

## Data injection

The driver can process touches from both physical hardware and virtual devices whereby the touch data is injected into the driver via the API function call `TBApiInjectTouch`. This interface is useful to support a device where the hardware is being controlled by other software. Similarly an application can be used to drive the system pointer without a physical device being used.

This API will function as long as an active device is configured. In cases whereby UPDD is not supporting an actual physical device, and therefore no active device exist, UPDD also supports virtual devices to allow for an active device to exist for the purposes of this API.

To use a virtual device you will first need a driver with support for a virtual device. For want of a better name we have named the virtual device 'Generic, Virtual'.

Then use the [adddevice option](#) in `upddutils` to add an instance of such a device, e.g.  
`upddutils adddevice 1`

A new device setting is used in conjunction with virtual devices. "virtual\_device" will by default have a value of 1 (meaning true).

The UPDD Status will show a virtual device as connected (OK) in this case. A program can set this value to 0 (meaning false) to have the status show disconnected (NOK).

This option is supported for all controller types but only a value of 1 has an effect, this forces the device into a connected state so that devices being used in this way are listed as working in user interfaces and to programs using the API.

Please note that certain aspects of the active controller need to be considered when injecting data, one being the defined co-ordinate range, which in the case of the Generic, Virtual device is 0 - 1024 for both X and Y. These values are held in settings `calx0`, `calx1`, `caly0` and `caly1`. For other devices they can be viewed using the `upddutils [device n] get "*" command`.

### **Posting raw data packets**

In addition to the above API to inject complete touch data it is also possible to post raw touch data to an active touch device using the API `TBApiPostPacketBytes` in the raw data format configured for the physical device.

## Data injection

We use this API to playback captured data using upddutils recorddraw to help troubleshoot reported touch issues for a given controller configuration.

### Example TbApiInjectTouch code

The code sample below is a test function that can either draw a circle or simulate a 4 point calibration sequence - this is a code fragment and not a complete solution!

```
int DoSelfTest(constvector<string>& aArgs)

{
    if (aArgs.size() != 3)
        return(Usage());
}

if (aArgs[2] != "circle" && aArgs[2] != "calibrate")
{
    return(Usage());
}

int t;

sscanf(aArgs[1].c_str(), "%d", &t);

Sleep(t * 1000);

double x, y;

if (aArgs[2] == "circle")
{
    for (int a = 0; a < 360; a++)
    {
        double r = a * 3.141592 / 180;

        x = 512 + (cos(r) * 300);
```

## Data injection

```
y = 512 + (sin(r) * 300);
TBApiInjectTouch(gSelectedDevice, (int)x, (int)y, 0, 1);
Sleep(20);
}
TBApiInjectTouch(gSelectedDevice, (int)x, (int)y, 0, 0);
return(0);
}

// calib point 1
int margin = 1023 / 10;
for (int n = 0; n < 50; n++)
{
    x = margin;
    y = margin;
    TBApiInjectTouch(gSelectedDevice, (int)x, (int)y, 0, 1);
    Sleep(20);
}
TBApiInjectTouch(gSelectedDevice, (int)x, (int)y, 0, 0);
Sleep(1000);

// calib point 2
for (int n = 0; n < 50; n++)
{
    x = 1023 - margin;
    y = margin;
    TBApiInjectTouch(gSelectedDevice, (int)x, (int)y, 0, 1);
    Sleep(20);
}
```

## Data injection

```
TBApiInjectTouch(gSelectedDevice, (int)x, (int)y, 0, 0);  
Sleep(1000);  
  
// calib point 3  
for (int n = 0; n < 50; n++)  
{  
    x = margin;  
    y = 1023 - margin;  
    TBApiInjectTouch(gSelectedDevice, (int)x, (int)y, 0, 1);  
    InternalApiSleep(20);  
}  
TBApiInjectTouch(gSelectedDevice, (int)x, (int)y, 0, 0);  
Sleep(1000);  
  
// calib point 4  
for (int n = 0; n < 50; n++)  
{  
    x = 1023 - margin;  
    y = 1023 - margin;  
    TBApiInjectTouch(gSelectedDevice, (int)x, (int)y, 0, 1);  
    Sleep(20);  
}  
TBApiInjectTouch(gSelectedDevice, (int)x, (int)y, 0, 0);  
return(0);  
}
```

Touch-Base Support

<http://support.touch-base.com/Documentation/50441/Data-injection>