

tpf-tool-advanced

This is the advanced equipment of the tpf-tools for macOS 10.14+ and above.

macOS "big sur" only Intel!

Important:

Make sure that you have installed JackOSX! If not, go to the `<install_JackOSX>` installer download. Be sure to take the 'TPF-Tool JackOSX Installer.command' to do this. After the JackOSX installation, you will need to reboot your Macintosh. If everything worked, you should now create an aggregate. How to do this is described here: <https://support.apple.com/en-us/HT202000>.

installer tpf-tool-advanced

The 'tpf-tool-advanced' corresponds to the previous basic idea of the tpf-tools, such as the 'tpf-tool-basic' and the 'tpf-tools-intermediate'. Advanced' stands for modularly structured application possibilities. (simple to quite complex settings).

The most important addition to tpf-intermediate is the possibility to create a 3D audio mix, for headphones (binaural), as well as one for loudspeakers in Ambisonics.

Audio and video are independent of each other. Ultragrid can, but does not have to be used.

Download the 'tpf-tools-advanced' @ GitHub:

<https://github.com/joambi/>

Install the tpf-tools-advanced:

- download the DMG
- open the DMG
- open the folder 'installer-tpf-tools-advanced'
- right-click the 'install-tpf-tools-advanced.command'
- enter your admin-PW
- follow the installer instructions.

The Installer installs the following components:

Applications:

- tpf-party.app (v1.1-beta4)
- tpf-client.app (v1.1-beta5)
- Ardour6.6.app
- uv-qt.app (Ultragrid 1.6+)

Ambisonics & Binaural Audio-Plug-Ins (free) installed in: -> /Library/Audio/Plug-Ins

- ICST Ambisonics Plugins -> ambisonics
- dearVRAMB/MICRO(free) -> binaural
- Sennheiser AMBEO VR Mic A-B Converter for the AMBEO-Mic
- IEMPluginSuite_v1.11.1 -> reverb

Scripts:

- tpf-runner.command
- tpf-save-audio.command
- tpf-load-audio.command

Folders:

- tpf-settings
- tpf audio-connections
- ardour-templates
- tpf-video

tpf-settings:

- tpf-basic.txt
- tpf-default.txt
- tpf-advanced.txt
- tpf-ambisonics (for Headphone & Ambisonics mix)

tpf-audio-connections:

- audio-basic.xml
- audio-default.xml
- audio-advanced.xml

ardour-templates:

- tpf-advanced-basic.ardour
- tpf-advanced-default.ardour
- tpf-advanced-advanced.ardour

tpf-video:

- video-runner-camera01.command
- video-runner-camera02.command
- video-selftest-camera01.command
- video-selftest-camera02.command

tpf-uninstallers:

- uninstaller-JackOSX.command
 - uninstaller-tpf-tool-advanced.command
-

Start the tpf-tools-advanced:

After the installation process, you can start the tpf-tools-advanced basic session.

1. double click on the file "04tpf-tool/runner.command".
2. follow the instructions of the tpf-runner.

Sources:

1. JackOSX.0.92b3 to /Applications/Jack -> <https://ccrma.stanford.edu/software/jacktrip/osx/JackOSX>
2. JMess to /usr/local/bin -> <https://github.com/jacktrip/jmess-jack>
3. Ardour.app -> <https://ardour.org>
4. Ultragrid -> <http://www.ultragrid.cz>

5. TPF-TOOLS -> <https://gitlab.zhdk.ch/TPF>
 6. IEMPluginSuite_v1.11.1 -> <https://plugins.iem.at/>
 7. dearVRAMB/MICRO.pkg -> <https://www.dearvr.com/products/ambi-micro>
 8. Sennheiser AMBEO VR Mic A-B Converter <https://de-ch.sennheiser.com/ambeo-abconverter>
 9. ICSTAmbiPlugins2.1.0osx.pkg -> <https://bitbucket.org/christianschweizer/icst-ambisonics-plugins/>
-

About TPF-Client

tpf-client is a low-latency audio transmission software-based on the jacktrip protocol and built-in Pure Data.

It tries to overcome some limitations that are often encountered when using the traditional jacktrip commandline utility:

- None of the endpoints are required to have a public IP address. All clients can be behind a firewall.
- The tpf-client reduces complexity when configuring a session with many endpoints.

The client registers itself to a tpf-server which keeps track of the connected clients. Thus the clients learn about the other clients and establish a jacktrip audio connection to their peers either by routing the packets through the tpf-server or directly to the peer by employing a technique called UDP hole punching (https://en.wikipedia.org/wiki/UDP_hole_punching).

You can download the client from:

<https://gitlab.zhdk.ch/TPF/tpf-client>

The client is pre-configured to connect to a tpf-server running on telematic.zhdk.ch (TCP-Port 3025) which is supposed to be running anytime in the area of Zurich, Switzerland. The tpf-server software can be found here:

<https://gitlab.zhdk.ch/TPF/tpf-server>

Depending on the locations of the endpoints it is advised to run a server closer to one of the endpoints in order to keep transmission latency low.

Prerequisites

Make sure to get the latest Pure Data from:

<https://puredata.info/downloads/>

You need the following externals to run tpf-client

* iemnet * osc * slip

You can install externals through the Pd menu: 'Help' -> 'Find Externals'

Running the client

To run the client, open the patch tpf-client in Pure Data. Typically, you run Pd with jack as audio backend, so that you can send audio from and to the tpf-client to other software. When running from the command- line, the recommended parameters are:

```
pd -rt -jack -in channels 8 -out channels 65 -nojackconnect -jackname tpf-client -open tpf-client/tpf-client.pd
```

Before connecting to the server, make sure to set a name in <Location> and all parameters. The parameters sr (sample rate) and bs (block size) must be the same for all clients. The first connecting client defines the values of those parameters for the session. After configuring those parameters the client can connect to the server by clicking the top left button. Blue indicates the connection is established. Red indicates that some errors occurred. Check the message for the reason. Reasons for connection failure include sample rate or name conflict (the configured name is already in use by someone else).

Once connected, other endpoints appear in one of the 8 rows. In order to establish audio transmission, either one side needs to initiate the connection by clicking the black square on the left and the other side has to confirm. A flashing button indicates the other side is waiting for confirmation. Once the audio connection is established, the corresponding button turns blue and the numbered squares indicate the number of received channels and the level of each. The numbers on the channel indicators correspond with the numbers in the qjackctl connection dialog.

By double-clicking the left square, a request for a connection using UDP hole punching is sent. When confirmed, a direct transmission between endpoints without using the server as UDP proxy is established. However, this feature is considered experimental and can't be used in certain network environments.

Ready-to-use macOS app

If your computer runs macOS, you may download a self-contained application bundle from:

<https://gitlab.zhdk.ch/TPF/tpf-client/-/releases/>

Bugs

* tpf-client crashes Pd when receiving a stream with 3 channels

For any bug, issue, or suggestion, please open an issue [here](#).

Authors

* Roman Haefeli roman.haefeli@zhdk.ch * Johannes Schütt johannes.schuett@zhdk.ch * TPF-Team @ZHdK

License

GPL 3.0 (see LICENSE.txt)