

Kernel interfaces

The driver posts touch data into the system via Xtest or [uinput](#) interfaces. The active system interface is described [here](#).

You can select the interface to use in the [UPDD Console](#), Touch Interface dropdown.

The Xtest interface, which is an X11 extension, offers mouse button presses and mouse movement functionality, allowing for mouse emulation via touch.

Uinput offers a user mode bridge to the Linux input subsystem and allows for the posting of both single and multi touch data events. This interface requires uinput support being part of the Linux kernel and should be available for most modern distributions.

Both interfaces allow for a [right click to be generated](#) if required.

UPDD for Linux currently ships with touch mode = uinput; single touch mode by default.

Depending on your touch requirements these selections should be considered.

- If native multi touch support is needed you must use uinput + multi touch mode. TUIO Server and/or UPDD API multi-touch are separate considerations and relate to application interfaces.
- If multiple X displays are used (i.e. anything other than DISPLAY :0) xtouch mode must be used.
- If multiple monitors (all on DISPLAY :0) are used then uinput or xtouch can be used (assuming that multi touch is not required in which case the uinput dependency takes precedence).
- **Important note:** there is currently no reliable support for uinput (single or multi touch) on multiple X displays. We have seen this work on some systems but it will have to be assessed on a case by case basis.

X interface

When using X the driver needs permission to open an interface to X.

In normal operation [UPDD Daemon](#) will execute the command 'xhost SI:localuser:root' which grants permission to processes already having root privileges on the localhost system to access the X session in which UPDD Daemon is running.

Kernel interfaces

We consider this to be reasonably secure; but in a secure environment if you need to use a different policy then you can set the updd global setting "xhost.enable" to define an alternate policy, e.g.

```
upddutils nodevice set xhost.enable "+"
```

will execute the command

```
xhost +
```

when UPDD Daemon starts.

If UPDD Daemon is not running for any reason you will need to grant this permission in some other way.

A quick way to do this for testing purposes is to enter the command

```
xhost +
```

or

```
xhost SI:localuser:root
```

in a terminal.

The command 'xhost +' turns off X access control allowing anyone on the network to connect to the system therefore this is not recommended in situations where this might be considered a security breach.

The default 'xhost SI:localuser:root' or an equivalent is recommended where security is a concern.

uinput interface

Default is uinput unless Xtest is specifically defined as the [active system interface](#) or the uinput component does not exist. For uinput, the driver creates either a single touch

Kernel interfaces

(default) or multi-touch virtual device and a separate virtual device to pass right click requests. Using the primary virtual touch device to post both touch data and right clicks caused issues in some distributions which were not seen when using separate devices.

Using the `lsinput` command you can view the UPDD virtual input device(s). Depending on the distribution in use `LSinput` may need to be installed. Run '`sudo lsinput`' to list the devices:

Virtual device for posting single touch events

`/dev/input/eventnn`

bustype : BUS_VIRTUAL

vendor : 0x1

product : 0x1

version : 1

name : "updd single touch"

bits ev : EV_SYN EV_KEY EV_ABS

Virtual device for posting multi-touch events

`/dev/input/eventnn`

bustype : BUS_VIRTUAL

vendor : 0x1

product : 0x1

version : 1

name : "updd multi touch"

bits ev : EV_SYN EV_KEY EV_ABS

Regarding the UPDD single touch OR UPDD multi touch devices, only one will exist depending on the value of the [setting uinput.multi_touch](#).

The single touch device is useful on systems that support uinput but do not recognise or have problems working with a multi touch device.

The multi-touch device is used to post both single and multi-touch data such that both types of system requests are satisfied. Single touch data is posted in a different manner to multi-touch data (even when a single touch is in use) so all touches result in two posting on the uinput event interface to satisfy both single and multi-touch system requests.

In practice uinput support varies even on some recent distributions, so on some distributions the multi touch device does not work at all and some do not support standard single touches being posted on multi-touch devices. For this reason UPDD allows [uinput usage to be adjusted](#) or just the [direct X interface to be defined](#).

An application that needs to receive touch events should open one of the named devices as appropriate, directly or indirectly via touch event APIs. Additionally UPDD right mouse

Kernel interfaces

button if right mouse click events are required.

In normal use no setting changes should be needed; however on some older systems with incomplete multi-touch support if the uinput interface does not work correctly the settings `uinput.single_touch` and `uinput.multi_touch` should be [adjusted](#). Try turning off one or both of these via the [Command Line Interface](#), Set command.

Touch-Base Support

<http://support.touch-base.com/Documentation/50458/Kernel-interfaces>