

Commander

UPDD Commander offers advanced multi-touch support along with TUIO protocol support for TUIO client applications and is available for macOS and Windows.

This document describes how to use and configure UPDD Commander functions.

Commander is compatible with UPDD V6 release 6.0.559 and above and UPDD V7.n.n

UPDD Commander defines a default set of gestures and actions that apply across the board for all applications, but also offers the ability to associate a tailored set of gestures and actions with individual applications so that gesture sets can be defined at the application level.

Commander software will either be activated automatically as part of the UPDD driver installation or by invoking 'UPDD Commander' from the UPDD application folder;

UPDD Ver	OS	Location
V6 & V7	macOS	/library/application support
V6	Windows	c:\program files (x86)\upd
V7	Windows	c:\program files\updd

Once loaded, the Commander settings dialog contains a setting to indicate that Commander should [automatically load at system start](#).

The vast majority of the UPDD Commander documentation was written for the MacOS implementation of the UPDD Commander software, hence screenshots are all taken from MacOS. In macOS UPDD Commander implements multitouch gesture support for touchscreen devices that is normally associated with multitouch trackpads whereas Windows natively supports the touch interface and gestures so UPDD Commander offers slightly modified functionality when running in Windows as discussed [here](#).

MacOS

When the Commander software is installed as part of the driver installation, start up items are created for all user accounts so that Commander will be running when switching users. Under macOS the Commander process is installed such that it will be invoked at each system startup.

Commander

The UPDD Commander daemon (background) process interacts with the driver and receives all touches from all the touch devices and determines the touch event being performed (i.e. tap, press, gesture) and processes these touches as dictated by the Commander settings.

Gestures are performed on the touch screen exactly as they are on a multitouch trackpad. The action associated with each gesture can be defined in the [touch function dialog](#). To utilize all available gestures you will need to use a multitouch touch screen that supports up to 5 touches, otherwise you will be restricted to the gestures that relate to the number of stylus supported on the touch screen. A number of videos have been posted on the web from end users, such as this one [here](#).

When Commander is handling the system pointer, the driver's own posting of single touch data into the system is disabled and Commander receives all touch from the touch device via the driver's API interface. If Commander is quit, then touch will revert to single touch via the driver's own system interface.

Depending on the Commander configuration, it will post all gesture and touch data as OS X native touch events, as TUIO touches and the UPDD API interface (allows UPDD Client application to receive gesture information), regardless of whether there is a gesture action being performed, as the gestures are more or less separate from the touches themselves.

Where individual gesture processing is not required (you do not require the gesture to trigger any actions for the performed gestures on the touch screen), you can set the gesture action to "No action", and the desktop and applications will still be able to receive the individual gesture and touch data.

There is also a setting to request that [touch data is posted into the system as tablet events](#), allowing a touch device to be used with Inking and other tablet-related features. It only affects mouse events produced by Gestures, though, since in OS X a tablet event is also a mouse event.

UPDD Commander also supports pen devices that present proximity, left click (via nib), right click, barrel button, pressure and eraser features as described [here](#).

Windows

When the Commander software is installed as part of the driver installation, the installation

Commander

can be configured to start up for all users. If it is not started automatically as part of the installer there is a global setting in the UPDD Commander dialog to 'Run at startup'

The UPDD Commander daemon (background) process interacts with the driver and receives all touches from all the touch devices and determines the touch event being performed (i.e. tap, press, gesture) and processes these touches as dictated by the Commander settings.

Gestures are performed on the touch screen exactly as they are on a normal Windows monitor or multitouch trackpad. The action associated with each gesture can be defined in the [touch function dialog](#). To utilize all available gestures you will need to use a multitouch touch screen that supports up to 5 touches, otherwise you will be restricted to the gestures that relate to the number of stylus supported on the touch screen.

When Commander is handling the system pointer, the driver's own posting of touch data into the system is disabled and Commander receives all touch from the touch device via the driver's API interface. If Commander is quit, then touch processing will revert to the driver's own system interfaces.

Depending on the Commander configuration, once the gesture is determined and any related action performed the touch data may then be posted to the OS as HID touch, pen or mouse data, as TUIO touches and the UPDD API interface (allows UPDD Client application to receive gesture information), regardless of whether there is a gesture action being performed, as the gestures are more or less separate from the touches themselves.

Where individual gesture processing is not required (you do not require the gesture to trigger any actions for the performed gestures on the touch screen), you can set the gesture action to "Post data to Windows", and the desktop applications will still be able to receive the touch data.

Touch-Base Support

<http://support.touch-base.com/Documentation/50592/Commander>