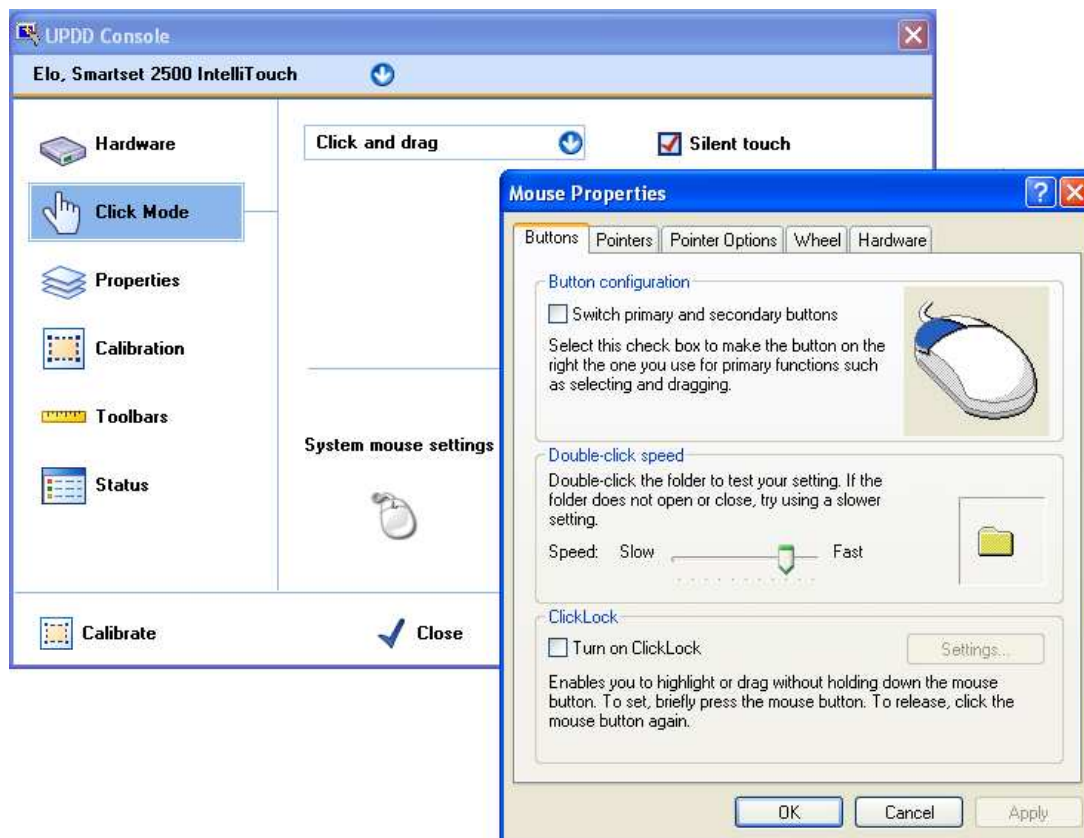


Some features and functions are referenced in the UPDD documentation that require further explanation. Here it is!

Double Click issues

Double clicking on a touch screen relies on double click related settings set to a tolerance that allows a touch screen to be used to generate the double click. In Windows, for example, there are 3 such settings, DoubleClickSpeed, DoubleClickWidth and DoubleClickHeight. These settings establish the width of the rectangle that Windows uses to detect double-clicks. If sequential clicks are positioned within the rectangle defined by DoubleClickHeight and DoubleClickWidth, and occur within the time specified by DoubleClickSpeed they are interpreted as double-clicks. Otherwise, they are interpreted as sequential single clicks.

When UPDD is installed it sets these internal settings (for the user performing the install) to values that cater for finger double click usage. In our old version 3 driver dialog we allowed these system setting to be changed. However, our version 4 dialog calls the system's mouse dialog to change double click settings, as seen below:



Most of these mouse dialogs allow the double click speed setting to be changed and this must be a slow (ish) setting to allow for touch double clicks. Too fast a setting will prevent double clicks. The other related settings can be found in the registry at HKCU\Control Panel\Mouse. The Width and Height default settings are 4 pixels and the speed is 500 milliseconds. UPDD sets these settings to 32 and 370 respectively.

New user issue

If a new user logs in they will get the default Windows double click settings which may make it difficult to double click. With UPDD version 4.1.6, build 1156 and above, the [UPDD daemon task](#) monitors the users. When a new user logs in it automatically sets these settings.

Settings changed by 3rd party software

There have been cases of double click difficulties following the installation of mouse drivers that have set these settings to values that prevent touch double clicks requiring the setting to be reset to 'touch' values.

Disabling double click

This is not a driver issue but a system issue. If touch double click is to be disabled we suggest that the settings be set such that it is almost impossible to achieve a double click with touch. Try Height and Width = 1 and Speed = 200 – see below.

Register settings

Should you need to manually set the registry for the Double Click Settings these lines can be copied to a .reg file and double clicked to update the registry settings:

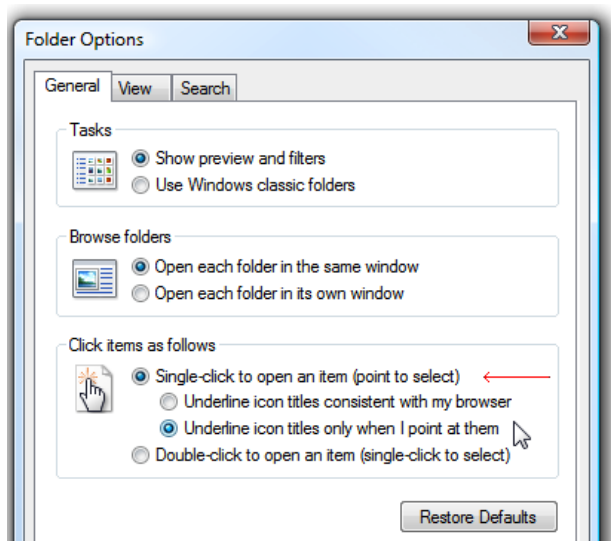
```
Windows Registry Editor Version 5.00
[HKEY_CURRENT_USER\Control Panel\Mouse]
```

```
"DoubleClickHeight"="32"
"DoubleClickSpeed"="370"
"DoubleClickWidth"="32"
```

On Windows XP the DoubleClickSpeed ranges from 200 (Fast) to 900 (Slow)

OS double click options

One potentially useful setting in Windows, when using the desktop via touch, is to invoke icons, folders and files using a single click rather than double click. This setting is located in the folder options as shown below:



UPDD generated double click settings

For a future release of UPDD we are considering processing our own double click setting for situations where it is still difficult to achieve a double click using system DC setting with certain touch devices such as slow infra red devices. In this case where UPDD DC setting deduce that a double click is required then two complete click sequences will be sent at the same location very quickly which will result in a double click.

Serial port auto detection

Since UPDD 4.1.6, build 1156, a serial controller can be set for auto-detection rather than set to a specific com port. This option is only available, and enabled, where a serial controller can receive and acknowledge a firmware command. Controller commands are held in [UPDD macros](#). Typically a macro sequence used to detect a controller would look like this:

```
[HEX]0A0141 //Check Controller Activity
[ACK]0A0141
```

In this example the driver sends 0A0141 to the controller which is echoed by the controller by way of acknowledging receipt.

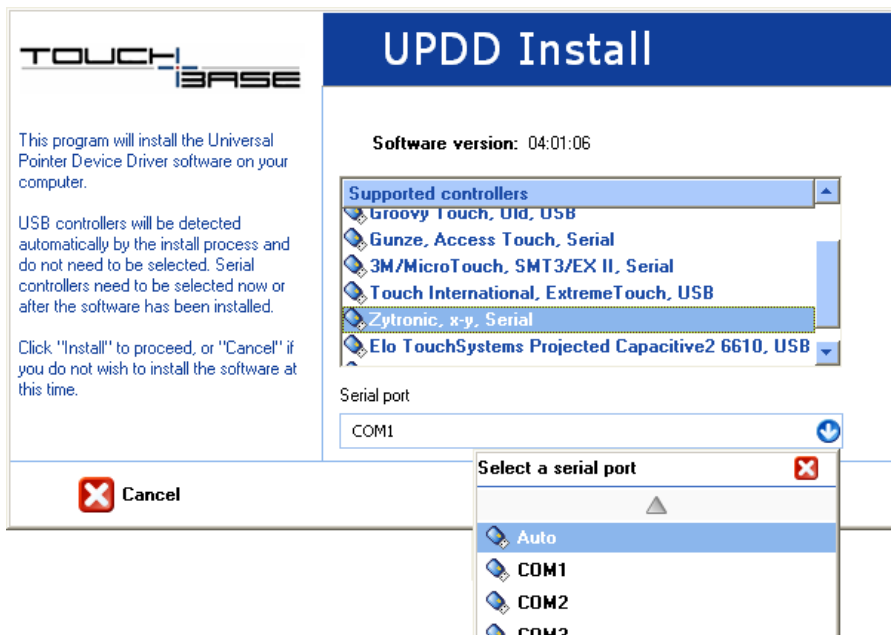
Some macros are needed for controller initialization rather than just containing commands to illicit a response but as long as the controller responds to the received commands then the macro can be used to detect the controller.

When the controller is set for Auto detect the driver issues the macro to each com port in turn looking for the response. If a response is received then the com port is set accordingly. If no response is seen from any port then the com port is set to 'None'. Each time UPDD is loaded it will automatically detect the device but on 2nd and subsequent attempts the detection will initially check the current port setting before checking thro each port in sequence.

Setting serial controller for auto detection

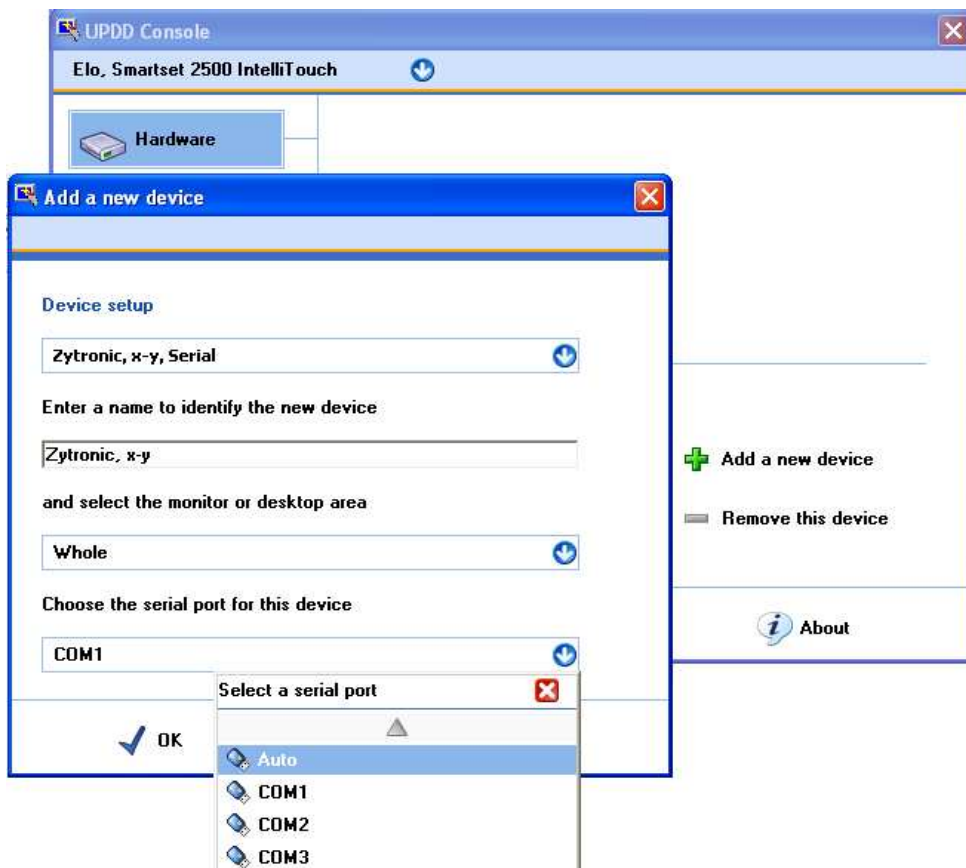
There are various ways to indicate a serial controller is to be auto-detected as follows:

- Set in build** A controller is defined in our production system as auto-detected such that is the default settings in all dialogs below.
- During installation** The controller is selected from the list and the com port set to Auto:



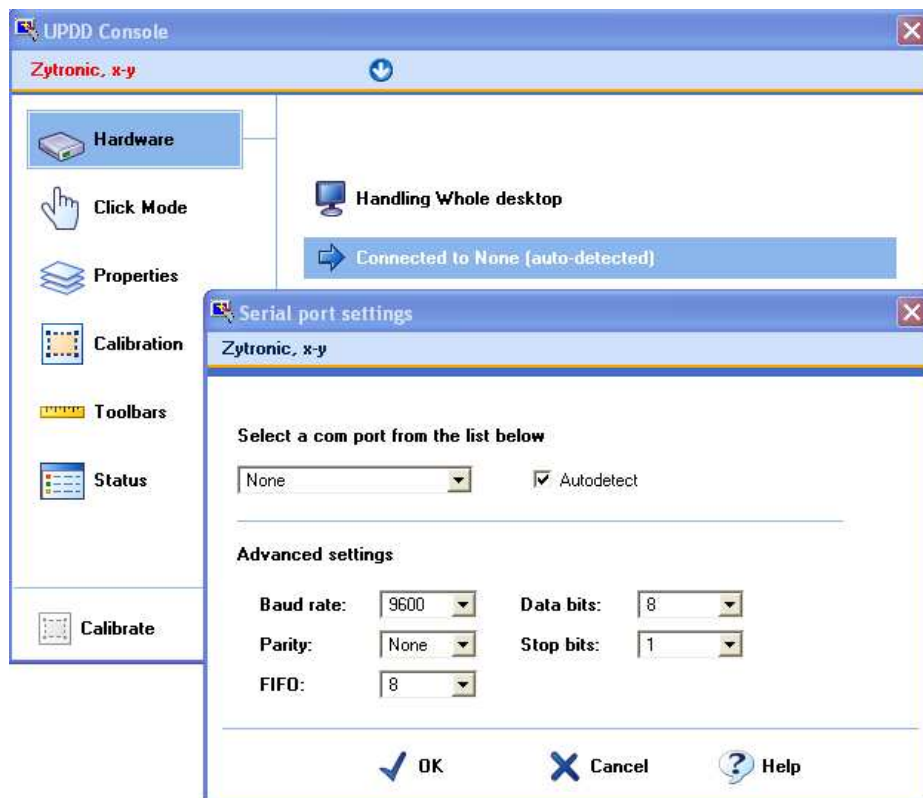
Adding a device in UPDD Console

When adding the device the com port is set to Auto:



Serial port settings

The serial port settings dialog shows the current port placement and indicates if this port has been auto-detected. In the example below the port is auto-detected but has not been found:



Right Click processing

In the majority of cases touching a touch screen or similar device will normally be the equivalent of generating a mouse left click. However, in some cases it is also desirable to be able to generate a right click via touch. With UPDD there are three possible methods; the Event Selector (to indicate if the next touch should generate a left or right click), using Interactive Touch mode (which caters for both left and right clicks) or using the right click mechanism built into the operating system; as described below. *It should be noted that when utilizing OS in-built right click functions any UPDD native right click processing is disabled.*

Event Selector

The [UPDD Event Selector](#), in its most simple terms is a user controlled trigger that indicates to the driver if the next touch should be a left or right click. Click the link for more details.

Interactive Touch

Using the [UPDD mouse emulation mode](#) Interactive Touch caters for both left and right clicks.

In normal touch usage a touch will generate a left pen down and allow for drag then will generate a left pen up at lift off. However, holding the stylus stationary invokes right click. The Interactive switch delay setting determines the hold delay and the visual notification setting indicates if visual feedback is shown during hold and right click, as shown.

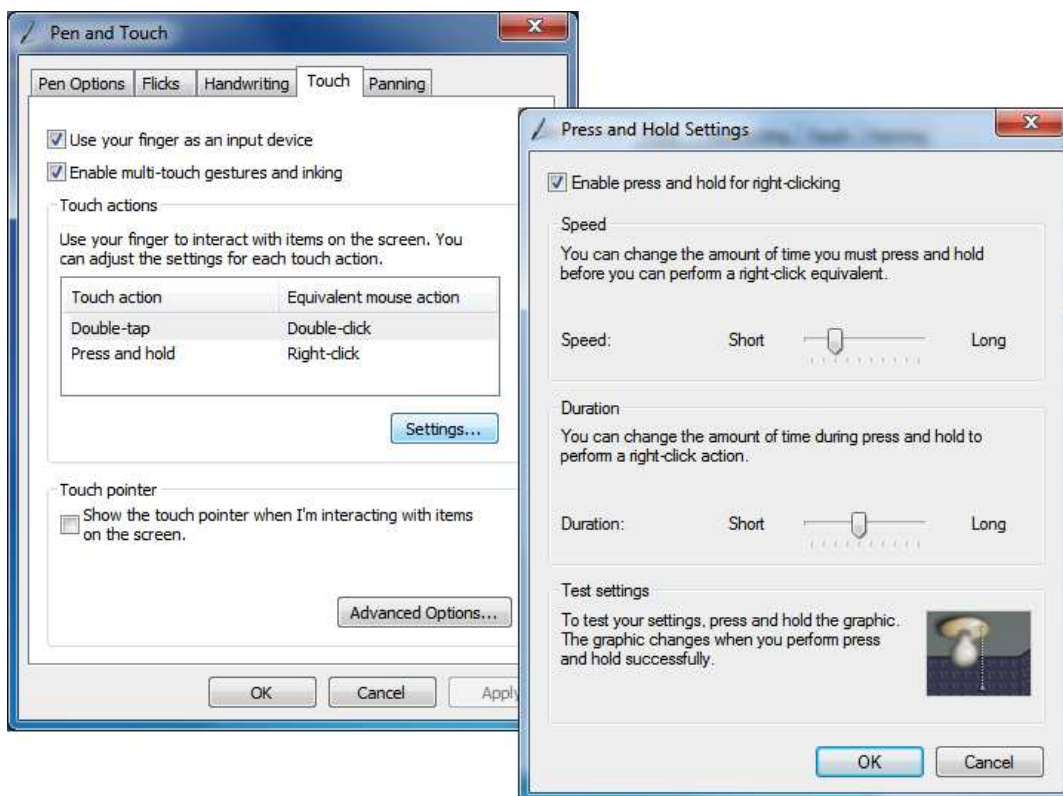


Operating System feature

The operating system may cater for right click generation using a pen or touch device and in some cases this can be utilised by the driver, as discussed below:

Windows

If using [UPDD 4.1.8 and above](#) in touch enabled versions of Vista or Windows 7 there is a setting in UPDD, called [Extended Touch](#), that passes touch data to the OS such that all touch features built into the OS are available to the touch user. In the Pen and Touch system settings you will find some settings specific to touch and you can associate a right click to one of the actions as shown in this Windows 7 example



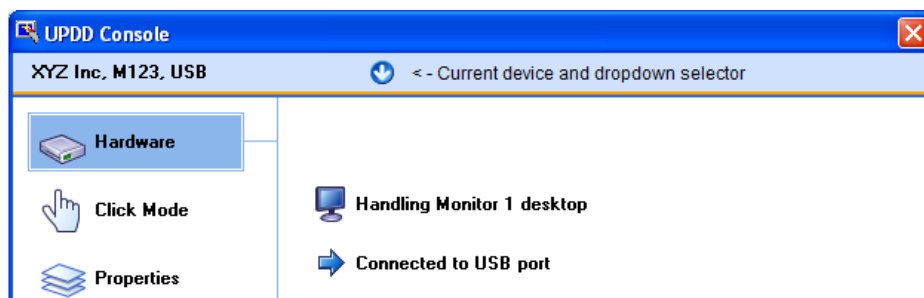
In this example, the Touch action 'Press and Hold' has been associated with the Right-click function. During the hold phase a circle will be drawn and when complete a right click will be generated. The circle drawn on a pen device is smaller than the circle drawn on a touch device. Pen and Touch settings dictate the hold time needed to initiate a right click.



Controller naming

In our controller definition database we allocate default controller names to help identify the manufacturer, model and interface, e.g XYZ Inc, M123, USB or a name as requested by the manufacturer or system integrator.

In the UPDD Console there is a Device list entry that shows the currently selected device and, if more than one device is configured on the system, offers a dropdown list to select other devices.



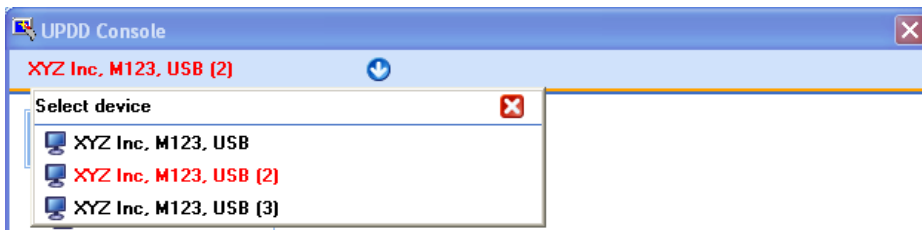
If another controller of the same type is plugged in then the driver will allocate a unique name based on the name of the controller along with a bracketed number i.e. XYZ Inc, M123, USB (2), XYZ Inc, M123, USB (3) etc

Each controller discovered on the system is [allocated a unique bind key](#) to help associate and identify UPDD controller settings with individual controllers as they are unplugged, reconnected or rediscovered, such as after a reboot.

In a 3 touch monitor device with all devices connected the device list would be seen as:

XYZ Inc, M123, USB
 XYZ Inc, M123, USB (2)
 XYZ Inc, M123, USB (3)

Any PnP devices that are unplugged would have their name displayed in red. So unplugging (2) would result in the device list as follows:



If the controller is subsequently reconnected, then the driver will calculate the bind key to find the previous entry and settings and re associate the device with its previous settings.

However, if the device in red is deleted from the UPDD console (using the 'Remove a device' option), leaving two entries:

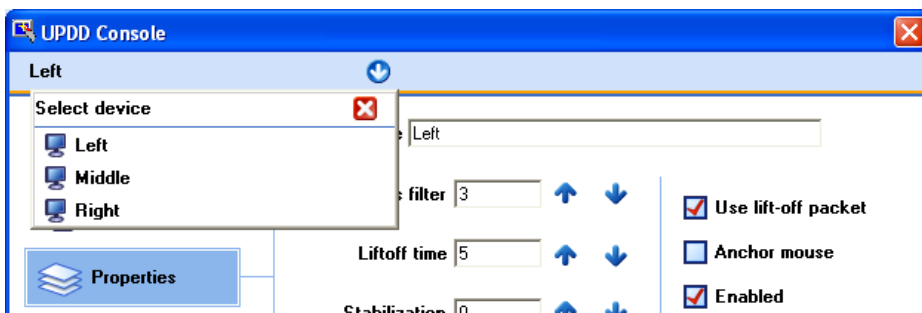
XYZ Inc, M123, USB
 XYZ Inc, M123, USB (3)

then the next time the device is connected no previous settings will be found and a new device entry will be created. Given (2) is the next unique device name then the device will reappear as XYZ Inc, M123, USB (2) but will be allocated default settings which may need manually adjusting.

Our recommendation is to not delete devices if they are to be reconnected at some future date.

Controller renaming

Controller names can be manually adjusted once the device is listed in the UPDD Console device list. Simply select the device, click on Properties and change the name. In a three touch monitor system you could name the controllers to reflect their position on the desk, say Left, Middle, Right as shown below:



In this instance, plugging in a 4th controller would result in the name of the new controller being XYZ Inc, M123, USB.

Daemon task

Under Windows desktop version of the driver the software utilizes a daemon (or background) task to implement certain functions related to the driver and user interface. The program is called TBdaemon.exe and resides in the UPDD application folder. In UPDD version 3 a this program was called TBSystry.exe.

The daemon task is invoked at system startup due to the 'tbdaemon' registry entry in the Windows run branch HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

The daemon task is responsible for the following functions:

- | | | |
|-------------------|--|---|
| System tray icons | | <p>Shows the rotate, event selector and menu icons in the system tray (if enabled – they can be disabled in the driver build if requested)</p> <p>Rotate: The rotate icon can be used to initiate desktop rotation and is only shown if the rotate interface is supported.</p> <p>Event Selector: Used to switch between left and right click generation. The 'red' area indicates the mouse click to be generated.</p> <p>Menu: Shows the UPDD System tray menu below.</p> |
| System tray menu | | <p>Enabled: Shows if the device is enabled at a UPDD level. Enabled state can also be switched. Devices can be enabled or disabled at the driver level irrespective of their physical state. Directly relates to the same setting in the UPDD Console, Properties page.</p> <p>Calibrate: Invoke calibration for a device.</p> <p>Adjust settings: Invoke the UPDD Console.</p> <p>Test: Invoke the test and drawing program for a device</p> <p>Event Selector: Invoke the desktop Event Selector function.</p> <p>Help: Load the driver's help files.</p> <p>About: Release information. The build id, as seen below, indicates the version (4.1.6) and release status 'R', the build no. (1221) within the release and the production system 'G' that generated the build:</p> |



Rotate monitoring	For supported rotate interfaces informs the driver if a desktop is rotated.
User double click settings	System double click settings need to be set such that double click can be achieved with a stylus. If a new user logs on these settings are set for touch usage.
Toolbar playback	Actions associated with a toolbar when touched are initiated.
Anchor mouse monitoring	Monitors the position of the cursor and resets the position after touch if the Anchor Mouse setting is enabled in the UPDD Console, Properties Page .
Live annotation	If live annotation is enabled performs the drawing.
Extended sound	If extended sound is associated with a toolbar plays the sound.
Beep on touch	If enabled in the UPDD Console, calls the Windows beep function to produce the 'beep'.
Video resolution tracking	Tracks the video resolution and switches to resolution associated calibration data if it exists. E.g. If a resolution switch results in the desktop video changing position the calibration would be out. This function sees a resolution change and looks for calibration data specifically associated to the new resolution. The calibration data must be held against a calibration style in the format RxxxxYyyyy, e.g. R1024X768. If no style exist the current calibration data is used. In most cases switching video resolution does not adjust video desktop positioning so a single calibration is fine for all resolutions.

Pen up processing

A pointer device generates data packets when in use, such as touch packets whilst a stylus is in contact with a touchscreen. When the device is no longer in use then obviously the data packets will stop being generated. A driver or application may need to perform a specific function at this point, such as pen up processing if the device is being used in mouse emulation in an appropriate mouse click mode.

Most touch screen devices, but not all, will indicate in the last data packet that the stylus has left the screen. These data packets are often referred to as the pen up or lift off packet.

If a device has a pen up packet defined in the UPDD controller configuration then there is a UPDD Setting called "Use Lift Off packet". This setting can be set in the UPDD settings file as appropriate or via the UPDD Console, properties page and is enabled by default.

UPDD will generate a pen up in mouse emulation mode when either:

- the 'pen up' packet is seen (and pen up processing is enabled)
- and/or when data packets cease (if Lift off time processing is enabled).

This approach caters for all methods of 'pen up' detection:

- The device generates pen up packets
- The device does not generate pen up packets
- The device generates pen up packets but due to communication error is not received or is corrupted.

To generate a pen up where data packets cease there has to be a short wait prior to generating the pen up to ensure the data has stopped rather than being a gap between the packets. In UPDD this short wait period is referred to as the Lift Off time settings, as described below.

Lift off time setting

The Lift off Time value specifies the time interval required to register a stylus lift after the last touch packet is received. Lift off time is defined in units of 20ms. This value is used to perform a pen up if the 'Use Lift off' packet is disabled otherwise Pen ups are generated as soon as the stylus leaves the pointer device display.

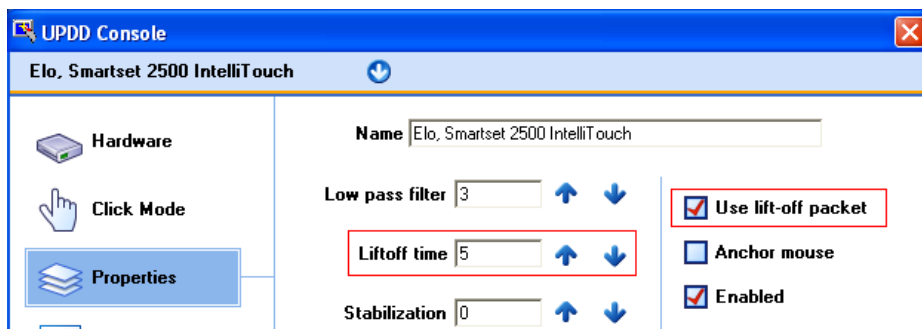
However, because this timer is triggered after each received touch packet it is important to ensure the value is greater than the time interval between data packets otherwise pen up events will constantly be generated. In default UPDD configurations this setting is often set as low as possible so that pen up is generated as quickly as possible after the last packet. However, if for any reason the packet speed is slowed down, multiple pen ups and pen downs will be generated due to the increased time gap between packets requiring the Lift off time value to be increased as appropriate.

Delta mode considerations

Delta mode refers to a pointer device mode whereby data packets with the same X and Y coordinate values (i.e. the stylus is stationary) are not sent by the device. In this mode, UPDD will generate a pen up when the stylus is held stationary. To cater for this situation a lift off time value of zero is defined to disable pen ups based on time.

Settings

The 'Lift off time' and 'Use pen up packets' settings can be set via the UPDD Console, Properties page or directly in the UPDD settings files:



If the console is not available the setting can be found in the UPDD settings files as follows:

UPDD version	Operating system	Location
UPDD 4.0.x	Windows	registry at HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TBUPDD
UPDD 4.0.x	Linux and Mac OS X	tbupdd.reg
UPDD 4.1.x	All OS	tbupdd.ini
UPDD 4.0.x	Windows CE	HKEY_LOCAL_MACHINE\Drivers\BuiltIn\TBUPDD\Parameters\{...}\1 - Where {...} is a GUID that identifies the package.

Within the files will be a branch 'Parameters'. With UPDD 4.0.x this will be followed by a GUID number (a very large number). For each controller there will be a controller number 1,2 etc. Therefore the location for a single controller system will be:

```
4.0.6 - CE      HKEY_LOCAL_MACHINE\Drivers\BuiltIn\TBUPDD\Parameters\{guid}\1
4.0.6 - Others HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TBUPDD\Parameters\{guid}\1
4.1.1 - All    [UPDD\parameters\1]
```

liftofftime=0x0000000n (n = 0 = disabled or lift of time value in units of 20ms)
use liftoff bit =0x0000000n (n = 0 = disabled, 1 = enabled)

Advance Console

With release 4.1.10 we hope to release the first version of the UPDD Advance Console.

The standard [UPDD Console](#) shows the basic driver settings and allows them to be updated. However, there are many other settings that may occasionally need to be updated and the Advance Console is being created to offer a GUI interface to these settings. Currently these settings need to either be set as required as default or updated manually.

Once this utility is available a separate web document will be created to document features and usage.

Live Annotation

With release 4.1.8 we have implemented Windows Live annotation function. This function allows a touch screen user to annotate over a live desktop. Annotation can be enabled, disabled, paused and erased. The line colour and width can also be defined.

This functionality came about from a project we undertook for a cable television company to allow a presenter to annotate over a live Windows desktop showing financial data. The presenter uses [UPDD Toolbars](#) at either side of the larger touch display (with no desktop visual representation – so not seen by the viewer) to invoke the annotate functions as required.

Annotation functions can be controlled via UPDD API calls and we have implemented two user interfaces of our own, as discussed below:

Toolbar interface

The UPDD toolbar interface implements toolbar cell actions to control all annotate functions. An example of an [Annotate toolbar is described in full](#) in the toolbar documentation.

UPDD Console interface

The UPDD Console, [Extensions dialog](#), exposes a simple UPDD interface to the Annotate functions.

Annotate API calls

The annotate settings:

```
AnnotateThickness
Annotate Color
AnnotateMode
```

Are set via the standard api followed by TBApiApply

e.g.

```
TBApiSetSettingDWORD(0, "AnnotateThickness",3);
```

Color is a Windows COLORREF value NOT RGB; example

```
COLORREF cr = RGB(r,g,b);
```

```
TBApiSetSettingDWORD(0, "AnnotateColour", cr);
```

There's no API to clear. This can be achieved as follows (C++ example)

```
#define WM_DOCLEAR 4365
BOOL CALLBACK _ClearPicture(HWND hWnd, LPARAM lParam)
{
    char className[200];

    GetClassNameA(hWnd, className, sizeof(className));
    if(!strcmp(className, "TBUPDD_Annotate_Class"))
    {
        :PostMessage(hWnd, WM_DOCLEAR, NULL, NULL);
    }

    return true;
}

EnumWindows(_ClearPicture, NULL);
```

The values for mode are

```
0=pause
1=annotate
2=stop
```

If annotation is in a stopped state then `activedw.exe` must be launched to activate annotation. Setting stop will cause `activedw.exe` to terminate.

Contact

For further information or technical assistance please email the technical support team at technical@touch-base.com
