

Windows: [Extended features](#)[UPDD device specifics](#)[Settings](#)[Feature usage](#)[Applications](#)Mac: [Extended features](#)Linux: [Extended features](#)[Contact](#)

Some operating systems have extended touch features, such as [flicks](#) and [utilities](#) that are utilised by both the operating system desktop interface and applications specifically written to handle the touch features.

For example, Windows Tablet editions of Vista implemented the touch extensions for tablet PC's (those built with touch LCDs) and at the time catered for single touch technology. Further, Windows 7 has some form of touch support in all editions and high end editions, starting with Home Premium, has additional support for multi touch technology, allowing for the use of [gestures](#) to control desktop and application interaction.

Linux kernel 2.6.30 is the first kernel to cater for multi-touch events and this is being extended with each release of the kernel.

The extended touch features made available in the system are dependant on the edition of the operation system and the touch technology in use. Additionally, the OS will only enable extended touch features against touch technology that conforms to certain hardware standards. In the case of Vista and Windows this has to be a recognised pointer device type that conforms to HID standards (or a software driver, such as UPDD, that implements a virtual HID device). Pointer device type devices, such as mouse, touch screens, tablet, pen and digitizers can all be developed as HID devices and when connected to the system and handled by the system's HID driver will automatically enable any relevant extended touch functions.

However, in cases where the hardware does not conform to required standards or the system driver (i.e. HID) does not have the desired functionality then a driver can be used to both satisfy the system requirements and/or implement additional touch functions over and above those available as extended touch features.

This document reflects the OS extended touch features enabled by UPDD 4.1.8 (and above) and will be updated as support is extended or improved.

### Windows extended touch features

This section describes the touch features available when enabled, either because a compatible HID device is in use or UPDD is installed on an operating system offering such features.

The following grid shows the touch functions enabled by the use of compatible HID hardware or the utilisation of a Virtual HID device, such as implemented by UPDD and other pointer device drivers:

Pointer Device	Mouse	HID Touch	HID Multi Touch	Non HID touch (Serial,PS/2,USB)	Non HID touch (Serial,PS/2,USB)
<b>Device Driver</b>	Mouse class	HID class	HID class with multi touch extensions	UPDD Mouse class interface - Note 3	UPDD HID MT class interface - Note 3
<b>Touch features</b>	Mouse	Single touch	Single and multi touch - Note 1	Mouse	Single and multi touch - Note 1 & 2

Note 1: Multi touch extensions only available in high end Windows 7 editions - see table below.

Note 2: Multi touch dependant on touch screen capabilities. Most (but not all) are single touch devices and therefore are not physically capable of multi touch albeit the UPDD Virtual HID is registered as a MT device. When handling a single touch device it is the equivalent of a single touch on a MT device.

Note 3: UPDD creates both a mouse and Virtual HID type device such that touch data can be delivered to the operating system via the mouse class or HID class driver. The desired delivery method is [switchable in the UPDD Console](#).

The following table indicates the touch features available by Windows OS editions:

Function\OS	XP	Vista	Windows 7 Starter	Windows 7 Home Basic	Windows 7 Home Premium	Windows 7 Ultimate	Windows 7 Professional	Windows 7 Enterprise
Single-touch mouse	✓	✓	✓	✓	✓	✓	✓	✓
Single-touch digitizer	✗	✓	✓	✓	✓	✓	✓	✓
Two-touch zoom	✗	✗	✗	✗	✓	✓	✓	✓
Two-touch horizontal scroll	✗	✗	✗	✗	✓	✓	✓	✓
Two-touch horizontal scroll	✗	✗	✗	✗	✓	✓	✓	✓
Two-touch double tap (2 finger)	✗	✗	✗	✗	✓	✓	✓	✓

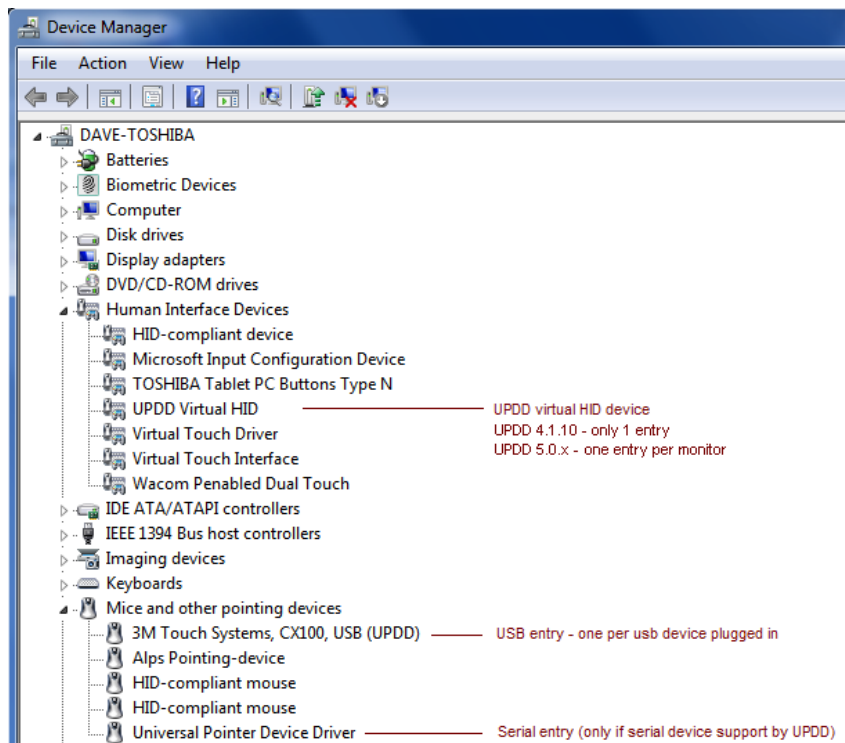
### UPDD device specifics

When UPDD is installed it creates a number of entries in the Device Manager that reflect both the hardware types supported by the specific UPDD build and also relate to the version of the OS.

If a serial controller is defined within the driver then there will be a serial device entry in the 'Mice and other pointing devices' section. A USB device will also be created in this section for any USB devices that are supported and plugged in. These device

types are currently created for all Windows OS supported by UPDD version 4.1.x (and above). In addition to these base line devices, starting with UPDD 4.1.8, an extra virtual HID device is created in Vista and Windows 7 so that the operating system enables any HID related features, such as extended touch features.

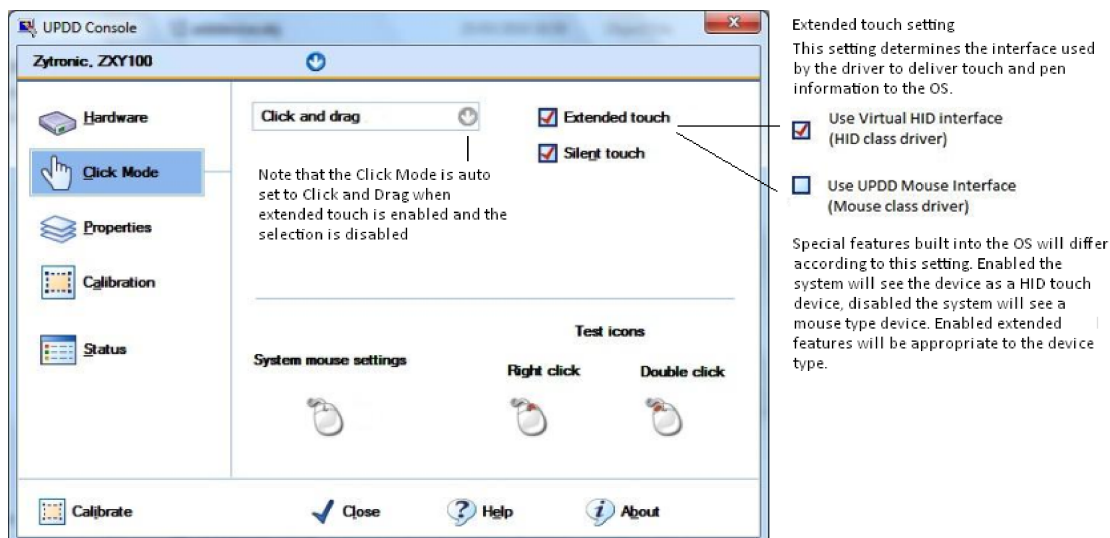
Currently we create the same Virtual HID multi-touch device, irrespective of device characteristics. In future releases we may tailor the virtual HID device more in keeping with the device characteristics (e.g. distinguish between single and multi-touch devices).



In the above example UPDD is installed on a Tablet PC that has other touch related HID devices (Virtual Touch Driver, Virtual Touch Interface, Wacom Penabled Dual Touch)

Therefore, starting with UPDD 4.1.8 touch data processed by UPDD can now be delivered to the Vista and Windows 7 operating systems via the mouse class interface or the HID class interface and processed accordingly. In single touch environments without extended touch functionality either feed will result in normal system pointer movement and mouse clicks. However, in environments with extended touch functionality the HID feed will also translate to system pointer movement and mouse clicks but will also [enable the extended touch features](#) where appropriate. In certain Windows 7 editions this will also result in gesture processing when using multi touch devices.


The system interface used by UPDD (mouse port or HID) is determined by a UPDD Console setting as it may be desirable in systems utilising UPDD where extended touch features are available but not desired for the UPDD Device to route touch data to the mouse class interface instead of the HID interface. This will reduce the extended touch features to those that work with a mouse type device so features such as gestures will no longer work with the UPDD device. It will also enable some UPDD features that are disabled when HID interface is used, such as the selection of different click modes.



#### Extended touch feature settings

To determine the type of touch available in your computer open the System dialog by selecting the Start button, right-click Computer and then Properties. You will see an entry such as the one below showing that both a Pen and Single touch device are available on the system:

System

Rating:  Windows Experience Index

Processor: Intel(R) Core(TM)2 Duo CPU P8600 @ 2.40GHz 2.40 GHz

Installed memory (RAM): 2.00 GB (1.85 GB usable)

System type: 32-bit Operating System

Pen and Touch: Pen and Single Touch Input Available

In this example, only a touch device (not Pen) is configured that offers multiple touch with 10 touch points.

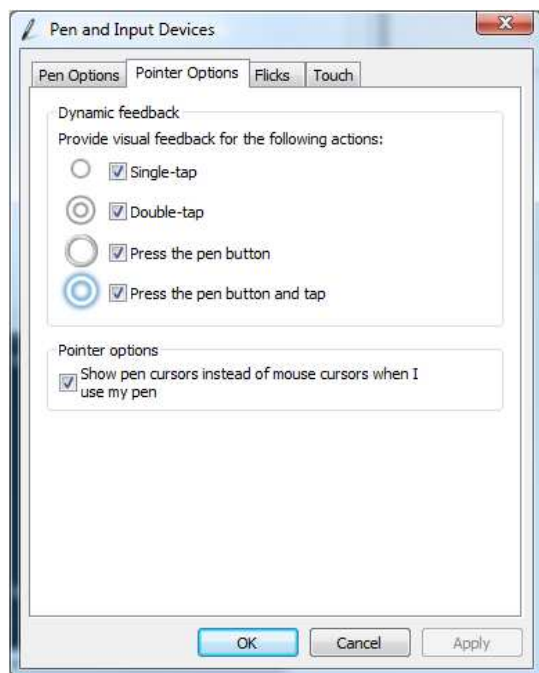
Pen and Touch: Touch Input Available with 10 Touch Points

With only UPDD installed the Pen and Touch will show a 2 touch point input device (version 4.1.8) and 16 point input device (version 4.1.10). This reflects the fact that we register a 2 / 16 point device to handle any type of touch screen (single, dual or multi-touch) device, irrespective of the device physically connected.

The extended touch feature settings are found in the system's Control Panel. *Not all editions of Windows Vista and Windows 7 will have all of the Pen and Touch options mentioned. This is a limitation of the edition of the operating system.*

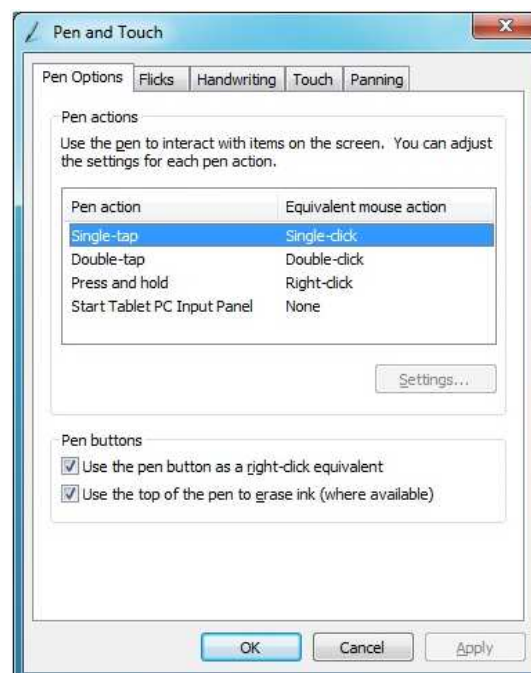
#### Vista

This shows the Vista **Pen and Input Devices** settings dialog:



#### Windows 7

This shows the Windows 7 **Pen and Touch** settings dialog:



Some settings are only shown when a multi touch device is configured on the system. Currently UPDD specifies all touch hardware as a multi touch device so these settings will only be relevant if a physical multi touch device is in use! In a future release we may cater for a variety of device types selectable in the UPDD Console depending on the desired functionality but the device type currently defined caters for most requirements.

#### Extended touch feature usage

This section is intended as a very brief introduction to the usage of extended touch features that are enabled when utilizing HID compatible touch / pen / digitizer type hardware or Virtual HID devices, as implemented by 3<sup>rd</sup> party drivers such as UPDD. The operation of some functions will be influenced by the settings above. Additional touch related articles are on the web, such as the [Windows 7 features page](#) on the Microsoft site or this [comprehensive article](#).

The functionality and screen shots shown here relate to Windows 7 but there will be very similar or equivalent features under Vista. The term touch is used to represent both Pen and Touch devices as they perform similar roles. Pen devices may also have buttons to initiate left and right clicks.

#### Desktop

When using a touch enabled desktop, depending on the touch hardware in use, some or all of the following touch features will be available:

#### Stylus Interaction

Action	Usage	Description
Movement	Cursor movement	The cursor will move relative to the point of touch.
Tap	Click generation	Touching the screen will result in a single click. Double tapping the screen will generate a double click. In some cases visual feedback will indicate the type of touch generated.
Double Tap		
Press and Hold	Right click generation	Press and hold or Press and Tap on the touch or pen device will normally generate a right-click. During the hold phase a circle will be drawn and when complete the right click will be generated. The circle drawn on a pen device is smaller than the circle drawn on a touch device. Touch and Pen settings dictate the hold time to initiate right
Press and Tab		

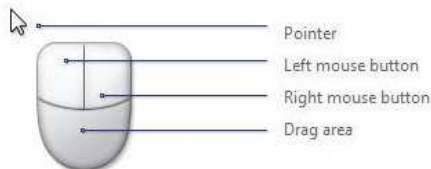
click.



**Utilities**

Touch Pointer

If configured in the Touch settings “Show the touch pointer when I’m interacting with items on the screen” then the ‘Touch Pointer’ is shown when interfacing with the desktop. The Touch pointer is a small visual aid shaped like a two-button mouse. It’s composed of a left-click button, right-click button, and a drag area. The system cursor stands ready a short diagonal distance away from the touch pointer, so that while dragging the touch pointer, you can see the cursor while you carefully target the small items on your desktop, such as resize handles, minimize/maximize/close boxes, checkboxes, etc.



Note that the cursor is not limited to one standard position away from the touch pointer. Dragging the touch pointer quickly toward a corner, or side of the display, will flip it around to another position, so that you’re never prevented from accessing even the deepest parts of the screen.

Input panel



The Input Panel is ‘parked (hidden)’ off screen. When the touch is used the edge of the panel is seen. Moving the cursor towards the panel edge will partially reveal the dialog that can then be selected for use. This panel offers keyboard and handwriting utilities, as shown below. The utilities can be invoked manually or automatically when user input is required.

When using the touch interface and user input is required small icons are shown next to the point of input to invoke the Input Panel utility.



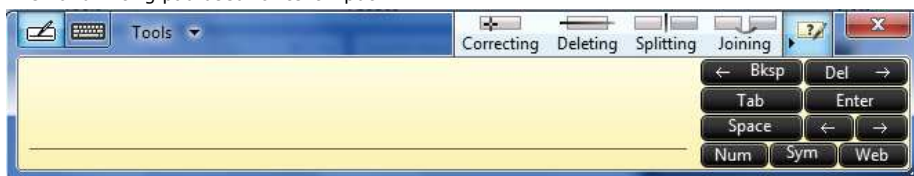
Keyboard

The onscreen keyboard used for text input:



Handwriting

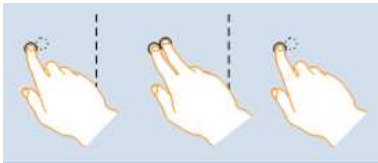
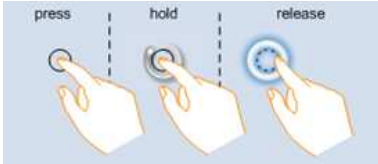

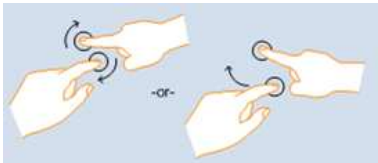
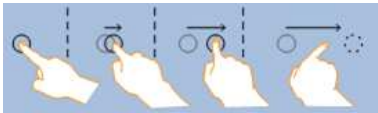
The handwriting pad used for text input:

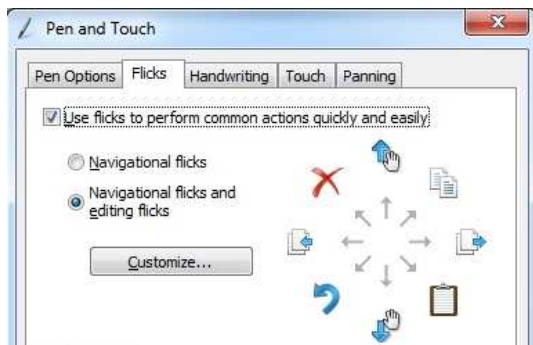


**System and Application Interaction**

Gestures These are motions you make with one or two fingers

GESTURE	WINDOWS USAGE	GESTURE ACTION	ACTION (○ = finger down   ◉ = finger up)	Description
Panning with Inertia	Scrolling	Drag 1 or 2 fingers up and down		Use panning to see another part of a page that has scroll bars. For example, you can pan to see part of a long document or spreadsheet that doesn’t appear in the window. When you pan with one finger, dragging your finger vertically moves the page, while dragging your finger horizontally selects text on the page.
Selection / Drag (left to right with one finger)	Mouse Drag / Selection	Drag one finger left / right		Note: Some programs will only pan with 1 finger, such as Windows Photo Viewer
Press and Tap	Right click	Press on target and tap using a second		

		finger		
Press and Hold	Right click	Press, Wait for blue ring animation to complete, then release		
Zoom	Zoom (defaults to CTRL key + Scroll wheel)	Move two fingers apart / toward each other		Use zooming to make an item on the screen larger or smaller. With a picture, zooming in shows a smaller area in more detail; zooming out shows a larger area.
Rotate	No system default unless handled by Application (using WM_GESTURE API)	Move two fingers in opposing directions -or- Use one finger to pivot around another		Use rotating to move a picture or other item on the screen in a circular direction (clockwise or anti-clockwise).
Flicks	Default: Pan up / Pan Down / Back, and Forward	Make quick drag gestures in the desired direction		Flicks are a specialised kind of gesture that only works with stylus or touch input. All you have to do to use them is perform an appropriate flick. Flicking your pen to the left generates a left or back action. Flicking your pen to the right generates a right or forward action.



There are also very many subtle differences in desktop interaction when the OS determines that a mouse, pen or touch is being used and adjusts accordingly. For example, items in lists boxes have a different layout (more spacing) if the list box was invoked by touch to allow for the entries to be easily selected with the finger. With a mouse induced list box the items are more compressed. This is just one small example of the different user interface between mouse, pen and touch. A good video overview to the touch features can be viewed here:

<http://channel9.msdn.com/posts/yochay/Windows-7-Multi-Touch-Overview/>

**Applications**

Many applications running on Windows 7 will inherit default touch interface provided by the OS. However, for a more specific or 'better' touch experience applications can directly handle touch, multi-touch, gestures and flicks and many of the standard Windows 7 applications have been updated to utilize the touch interface, such as [Windows Explorer](#). You can also, for example, use Paint to simultaneously draw with multiple stylus, as shown in the screen shot below using two stylus at the same time:

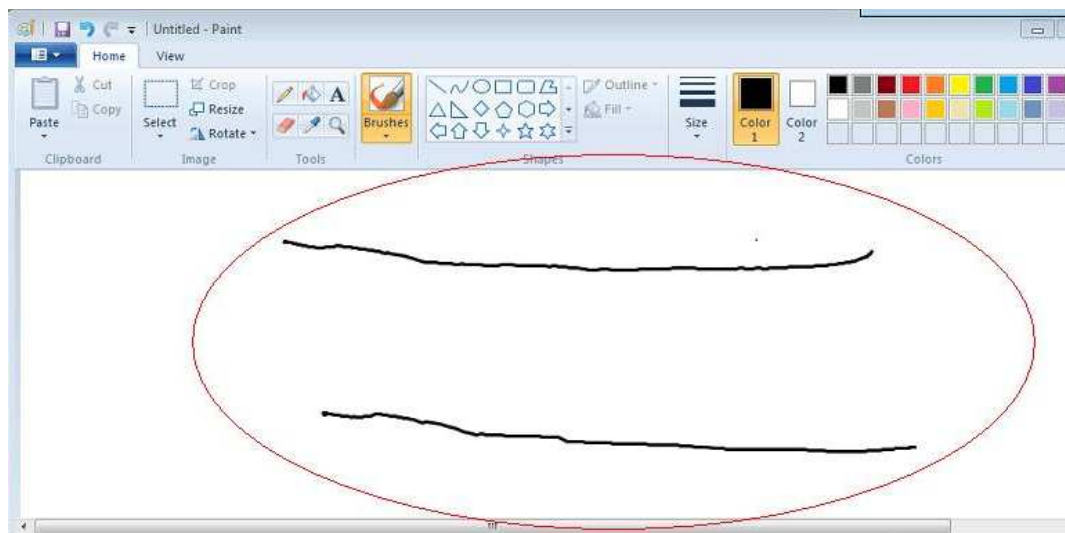


Photo applications use gestures to size, rotate and move photos. Multi-touch games have been developed for multiple users.

Many technical articles are available describing the various related application programming interfaces for developers to create touch aware applications.

Very informative development video series can be viewed here:

<http://channel9.msdn.com/posts/yochay/Programming-Windows-7-Multi-Touch--Part-1/>

<http://channel9.msdn.com/posts/yochay/Programming-Windows-7-Multi-Touch-Part-2/>

Most samples shown in the videos can be found in the Windows 7 SDK. You can find them in the \Program Files\Microsoft SDKs\Windows\v7.0\Samples\Touch.

### Linux extended touch features

Multi-touch events were first catered for in kernel 2.6.30 and were officially released with 2.6.31 so any distribution using this release will cater for some form of multi-touch support. The main web site for this development is [here](#). This site list the requirements to achieve MT in Linux offering advice on supported hardware, drivers, installation and demo programs. At the time of writing (May 10) this seems to be ongoing development as the development team have written "we are now involved in the thinking on how to reorganise input management in both Xorg and the kernel to better deal with multi-touch, including gesture recognition."

We have made changes to UPDD Linux 4.1.8 to interface with this evolving standard and have working drivers for multi-touch hardware supported by UPDD which has been tested on Fedora Core 12. We have also modified some of the supplied demo programs to interface with our implementation. Our development is also ongoing.

With UPDD Linux MT we offer the following:

- Support for MT controllers
- File access to stylus data
- UPDD API access to stylus data
- Pass though of stylus data to X multi touch interface

### Mac extended touch features

Multi-touch events under Mac OS X have been around for some time and were available on systems fitted with a multi-touch trackpad and latterly a Magic Mouse with its touch sensitive shell.

We have developed a [gesture extension for Mac](#) that can be installed alongside the UPDD driver that injects multi-touch input into the OS such that the standard gestures can be performed on a touch screen.

### Contact

For further information or technical assistance please email the technical support team at [technical@touch-base.com](mailto:technical@touch-base.com)