



## Overview

**TUIO** is an open protocol for the communication of data from devices like a multi-touch display, an interactive surface, or a computer vision-based motion tracker. The sorts of information it describes includes touch events and object locations, much like with the Microsoft Surface, as per the video [here](#). TUIO uses a client/server model for communicating this data, so any program that is a TUIO client can receive touch or object data from any TUIO server.

TUIO is based on [Open Sound Control](#) (OSC) a specification for sending messages interactively, which while originally intended for sharing music and performance data, has been increasingly used for other kinds of non-musical data.

OSC transmits information across standard IP networks and the internet using [User Datagram Protocol](#) (UDP), so it's supported by a lot of systems, including Windows, Mac OS X, and Linux. Any system that supports OSC can also support TUIO, and consequently, there are many TUIO client and server programs across these platforms. Further, because TUIO uses UDP for communication, a TUIO server and client needn't be running on the same computer, or even on the same local network, though both running on the same computer probably covers most usage cases.

With the UPDD-to-TUIO server program, any application that is a TUIO client and receives TUIO touch data can be controlled with a UPDD driver supported touch-screen, thus expanding the number of multi-touch enabled applications that the UPDD can support, the schematic being:

Touch hardware > UPDD driver > UPDDtoTUIO.app > TUIO client application > end user interaction

The main limitation is that UPDD-to-TUIO does not support object data, as the UPDD only tracks touches on a multi-touch display. However, the main use of TUIO is to communicate multi-touch data, so most TUIO applications will be usable with the UPDD in spite of this limitation. If it should become necessary for the UPDD to support TUIO objects as well, an interface could be created to allow for the creation and positioning of virtual objects, much like the TUIO simulators that exist for testing and demonstration purposes.

Currently UPDD-to-TUIO server exists as a stand-alone application that needs to be compiled for each operating system but will shortly be integrated into the main driver so will become part of the standard driver offering for each OS. The first version of this UPDD-to-TUIO server has only been made available for Mac OS X. Other versions will be made available if required before it is integrated into the driver.

Given that TUIO servers communicates using UDP then the only configuration necessary for the TUIO server is the IP address, port number and message packet size. However, since most usage cases for a TUIO server is to have it running locally on the same computer that the TUIO client programs are running on, a reasonable default configuration is for it to use localhost with the default TUIO port number and packet size and these are currently hardcoded in the standalone server application. Given that advanced users may require a different configuration this will be catered for in the full driver integration of the TUIO server.

## Mac OS X

The UPDD-to-TUIO server for the Mac OS X is available [here](#).

To utilize this service in the Mac environment simply run UPDDtoTUIO.app, which is in located in the .zip file) on any Mac with the UPDD driver installed. There are currently no configuration options.

Once the TUIO server is running then any Mac application that is a TUIO client should respond to touches. In our tests we utilised the following Mac TUIO applications:

For the basic Java TUIO client:

1. Go to <http://www.tuio.org/?software>
2. Click the link under "TUIO Client Reference Implementations" named TUIO\_JAVA.zip. TUIO\_JAVA.zip should download automatically
3. Unzip the file 4. In the TUIO\_JAVA folder, open TuioDemo.jar. A window should appear with the title "TuioDemo" which should display touches.

For TuioPainter:

1. Go to <http://www.patriciogonzalezvivo.com/blog/?p=238>
2. Click the link that reads [ Download the Mac OSX App ]
3. Unzip TuioPainter.zip
4. in the TuioPainter folder, open tuioPainter.app. It should allow you to paint via touches.

Note: This program did crash on launch for one of the two Macs we utilized in our tests.

For PyMT:

1. Go to <http://code.google.com/p/pymt/downloads/detail?name=pymt-0.5.1-osx.dmg&can=2&q=>
2. Click the download link named pymt-0.5.1-osx.dmg
3. Open pymt-0.5.1-osx.dmg
4. In the PyMT finder window that opens, open PyMT.app
5. In the same window, open the Examples folder
6. In the Examples folder, navigate into the folders for any of the examples you want to launch. A simple one is in Examples/apps/paint 7. Drag the .py file in this folder (or any of the other example folders) on to the PyMT icon in the

Dock. (NB: ignore the file\_\_init\_\_.py)

8. The example should launch and respond to your touches.

### **Windows and Linux**

The UPDD-to-TUIO server will be made available in these operating systems as and when required. It is our intention to fully integrate the TUIO server into the driver removing the need for a separately compiled application but the standalone module will be made available for these OS if required before integration is completed.

### **Contact**

For further information or technical assistance please email the technical support team at [technical@touch-base.com](mailto:technical@touch-base.com)