

Data manipulation

The driver receives stylus touch data from the hardware interface and posts the coordinates and other information, such as pen state (down / up) etc into the operation system.

The stylus touch data is dispatched into the system as dictated by the speed of the data sent by the touch device.

In most cases this process is transparent and just works and the data does not need to be manipulated. However, the data can be manipulated or altered by various driver settings and configurations as follows:

Setting / configuration	Description
Mouse interface	Only data from the 1st stylus is posted into the mouse interface.
Multi-touch interface (HID/Uinput/Gestures)	Data from all stylus are posted into the interface.
Windows UPDDVH (HID)	With UPDD 6.0.388 and above for any stylus that is in contact, if 12ms elapses without seeing a packet, the driver injects a dummy packet into the data stream to satisfy the data rate expectations of the Windows HID interface. Delta mode controllers (no data sent when stylus held stationary) or slow controllers (slow data packet rate) can result in Windows generating a pen up in these cases thus preventing a long touch, say to generate a right click.
MacOS Gestures	If ' Click nearest UI element ' is enabled the gesture software will adjust the touch to the nearest UI element if the actual touch is close to a UI element.

Data manipulation

<code>assisted_double_click.time_ms</code>	Assisted double click assists double clicking by generating a second touch at the same location of the previous one. 2nd click needs to be within a certain time and distance of 1st click.
<code>contact_timeout_secs</code>	<p>If set to a non zero value any contact (touch) longer than the specified number of seconds is cancelled (forced lift) and ignored until the contact ends. Other contacts are unaffected.</p> <p>Depending on circumstances this can be useful to filter out ghost touches. It will only be effective where the ghost touches are more or less constant.</p>
<code>deglitch_cal_points</code>	<p>If enabled (both settings >0) then if the co-ordinates within a touch packet differ from the previous packet by more than the <code>cal_points</code> specified then the touch is considered a glitch and ignored, based on a range of 0-65535 X and Y.</p> <p>The number of packets define how many packets are received with the 'glitch' co-ordinates before the glitch function is reset.</p> <p>i.e A touch device is occasionally sending out a single packet of incorrect co-ordinate data, setting the <code>Cal_points</code> to 6000 and <code>packets</code> to 2 will ensure that</p>
<code>deglitch.packets</code>	

Data manipulation

filter.low_pass.gain	<p>any single packet with co-ords that differ more than 6000 and only last for one data packet will be ignored.</p> <p>When set to a non-zero value low pass filtering is enabled.</p> <p>Low pass filtering provides smoothing of input, removing jitter from inputs subject to noise or similar distortion.</p>
	<p>LPF will introduce some lag, especially at higher values; but at low values it provides smoothing with minimal loss of fidelity and lag.</p>
	<p>The appropriate value will have to be determined by experiment, but typically values of around 5 will give good results.</p>
	<p>Values over about 20 will tend to give extreme lag.</p>
	<p>The feature is multi touch aware; operating on all contacts independently</p>
ignore_multi_touch	<p>If true (1) then multi touch events are discarded - only processes first touch stream.</p>
liftoff_time_ms	<p>When set to a non-zero value a lift off event will be injected in the case that a touch event is not seen within this threshold (in milliseconds)</p>

Data manipulation

following a previous touch event. This is only required for devices that do not generate a pen up event at stylus liftoff as the driver relies on this to register a pen up. This is only required for devices that do not generate a pen up event at stylus liftoff as the driver relies on this to register a pen up. Should be set to 0 for delta mode devices. See [here](#) for further info.

sample_rate

Specifies the percentage of position report packets that are passed to the OS and the UPDD API. Useful if it is considered that the OS is being overwhelmed with too much touch data.

Note that the algorithm works on blocks of 10 packets; so for example a setting of 20 means that 2 of every 10 packets are processed.

Consequently only multiples of 10 are useful, for other values, the value is rounded down to the next multiple of 10.

synthesize.delta_mode

If set to on (1) a device will work in synthesized delta mode. Only touch packets that represent movement or a change in a contact state are delivered to the operating

Data manipulation

touchdown_filter	<p>system and the UPDD API.</p> <p>If set to a non zero value this causes the first N packets in any touch that represent a touching state to be treated as not touching.</p>
upddvh.keep_alive	Windows 10 will reject, or not fully process, incoming touch data that falls below a certain packet rate threshold.
upddvh.keep_alive_interval_ms	<p>For 'slow' touch devices that are not transmitting data at the required data rate the driver can be configured to inject touch packets into the data stream to satisfy this Windows 10 requirement.</p> <p>Two global settings control the keep alive function:</p> <p>upddvh.keep_alive = 0 (off) or 1 (on) – default = 1.</p> <p>upddvh.keep_alive_interval_ms - Specifies the time interval upon which to inject keep alive packets when using upddvh.keep_alive - default = 15 – if the gap between data packet is >15 ms we will inject dummy packets. The function is cancelled at the end of a stylus contact.</p> <p>A driver reload is needed for any updates to these setting to take affect.</p>

Data manipulation

<http://support.touch-base.com/Documentation/50522/Data-manipulation>