit application regardless of the version (32/64 Bit), it can auto register the 32 Bit version of the JackRouter dll but the 64 Bit version has to be registered manually. (If you only use 32 Bit DAW applications you don't need to register the 64 Bit version). The register button only works if you installed the JACK Audio system at the default location that the installer uses. Otherwise you have to open your cmd console as administrator and type : `regsvr32 "C:\Program Files (x86)\Jack\64bits\JackRouter.dll"` . (Use the path of your installation.)
4. When you install a new version of JACK Audio on a computer that already has an old version you should first unregister the old JackRouter.dll. (Or by hand: `regsvr32 -u "C:\Program Files (x86)\Jack\64bits\JackRouter.dll"`)

4.1.2 Settings



Figure 4.2: JACK4U Editor Settings

- Here we could specify several paths that we use for better administration of the system. In general you could do everything by hand in the Windows Explorer but to get the right feedback from other applications it is mandatory to start the apps from Unity.
 1. Qjackctl application as this is our tool to administrate and control the JACK Audio Server
 2. A DAW like Ableton Live or NI Traktor for example as our audio source for JACK
- After we specified the paths to the exe's we can open&close these programs from the "Controls" tab. (If you click on the displayed path the Windows Explorer opens at the directory of the path)

Chapter 5

JACK Audio Setup with Qjackctl

To setup the JACK Audio Server we use the graphical administration tool Qjackctl.

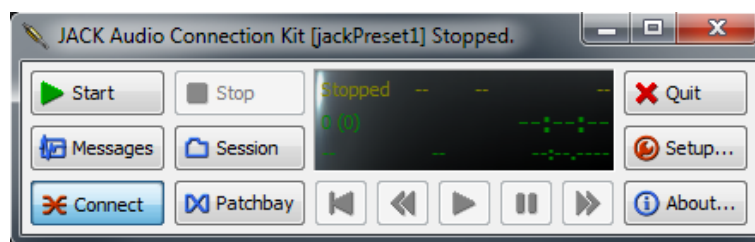


Figure 5.1: Qjackctl

First we have to setup our Server:

5.1 Setup

- Go to the Settings tab.

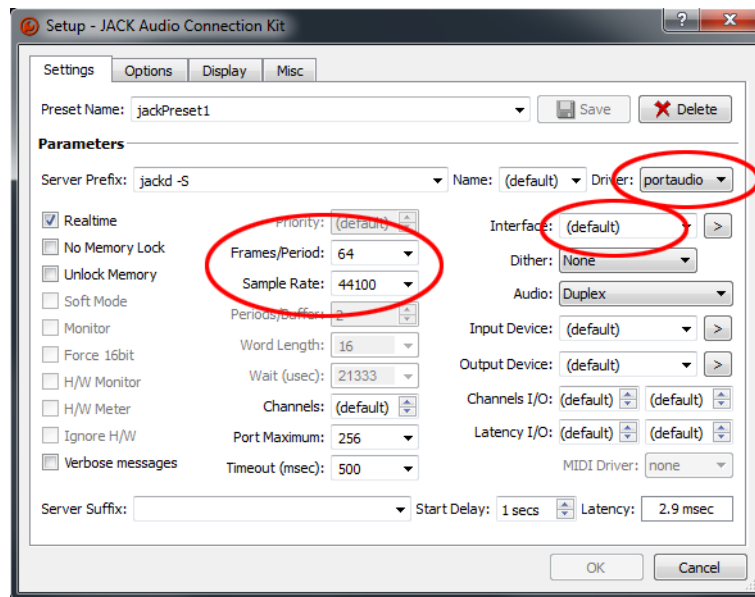


Figure 5.2: Settings

- Choose the right driver: "portaudio"
- Choose the right Interface: Depending on you Windows version and your installed ASIO drivers you have to use here "(default)", "ASIO::ASIO4ALL v2" or the driver from your ASIO soundcard. In our internal tests we used on Windows 7 "(default)" and on Windows 8 "ASIO::ASIO4ALL v2", respectively "ASIO::Focusrite USB 2.0 Audio Driver" with a Focusrite Scarlett 2i4 card.
- Set the right buffer size (Frames/Period) to get the minimum latency without dropouts. This depends on how capable your computer system is. (Low values => better latency but higher probability to get audio glitches&dropouts) You have to make your own tests but a value of 64/128 should work on most of the systems without problems.
- Set a Sample Rate. (Usually 44100)
- The rest of the Settings should be left to the default settings
- At the Misc tab you could set additional parameters that are useful. It is recommended to set the Start/Stop JACK audio server options.

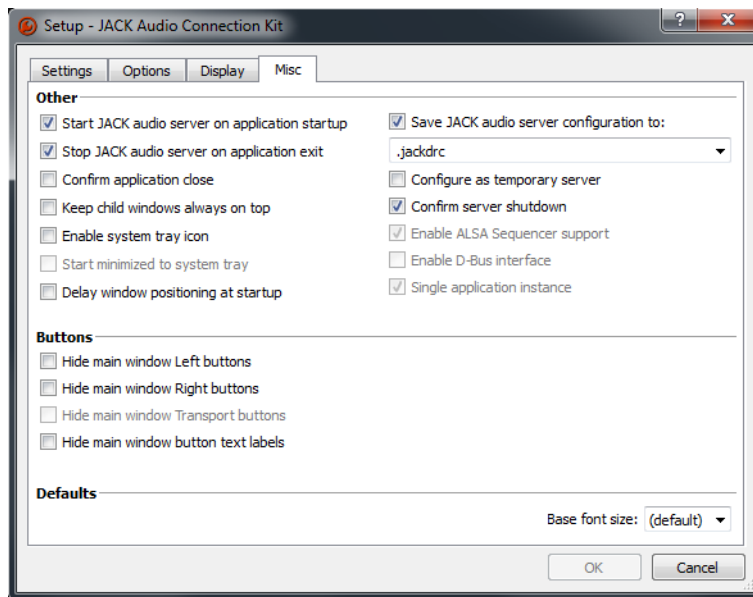


Figure 5.3: Misc

5.2 Messages

- The Messages window acts like a logging system and shows the status messages of the system. It is very useful for debugging if you have any problems with JACK Audio.

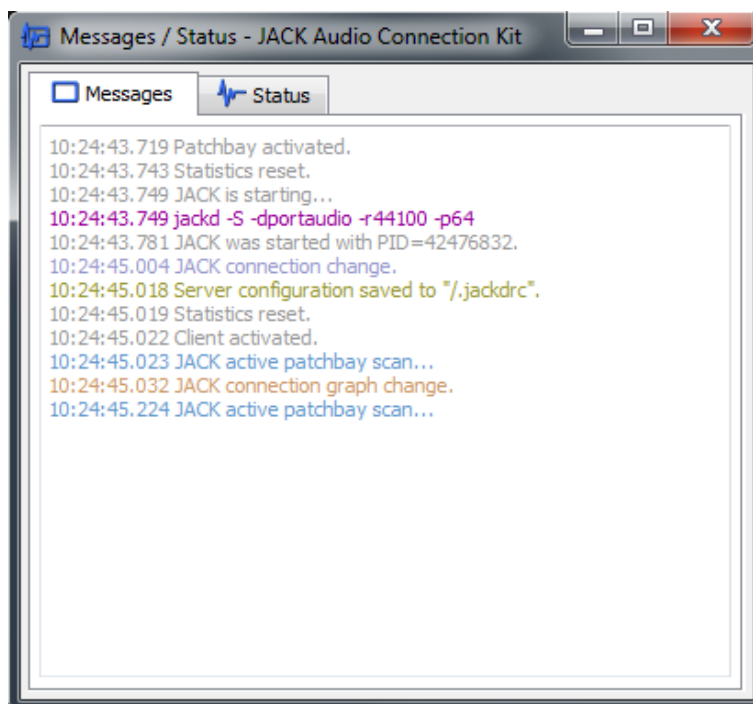


Figure 5.4: Messages

5.3 Connect

- In the Connect window you can do the routing of your audio. On the left side all available applications with their output ports are shown. On the right side all input ports are visible. Actual connections are established between source (Output Ports) and destination (Input Ports) by selecting one item in either side and pressing the Connect button at the bottom-left of the Connections window. It is also possible to establish a connection by dragging an item and dropping it on to the other side (Note: it doesn't matter which side you start dragging from).

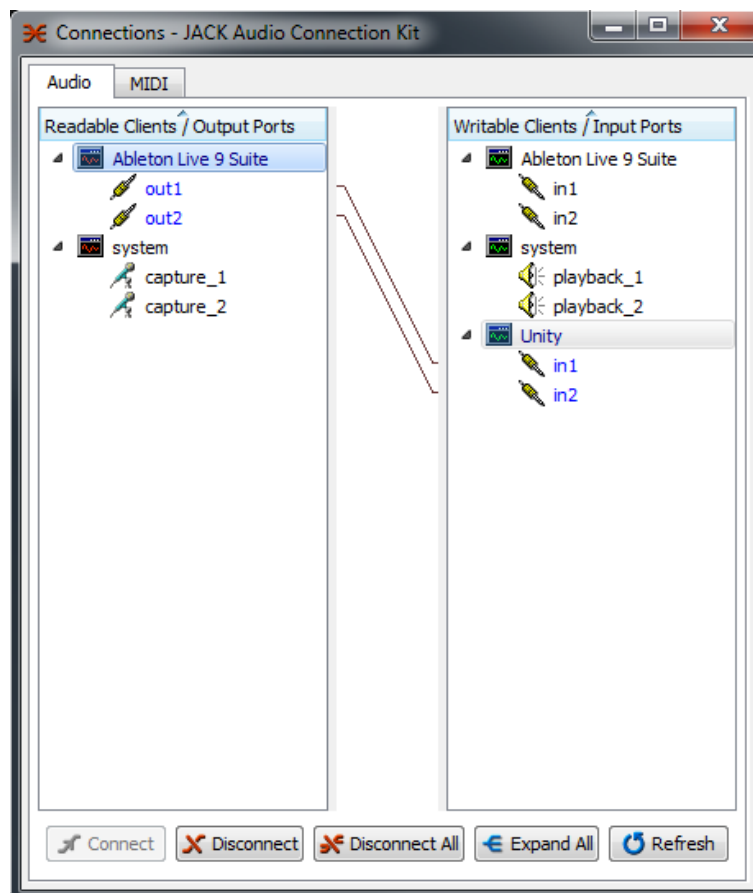


Figure 5.5: Connections

5.4 Patchbay

- All connections made in the JACK Control interface persist as long as the JACK session lasts. To maintain a custom and persistent connection layout you'll have to edit and activate a patchbay definition profile. Patchbay definition profiles are connection models that are edited and created on the JACK Patchbay window, which is accessed via the Patchbay button on the main JACK control panel.
- The JACK Patchbay window is very similar to the Connections one. However, it is here where you prepare and set your intentional persistent connections, not actual connections. When activated, this patchbay definition profile will keep all declared connections automatically, as long as the JACK Control Panel is kept active.

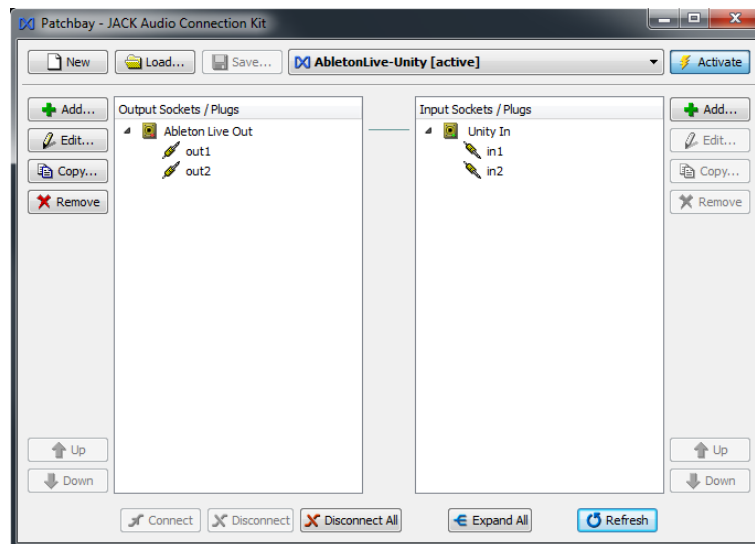


Figure 5.6: Patchbay

Chapter 6

Unity Setup

6.1 Set the AudioSettings

- Since Unity 5 you can set the sampling rate of Unity via “Edit/Project Settings/Audio”
- In the AudioManager you should set the “Sample Rate” to the value you use in JACK and your DAW.
- If you left this value to 0 Unity uses the Sampling Rate that Windows uses (default 48000) so beware as most of the time you should use here 44100.

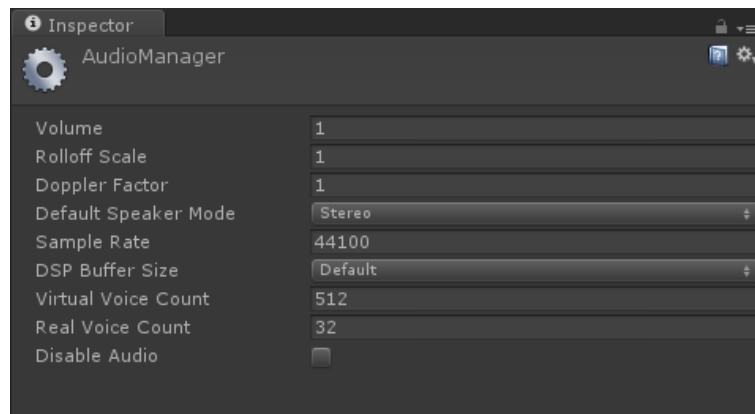


Figure 6.1: AudioSettings

6.2 JACK4U Editor

- At the Controls section of the JACK4U Editor you have the main tools to open & close the components of the JACK Audio system

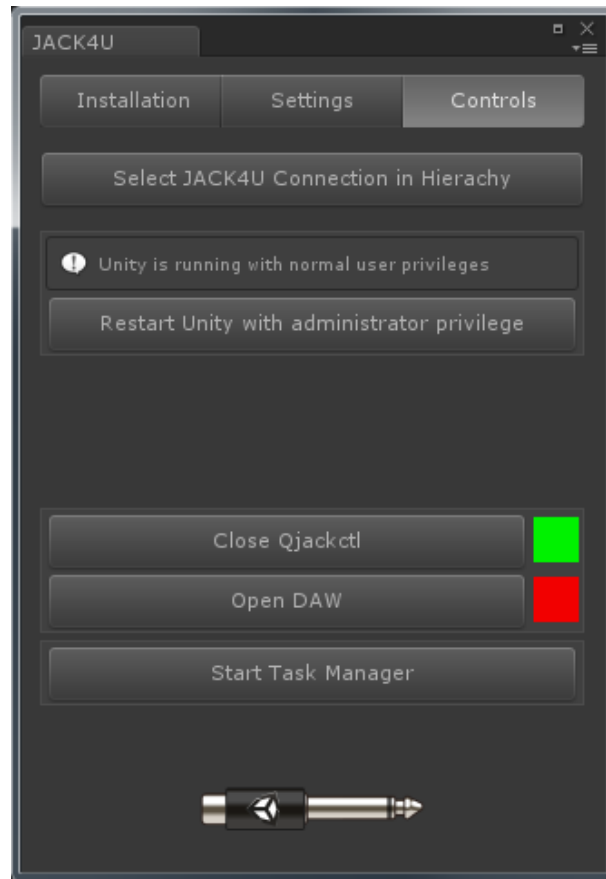


Figure 6.2: JACK4U Editor

- At the top is a button to create a JACK4U Connection. If you have already a JACK4U Connection component in your scene this button is for selecting the GameObject in the Hierarchy.
- There is a restart Unity button and a status message with the current user privileges of Unity. The Microsoft Windows operation system comes with the UAC (User Account Control) security feature ([link](#)). This makes your system more secure but has some impact on the JACK Audio Kit. In some JACK Audio installation guides it is recommended that you should run JACK as administrator if you have connection problems. If you run one of components of the JACK4U as administrator you have to run every process as admin, otherwise the processes can't communicate with each other. When you run Unity with admin privileges all processes that we open from the editor will also elevated to admin rights.
- If you have specified the paths of Qjackctl and your DAW you can open&close these apps from the JACK4U Editor. Keep in mind that this feature only works when we started the processes from the editor as we need a process id. If you start a DAW from the Windows Explorer we don't have this id so we can't communicate with the process. (You can click on the status box of the buttons to bring the selected application into front if its running)

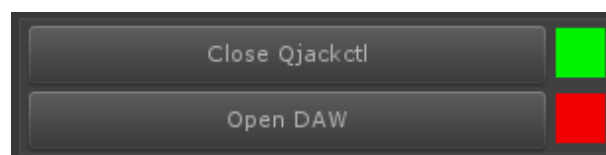


Figure 6.3: open&close application buttons

6.3 JACK4U Connection

- All the settings that are related to connecting to JACK Audio are done on the JACK4U Connection component. It also provides status and error feedback so you can easily monitor the system and better debug.
- To connect to the JACK Audio System we only need one JACK4U Connection in our scene. (If you try to add a second connection you get a prompt that this is not allowed. If you duplicate a GameObject with a JACK4U Connection component it will be destroyed when you enter play mode.)
- At the JACK4U Connection component you configure your connection to the JACK Audio System. These settings should match with the settings you made for the JACK Audio Server with Qjackctl. The main properties you have to administrate are:
 - Sample Rate: Has to match with sample rate of JACK Audio (otherwise the connection is refused)
 - Frames per Buffer: Should match with the Frames/Period settings of JACK Audio
 - Input Channels: As we want to receive a stereo signal this is set almost to two
 - Output Channels: This setting is related to the JACK Audio Out channels you will see in Qjackctl. Normally you will set this to 'none' as Unity routes its audio to the system out per default. You can't circumvent this and this connection is independent from JACK Audio so it will not be displayed in Qjackctl. When you set some output channels here there will be an additional output for JACK created. (The audio signal that comes from the JACK system will be directly looped through, so you can create a daisy chain of your apps.)

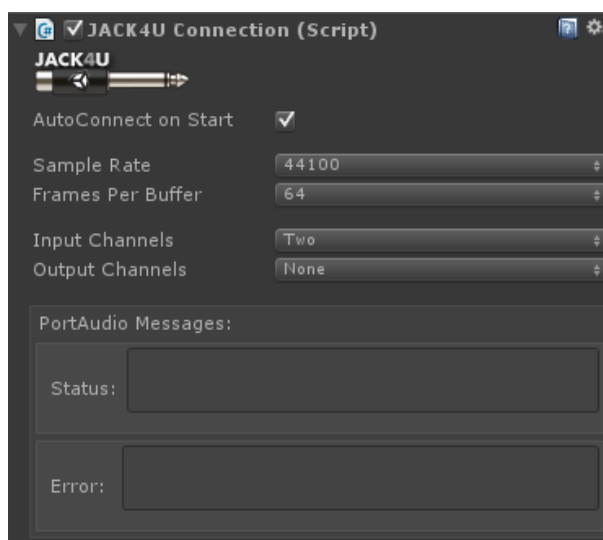


Figure 6.4: JACK4U Connection

6.4 Check your AudioListener in the scene

- The JACK audio stream is injected into the Unity DSP chain via the `OnAudioFilterRead` method. This means that the JACK4UConnection requires an AudioSource component to work (provides the audio of JACK to Unity), but without an AudioListener you won't hear anything.
- Since Unity 5 you can't have a AudioSource and a AudioListener attached to the same GameObject if you use a custom DSP filter script as JACK4U uses. So you can't have an AudioListener on your JACK4UConnection GameObject!
- To get always 100% audio volume you should set the transform of your JACK4UConnection to the same position as your AudioListener. Unity for example adds per default an AudioListener to your main camera so it is best to make the JACK4UConnection a child of your camera.

Chapter 7

Run the system

7.1 Start the JACK Audio Server

- As we want to connect to the JackRouter device the first step is to start the Jackd Server process. Depending on your computer setup this process is already running in the background. In most cases we start JACK with our graphical administration tool Qjackctl.
- It is crucial to understand that if you start Qjackctl you don't start the Jackd server process! (But in the 'misc' settings of qjackctl you have the option 'Start JACK audio server on application startup' so it is started automatically). With the Start/Stop buttons you control the server status by hand.



Figure 7.1: JACK Audio Server running

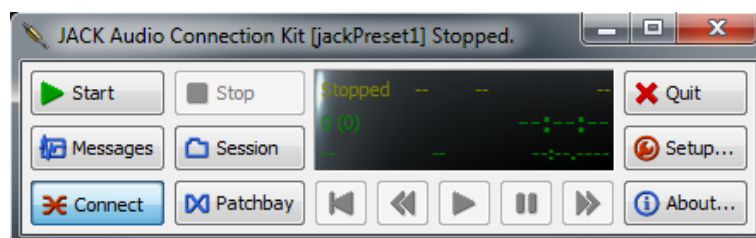


Figure 7.2: JACK Audio Server stopped

- You should also be aware of that if you quit the Qjackctl app doesn't mean automatically that the JACK Audio Server is shut down. A problem for JACK4U on 64 Bit systems for example is that if you close Qjackctl from the JACK4U Editor the Jackd Server process stills runs in the background of your system (Unity can't stop 64 Bit processes).
- If you start/stop Qjackctl from the Windows Explorer you always get prompt that you shut down the JACK Audio Server when you quit Qjackctl.

- To prevent confusion you always should monitor your processes via the Windows Task Manager and check if the 'jackd' process is running.

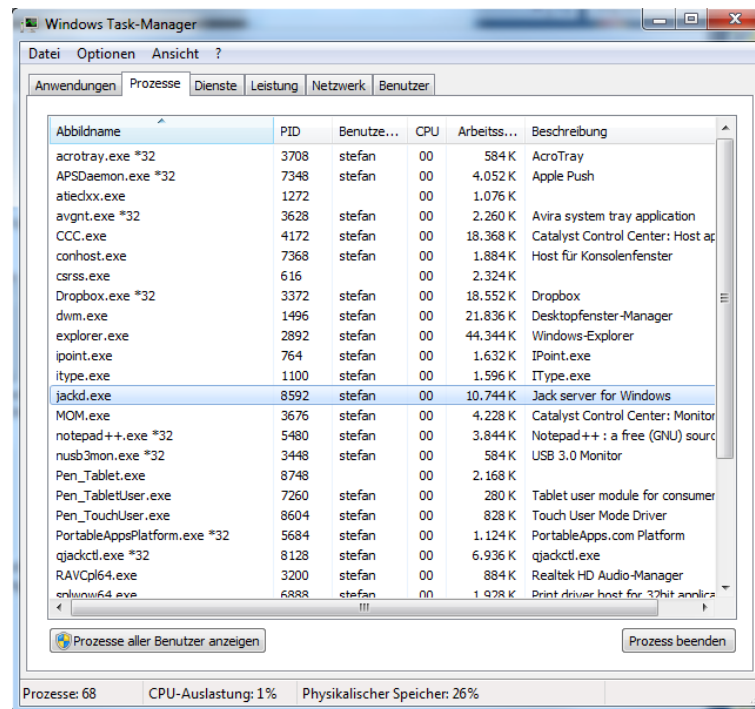


Figure 7.3: jackd process in the Task Manager

7.2 Start your DAW

- When the JACK Audio Server is running we start our DAW program to feed audio into JACK.
- First we have to specify the Audio Device driver type (ASIO) and then we select the 'JackRouter' as audio device.
- In Ableton we have to go into 'Options/Audio' and do our settings

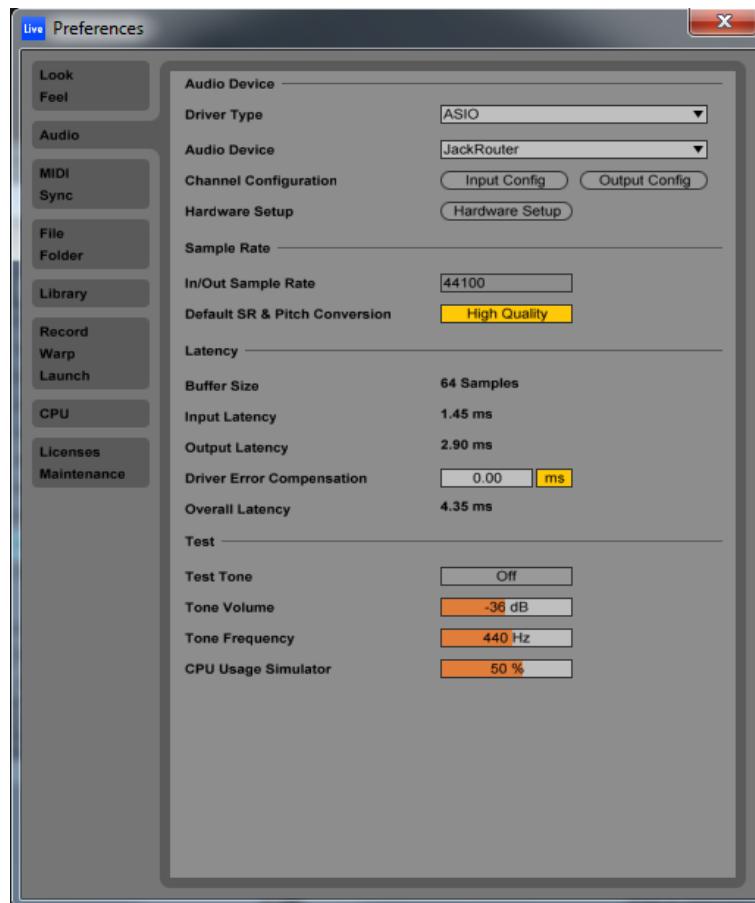


Figure 7.4: Ableton Live settings

7.3 Start JACK4U Connection

- Now we enter the play mode in Unity. If you turned on the Auto connect option in the JACK4U Connection component Unity tries to connect to the JACK Audio system. If any problems arise you get feedback at the PortAudio Messages section.

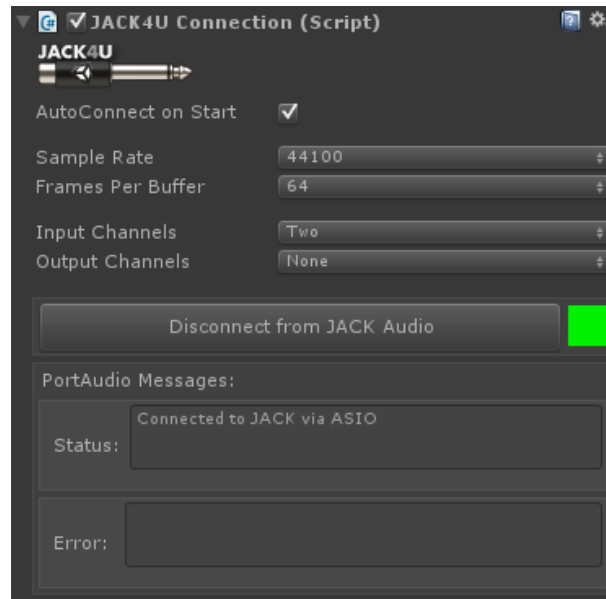


Figure 7.5: JACK4U Connection running

- When a connection is established you can switch to the Qjackctl window and open the Connections window. Here you see all the running apps that are connected to the JACK Audio system. As we running from the Unity editor we should see the Unity app.

- Now we drag &drop a connection from the DAW client to the Unity client to route the audio from the DAW into Unity. You now hear the audio twice: the direct output from the DAW to the system out and the audio out from Unity. This is useful to check how the latency between your DAW source output and the Unity output is.

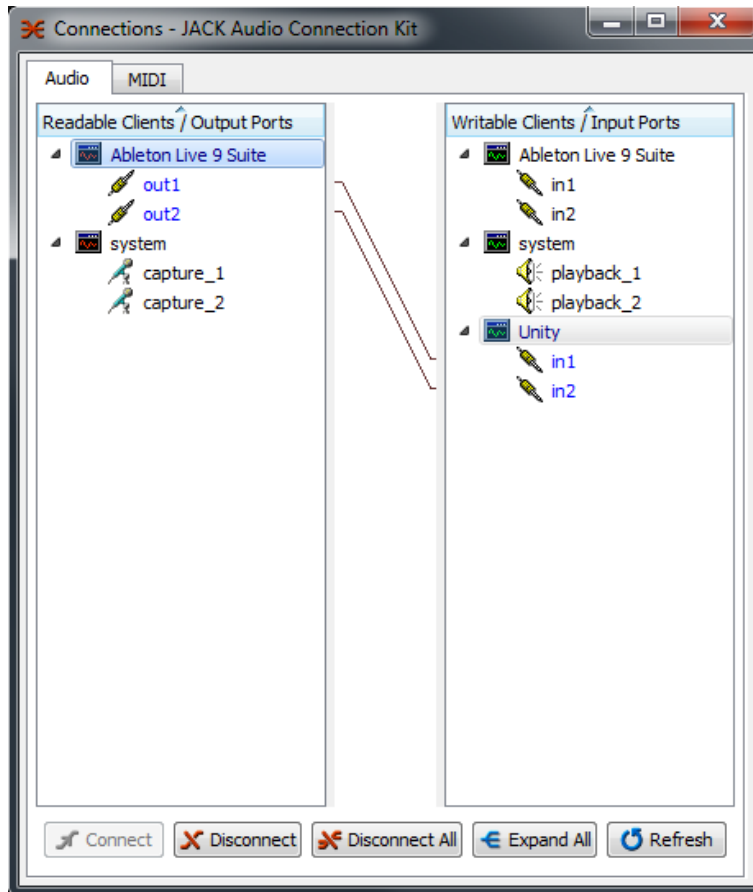


Figure 7.6: Connections

- In general you should turn off the Unity audio output in the audio mixer from Windows when you are in a live performance situation.
- You can save your audio routing with Qjackctl in a Patch file and activate a saved Patch so you don't need to rewire all connection by hand every time.

Chapter 8

Known Issues

- When you stop the JACK Audio Server the connected clients could freeze. You should always stop/disconnect all clients before you stop/shutdown the JACK Audio Server.
- General Problems with 32/64 Bit Systems
 - As most of the actual systems are now 64 Bit we still use a lot of 32 Bit programs. Unity 4 is a 32 Bit program and with this fact there arise some problems with inter- process communication.
 - JackAudio comes in 32 Bit or 64 Bit versions.
 - If you open other application like Ableton(64Bit) from the windows explorer and not from within Unity, JACK4U could not interact with the process like open or close the app.
- If you close Qjackctl from within Unity the JACK Audio Server will stay as process in the background. This has some drawbacks and some advantages. The main problem is that we lost the control to start/stop the jackd process. If you do some changes in your JACK settings they will not update the running Server process (You can check this in the "Messages/Status" window of Qjackctl). It could happen that you try to stop the Server with the Stop button but you still hear you audio. You have to kill the jackd process by hand with the Windows Task Manager to stop the JACK Audio Server. Then you can restart the Server with Qjackctl and everything works as expected. In general it is useful to run the jackd server in the background so you can start & stop your apps that want to connect to JACK without always thinking about if JACK is running. On the other site it could cause confusion when you change parameters or want to stop JACK with Qjackctl in this state and you have to kill the jackd process by hand.

Chapter 9

FAQ

- The JackRouter is not available
 - When you running a 64 Bit version of your DAW you have to register the 64 Bit version of the Jack↔ Router.dll manually.
 - Check if the 'jackd' process is running via the Windows Task Manager.
 - Check in Qjackctl if you have choose the right driver (portaudio)
- JACK4U Connection always terminates when I try to start
 - Is the PortAudio.dll located in the "Assets/Plugins" folder.(Did you moved the "Plugins" folder out of the JACK4U Asset folder?)
- I hear nothing in Unity
 - Are you have started PortAudio in your JACK4U Connection component?
 - Are you made all connections in the Connections/Patchbay Window of Qjackctl?
 - How large is the distance of your UniJackConnection GameObject to the AudioListener in your scene?
- I hear dropouts and glitches
 - Check at your AudioSettings ('Edit/Project Settings/Audio')if your sample rate is matching with the settings in Qjackctl.
 - Check if there running unnecessary processes on your computer.
 - Close all apps that don't need to run
 - Close all background processes that are not crucial for your system.
 - Turn off any Anti Virus program.
 - Test your system with a higher 'Frames per Period'. 128 should work on most of the systems with a sufficient latency.
- I always hear my audio twice
 - Unity routes its audio signal independently from JACK Audio to the system out. When you only want to hear the audio from your DAW you have to use the Audio Mixer from the Windows operation system to turn down the sound from Unity.
 - Otherwise you can disable the connection from your DAW to the system out in Qjackctl. Then you will hear only the audio signal with the latency from Unity.

Chapter 10

History

Version 1.1 - 2015.30.07

- Unity 5 support
 - PortAudio plugin structure changed to x86/x86_64
 - added a 64 Bit version of the PortAudio.dll
 - removed the options for Sample Rate, DSPBuffer and DSP Buffers as this is now handled via the AudioSettings from your Unity Project
 - The AudioListener is now attached to the main camera as you can't have an AudioSource and an AudioListener on the same GameObject if you use a custom DSP filter script.
- Documentation update

Version 1.0 - 2014.06.27

- Initial release

Chapter 11

Credits

- **ASIO4All** by Michael Tippach
- **JACK Audio Connection Kit** is provided by the JACK team. (Original authors: Paul Davis, Stéphane Letz)
- **JACK4U** uses a modified version of **PortAudioSharp** written by Riccardo Gerosa
- **Generic Ring Buffer Class** by Simon Wittber

Chapter 12

Links

- **ASIO4ALL:** <http://www.asio4all.com/>
- **JACK Audio:**
 - <http://jackaudio.org/>
 - <https://github.com/jackaudio>
- **PortAudio:** <http://www.portaudio.com/>
- **PortAudioSharp**
 - Original version <https://code.google.com/p/portaudiosharp/>

Chapter 13

Support

If you need support or have any question/suggestions please contact us.

- Website: <http://jack4u.monoflow.org/>
- Email: info@monoflow.org
- Unity Forum: <http://forum.unity3d.com/threads/253572/>

Copyright © 2014 by



