Windows Desktop Installation

Revision 1.05, 17th Jan 2015
www.touch-base.com\documentation\installation

Deliverables	Requirements	CD distribution	Installation Notes	Signed Drivers	Sysprep	Installation	Options
Touch Interaction	Utilities	Useful settings	Calibration	Console	Serial Notes	Uninstall	Components
Processes	Daemon task	Multi-monitor	Rotate	System Settings	Touch Utilities	Troubleshooting	Contact

Welcome to UPDD Windows desktop specific installation instructions and related notes for UPDD version 5.1.x.
If using UPDD 4.x.x or 5.0.x please refer to previous Windows desktop installation instructions.

Development history can be viewed from the UPDD Console and can also be viewed here.

These notes should be followed to install the UPDD 5.1.x pointer device driver on Windows Desktops running Windows, 2003, XP/XPe, Windows 64 bit, Vista 32 / 64 bit and Windows 7 and 8 32 / 64 bit.

License Notice

The software is licensed software and as such requires a license per system when the production version of the software is installed. Production software is either supplied by pointer device manufacturers or system integrators (who are entitled to distribute the driver) or is available directly from Touch-Base sales.

Evaluation version restrictions

Evaluation software, which is available from our Download Centre is mainly used to test the touch function is working, has certain restrictions:

- When working in 'mouse emulation' mode clicks do not work after 50 touches, only movement, at which point a nag screen is shown. Selecting OK on nag screen offers a further 50 clicks.

- The nag screen is processed by the driver's daemon task, aidaemon, so if this is not running the nag screen will not be shown and the counter will not be reset.

Deliverables

Unless otherwise requested, the software is distributed via email as an HTTP link, in either compressed .ZIP format or in an executable .exe format. There are separate files for 32 and 64 bit systems.

System Requirements

Touch device	A supported USB, Serial or PS/2 touch device.
OS	Supported Windows desktop OS.
	Important note. XP service pack 3 introduces significant and important updates relevant to USB hardware support and driver interaction that we consider it essential for the correct functioning of our driver. Please ensure any XP systems are upgraded to SP3 before installing our driver.
	It is our general recommendation that these operating systems are kept up-to-date with the latest service packs.
C run-time libraies	UPDD Version 5.1.x no longer statically links the C run time library, instead relying on it being installed on the system. If any programs fail during install, likely with the error "The application failed to initialize properly" then it is possible the C run-time lib is missing. It can be downloaded from the Microsoft web site at http://www.microsoft.com/en-gb/download/details.aspx?id=29
Access to TCP IP port 4141	The driver requires access to TCP IP port 4141 for internal computer processing only.

CD Distribution

The setup program is placed on the CD alongside a autorun.inf file. Insert the CD and the autorun file automatically invokes the setupnn.exe program if the autorun.inf file contains:
[autorun]
OPEN=.\Setupnn_05.01.nnnn.exe (where nn = 32 or 64 bit)

Pre installation notes

1. **Uninstall previous touch drivers**
   As a matter of course it is highly recommended that any other non UPDD touch drivers be uninstalled (followed by a system reboot) prior to installing UPDD.

2. The installer creates a folder tbtemp in %temp% as defined by the environment variable TEMP. The environment variable and folder must exist. The component to expand cabinet archive files must also exist.

3. **USB consideration**
   It is recommended that where possible the USB device is connected to the system after installation. This allows the driver to discover the device once connected and take the appropriate action, avoiding conflict with the Windows device management. If during install or when the device is connected the Windows Hardware Wizard is invoked use the Wizard to select the UPDD driver (this will only happen with certain USB controllers). This completes registration between the device and UPDD. ***DO NOT cancel the wizard*** as it is possible the touch device will not work. More information on this and other USB issues relating to UPDD install is available here.

The driver will, where possible, automatically take control of USB devices configured in the package. However, if another OEM driver is installed to control the device the UPDD entry will show an error in the device manager as UPDD will not automatically try and remove an existing driver.

To rectify this problem uninstall the existing driver software or remove the device created in the device manager by the other driver and reinstall UPDD. If this is not possible manually associate UPDD with the device.

4.   The installation procedure is used to install the software for a single touch screen / UPDD supported pointer device. In a multi-touch screen or pointer device environment invoke the UPDD Console - Hardware dialog to add additional non PnP devices after installation. USB (PnP) devices, supported by the driver, will be added automatically as they are connected to the system; however you will need to adjust the desktop association to newly discovered devices. See the Multi monitor and multi device documentation for further information.

5.   **Windows XPe**
If UPDD is installed on an XPe system using the standard setup.exe file, typically to test the driver prior to embedding, UPDD installation has certain dependencies that need to be present for successful installation. One of the less obvious dependencies is the use of RegEdit during install to set up the registry. If RegEdit is not present all selections and dropdowns are blank.

6.   **Windows 8.x – *very important note***
UPDD driver can be used to allow legacy, non HID and non certified touch hardware to work in Windows 7 and 8 systems, both in mouse emulation and HID compatible mode – the latter allowing all the built in Windows 7 and 8 extended touch features to be utilized. To allow legacy hardware to function in the HID extended touch mode UPDD implements a Virtual HID device through which it posts all touch data.

However, we are aware that some OEM System Builder editions of Windows 8.1, aimed at dedicated touch systems (Notebooks, Tablets etc) now require touch hardware to be certified for Windows 8 and 8.1, based on completion of WHC tests with the certified code held in the device's firmware. Systems built using OEM system builder guarantee that the touch device has passed all the stringent touch tests and requirements and will deliver the best end user touch experience.

When using UPDD pre 5.1.1239 in these system Windows will recognize that the Virtual HID device is not WHC certified (it is digitally signed, having passed the HCK test, but this is a different signature) and extended touch features for this device will be disabled. However, with UPDD 5.1.1239 and above we use a different touch interface to post touch data into these OEM versions of Window 8.x which reinstates touch functionality for UPDD supported devices. This interface is used on all Windows 8.x systems.

7.   **Very important PS/2 notes**
The current implementation handles the device as a proper PnP PS/2 device. However, there are some restrictions:

a)   The PS/2 device must be selected during installation; it is not a hot pluggable device (unlike other PnP devices) and we have difficulties with manual addition so the PS/2 devices (like other PnP devices) are not shown as a device that can be added manually. USB PnP devices are automatically added by the new UPDD PnP interface.

b)   It is recommended that the system be booted with the ps/2 device plugged in prior to install (same as for a normal mouse). We are concerned that some old BIOS's will release the PS/2 interrupt if no PS/2 device is seen on the system.

c)   During extensive tests we found that on a couple of occasions the device did not work after install. We have not been able to reproduce this install issue. If you do find that the touch screen is not working after install then:

   1.   Check for a PS/2 UPDD entry in the Device Manager.
   2.   If a UPDD entry is not seen use the hardware wizard to update the driver for the ps/2 touch device to UPDD. Note – once or twice we got an error "Not enough memory to complete this request" whilst switching drivers this way. It always worked on the 2nd attempt.

8.   **Signed drivers**

Our driver and utility programs carry an Authenticode Certificate. An Authenticode digital signature guarantees that the software was produced by the individual or company named in the certificate, which has been verified by the authority that issued the certificate.

However, not all builds of our driver will be WHQL digitally signed. Windows will automatically show a warning dialog if a digitally unsigned driver is installed. Some UPDD installs, that are installing an unsigned driver, have been configured to automatically handle the warning dialog. This results in a 'dialog flash' during install. Other UPDD installs have been configured to not handle the warning dialog, leaving the user to handle the dialog, as seen below;

Windows can be configured to not allow the installation of unsigned drivers. Drivers have been identified as one of the major causes for the past instability of Windows. Drivers can be submitted to Microsoft to be approved and 'digitally signed' to show that they have passed specific driver acceptance tests. UPDD cannot on its own be submitted for approval. It can only be approved with the pointer device. Where UPDD has been

submitted to Microsoft with appropriate hardware it has passed all tests and been approved and digitally signed. You may or may not be installing a signed version of this driver but you should not be unduly concerned if you are installing an unsigned version. For more information the WHQL and digital signature documentation.

Given that signing is related to the driver and controller combination we can either supply a driver that only supports a signed controller(s) or the driver may support many controllers, some signed and some not. In this case the driver is only signed when using a signed controller. If a supported controller is used, that has not been signed, the system will indicate that this is not a signed combination.

9.   **Sysprep utilization**
Sysprep is a tool that is designed for system administrators, Original Equipment Manufacturers (OEMs) and others who must automatically deploy the Windows operating system on multiple computers.

If Sysprep is installing a Windows image which includes UPDD you must enable the kernel mode operation prior to preparing the system for sysprep.

10. **Very important BIOS settings issue**
A hardware related issue has been reported on some systems with the BIOS settings of UPC and PLD. This has been seen on some systems running Windows 7 that have been reverted to XP. When these systems hibernate or standby they do not return as expected and the system appears totally dead. Investigation revealed that the driver handles and passes on a IRP request (as expected in power management processing) which is never completed and the driver 'hangs' waiting for the completion notice. We suspect this may be a BIOS issue on these particular systems. However, we have experimented with not passing on this IRP request and in tests this did not have any ill effect and the system resumed correctly. The following DWORD registry entry can be used to block the driver's IRP forward processing:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tbupddsu\NoForwardPower=1

A .reg file to set this setting is available here.

Update – 27/09/10 – We discovered on some systems that when returning from hibernate (deep sleep) the USB device was not powered up if the IRP was not processed. The driver can detect that the USB device is not powered up and force a stop / start of the device to initiate power up such that the touch continues to work.

Note: We would consider this function to be a hack to overcome a hardware issue and should be used with caution.

Installation Procedure

In the majority of cases the supplied setup program will be invoked to install the driver. In some situations, such as installing the minimum component, you may want to control the files that are installed on the system and this is covered in Manual install section

Using the supplied set up program

Installation of the driver must been performed whilst logged on as a user with Administration rights. Please note this is not the same as running as an Admin user. The user account must have Administration rights. If you do not have these rights then in some OS you can right click on the setup.exe program and select 'Run as administrator' and enter the appropriated credentials as requested.

Expand the .ZIP file or download the setup program from the http link provided and then run the appropriate setup program. Run either the setup32 or setup64 depending on your system's architecture.

Prior to installing, you should uninstall any UPDD version 3 or 4 driver. You can optionally install over previously installed UPDD version 5 drivers.

Once invoked, the install initially checks that the OS is supported will issue a notice if not and cease installation. It then checks that the driver has support for the system's architecture. E.g. installing a driver only built for 32 bit system will not install on a 64 bit system. In the latter case you need to request a setup with the required support.

If the OS is supported the standard install (running the setup program without any parameters) will show a list of supported controllers in this build.
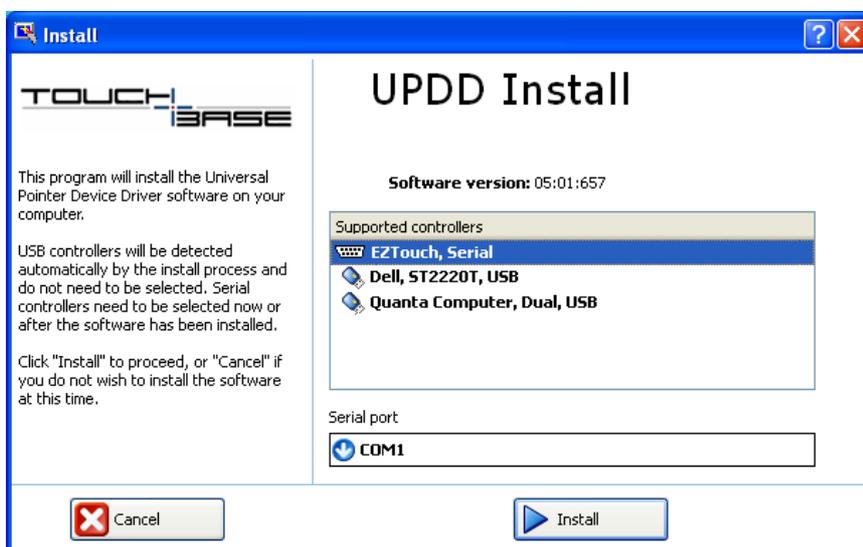
In the above example the supplied driver supports 2 USB devices and 1 serial device.

If installing for a non PnP touch device, such as a serial device, you need to select the controller from the list. PnP devices will be automatically discovered by UPDD driver so there is no need to select. If you select a serial controller you will be able to define the serial port. Additional non-PnP devices can also be added after install as required.

Keyboard hotkeys, via the Alt key, can be used to control the install dialog. Up and down arrow keys can be used to traverse the controller list, Alt C to cancel and Alt I to continue with install.

During install the system may issue certain warning dialogs, such as the installation of a non signed driver (if installing a non signed version of our driver!) and under Window 7 User Access Control dialogs to confirm updates to the system are allowed. If at any time during the install or after the first reboot the system asks for the location of the driver to complete install the folder to be selected is <program files>\updd (this should not occur!).

Once the install has completed and all being well the touch should now be active albeit not calibrated. If the cursor does not react and calibration fails then try rebooting the system – a reboot is currently needed for Win 8.x touch interface to work. If touch is still not working after install and reboot please refer to the troubleshooting section below.

Setup Install Options

-s   **Silent install mode**

Running setup –s indicates that no UPDD install dialogs are to be shown. Default settings for the touch controller are used unless a clone file is found in which case the settings in the clone file are used.

Only the first serial or Virtual device, if any are supported in the driver bundle, is configured. In both case, PnP devices will be handled as soon as they are discovered by the driver.

-n   **No default serial/virtual controller** *with silent option*

Indicates that there is no default serial or virtual device (if supported in the bundle) configured automatically. In this case PnP devices will be automatically detected but serial or virtual devices, if supported in the package, will need to be added automatically.

This can be combined with the silent option by specifying "-sn".

-r   **Reinstall** *with silent option*

Starting with version 5.1.702 installer has a new command line option "-r" for reinstall.

This can be combined with the silent option by specifying "-sr".

The reinstall option

1.   Does not remove / reinstall devices.
2.   Does not replace the settings file.
3.   Provide default values for new setting values. Existing settings are untouched, however if between the old and new version new settings have been introduced then suitable defaults are added for these. This is unlikely to be significant functionally. Performing a comparison of previous and new settings files might show differences attributable to this.

This provides for an upgrade installation in many scenarios.

Notes:

1.   Due to the wide range of legacy versions and software configurations that could be upgraded this process cannot be guaranteed. If using for a business critical rollout then a check must first be made that a successful upgrade is achieved for each version to be upgraded.

2.   The silent option will hide all dialogs, with the exception of any critical errors, including a restart warning message that could be issued due to a number of factors, such as the inability to stop the current device driver:
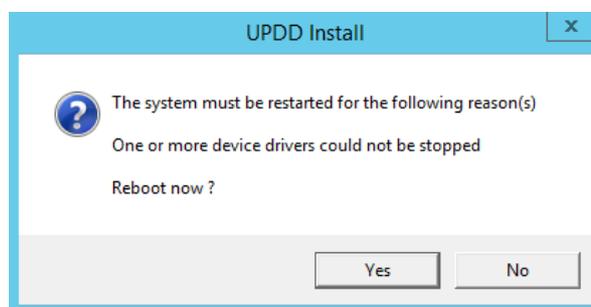
This being the case it is advisable to restart the system after installing with –sr.

Important silent install mode considerations: Silent install suppresses UPDD install dialogs. However there is always a possibility that system dialogs will be shown, either because you are installing an unsigned version of the driver or because UAC (Windows 7 user access control) settings dictate that the system seeks permission before changes are made to the system by the installer. Total silent install can only be achieved if the driver is signed and UAC is configured such that no user prompts will be issued or run with the correct administrator privileges.

-np   **No progress bar**

Progress bar suppression

Clone   Duplicate (clone) settings

Once installed the driver can be configured via the UPDD Console. These setting can be replicated (cloned) to a file in the UPDD Console – Status dialog – Dump settings option. This process will create a .reg file (4.0.x) or .ini file (4.1.x) which, for cloning purposes, should be renamed to upddclon.reg (UPDD version 4.0.x) or tbupdd.ini (UPDD version 4.1.x)

To use this file during UPDD setup, create a folder to hold the updd setup program (setup.exe) and a sub-folder UPDD_EXT within this folder to hold the clone file.

E.g. To perform a 4.1.x clone install:

C:\temp

Setup.exe

C:\temp\UPDD_EXT

Tbupdd.ini

When setup is invoked it will find the clone file and perform a silent install using the settings within the clone file. The setup.exe should **not** be in a folder called c:\program files\updd

Note:- This method of cloning has certain limitations and is being replaced with a custom settings file as described below.

The limitations with this method are that the clone file is only processed during install for the controllers defined in the clone file. Controllers plugged in after install will not inherit these settings. The settings file also has a fixed link to the location of UPDD application folder which could be different on a different language system, i.e. c:\program files\updd – English, C:\Programme\UPDD – German. There are other issues, hence being superseded.

File   The setup program will copy any files placed in sub folder UPDD_EXT into the UPDD application folder during install, replacing any that previously existed. This can be used to update files contained within the setup program.

CDS   Customised driver settings – *available since UPDD 5.1.827*

The install package will create UPDD settings file tbupdd.ini. This contains the default driver, controller and calibration settings. To install with user defined custom settings we now allow end users to customise settings in a file - extra.ini. This file was previously embedded in the setup for customised builds. This file can now be placed in the subfolder .\updd_ext relative to the location of the setup*.exe file on the installation media allowing most settings to be customised by an integrator.

This will allow, for example, calibration data to be created (tbcalib dum4tba) from a calibrated system for use in subsequent installs.

Manual install

This install could typically be referred to as an .INF type install, that is, using the .INF file to install the kernel device driver and its dependant components. In the case of the UPDD driver a few more elements are required, as discussed below:

You will initially have to install on a system using the setup.exe to extract all the components from the UPDD installer. You can then follow these instructions to create a minimum install:

1) Create a UPDD application folder (typically <program files>\updd)

2) Copy all UPDD components (from a working system) to this folder

3) Copy tbupddsu.sys to the Windows driver folder (e.g. <windows>\system32\drivers)

4) Install using the appropriate inf file (e.g. xxxxv3mo.inf) in <program files>\updd folder (by using whatever strategy you use). See list of available .inf files below.

5) Install the tbupddwu service with the command Tbupddwu –I 4141

6) Modify the service entry to "Restart the Service" in an event of a failure. To do this you can run the wuauto.reg file.

7) Invoke Aidaemon and TIdaemon (Win 8.x) processes.

Notes:

1) In cases where you want to minimize the number of files, leaving the bare minimum to allow the driver to function in the UPDD Application folder we believe you can delete most files in the UPDD application folder and leave:

File	Usage
Tbupddsu.sys	Kernel mode driver component
Tbupddwu.exe	User mode driver component – runs as a service
Tbupdd.ini	Settings file
ACExxxxxxx.DLL	Inter process communication library
QT-xxxxx.DLL	QT graphic library

These files must all be in the same folder.

2) In situations where you need the driver to run from a different folder you can update the setting 'updd folder' setting in the tbupdd.ini file (which will eventually be located in the new path!)

[updd\parameters]
updd folder=C:\Program Files\UPDD – specify the required pathname

Remember to copy the required files into this alternate path.

3) The .INF files that are used in the UPDD package are as follows. Select the one(s) that is(are) most suitable for your needs:

INF file	Usage
Updd3vmo.inf	32 bit unsigned
Upd63vmo.inf	64 bit unsigned
Updd3xxxx.inf	32 bit signed (xxxx = OEM id for signed controller/driver combination)
Upd63xxxx.inf	64 bit signed (xxxx = OEM id for signed controller/driver combination)
Upddhid.inf	Virtual UPDD HID device for Vista, Windows 7 and 8 extended touch usage. The upddvh.sys file should be placed in the Windows driver folder.

Desktop and application touch interaction

Once the driver is installed and calibrated then the most basic touch interaction resulting from the touch screen being used is to perform mouse emulation such that a touch on the screen emulates a mouse button down, mouse movement and button up. In many cases this will satisfy simple single touch usage. However, there are a number of touch interfaces that can be utilised within our driver, as described below:

Interface	Brief description
Mouse emulation	The most basic, single touch interface, available in all Windows OS. Touch data is posted to the OS via the mouse interface.
Gestures Virtual HID interface	For Windows 7 we utilise a Virtual HID device to post touch data to the OS such that the extended touch features native to the OS are enabled and accessible via the UPDD controlled device, irrespective of type of touch device in use (USB, Serial, PS/2). Utilisation of the UPDD Virtual HID interface can be disabled by reverting to mouse emulation mode in the UPDD Console, by deselecting the extended touch option, in which case the UPDD touch screen will not utilise the extended touch features.
Win 8.x touch interface	Certain builds of Win 8.x, such as OEM system builder, only work with approved touch hardware that has passed stringent Win 8.x HCK tests and have embedded hardware compatibility signatures. It is for this reason that some HID compliant touch hardware will not work in these versions of Win 8.x systems. These systems also reject the UPDD Virtual HID (software emulated HID device) implementation that allows legacy touch hardware to function in Win 8 systems. However, there is a specific Windows 8 API based touch interface that allows touch data to be posted to these systems which is now available in UPDD version 5.1.1248 and above for all Windows 8 systems

When UPDD is installed with a default configuration it creates up to three different interfaces to post touch data to the OS as follows:

OS	Interface	UPDD Settings
ALL	Mouse port	XP – Only interface used so no UPDD Settings
	Single touch mouse device	Works across all desktops, login, screen saver & main desktop
Win 7/8	Virtual HID	Works across all desktops, login, screen saver & main desktop
	Software implemented multi-touch HID device	
Win 8	Win8 Touch Interface	Works on main desktop only
	Multi-touch user mode API	

Win8TI interface works on the main user desktop so one of the other interfaces must be used during login screens and any other 'secure desktop screens' and on the desktop whilst waiting for the Win8TI interface to load. Given that some Win 8 OEM systems will block the Virtual HID interface then by default the mouse interface is used as the alternative interface this is known to work in all Win 8 systems.

Please note that when the mouse interface is used at the login screen you have to manually invoke the on-screen keyboard, it does not show automatically when the login field has focus

Notes:
1) **System touch support notification**
   When Win8TI is installed as the only touch interface Windows will show the touch implementation to be 'limited' due to the fact it is not implemented via the hardware / HID interface. If other touch devices are registered with the system (physical HID device, UPDD Virtual HID etc) a different Pen and Touch notification might be shown.

System		
Processor:	AMD A4-1250 APU with Radeon(TM) HD Graphics	1.00 GHz
Installed memory (RAM):	8.00 GB (7.47 GB usable)	
System type:	64-bit Operating System, x64-based processor	
Pen and Touch:	Limited Touch Support with 16 Touch Points	

However, it is our observation that the Win8TI implementation offers full touch functionality.
2) **Interface Settings**
   A number of settings can be used to affect the behaviour of the UPDD Win8 Extended Touch interfaces as described below. Settings are held in the UPDD settings file and can be adjusted using the UPDD Command Line User Interface.

e.g. tbutils nodevice setting dw dontinstallupddvh 1

The Windows 8 Extended Touch interfaces (Virtual HID or Win8TI) are only used if the UPDD Console, Click Mode, Extended touch check box is enabled otherwise the single touch mouse interface is used.

Win8TI settings

Setting	Meaning
useupddvh	If set (1) use the Virtual HID as alternative interface.
dontusetouchinterface	If set don't use Win8TI.
	If the Virtual HID interface is available (and known to working) then this setting can be used to prevent switching to the Win8TI and to use Virtual HID on the desktop. If the VHID interface is unavailable the mouse interface (single touch only) is used On non-Windows 8 systems this settings is ignored.

Virtual HID settings
For Win 8 OEM systems where it is known that the Virtual HID interface is blocked then the 'dontinstallupdddvh' setting can be used to request that the driver does not register the Virtual HID device(s) with the system. If this setting is created after the device has been registered then the device(s) will need to be manually deleted.

Right click on each UPDD VHID entry and select Uninstall.

This setting is intended to be preset in UPDD builds that do not want the Virtual HID device(s) to be created.

Virtual HID versus Win8TI
On Win8 systems whereby both the Virtual HID and Win8TI are known to be working there are pros and cons to using either interface as follows:

Interface	Pros	Cons
Virtual HID	Extended touch operations are supported at system startup, i.e. the onscreen keyboard will be invoked automatically.	If multi monitor operation is also required then the Tablet PC setup option must be utilized to identify the touch screens.
Win8TI	Multiple monitor setup is simplified as this is handled entirely by UPDD removing the need to use the Tablet PC setup options of Windows	Extended touch operations are not supported at system startup, i.e. on the logon screen and until the desktop components have loaded completely.

Current restrictions with the Win 8.x Touch Interface
- Is implemented in a separate process TIdaemon that must be running.
- This interface does not work on the login screens or other secure desktops. One of the other interfaces (mouse / virtual HID) will be handling the touch in these cases as described above.
- It requires a reboot after install:
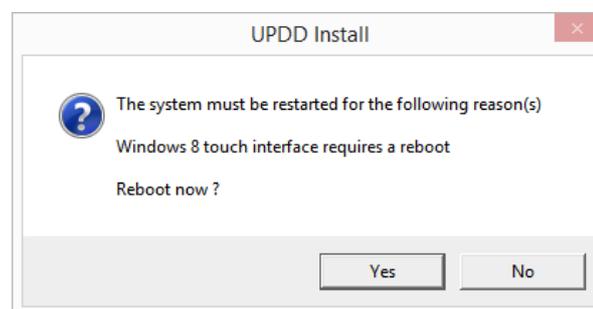
We hope to resolve the reboot issue in a future release.

TUIO	Allows TUIO Client applications to receive touches via the UPDD TUIO server interface.
UPDD API	Allows UPDD API based touch applications to receive touches via the UPDD API.

UPDD driver utilities

The installation process not only installs the driver but also a number of driver related utilities.

The main utilities are the UPDD Console (settings interface), Calibration utility, Test program and the Command Line interface. These are placed in the UPDD applications folder c:\program files\updd and some can be invoked from the Windows Program Manager and System Tray.
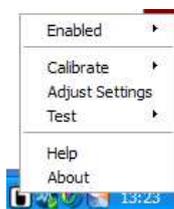
Program Manager

System Tray

Menu bar item
Introduced in release 5.1.x the UPDD Menu bar item offers links to the main driver utilities:

Indicates if the driver is to process the touch data for the device – use with caution if touch is the only interface!
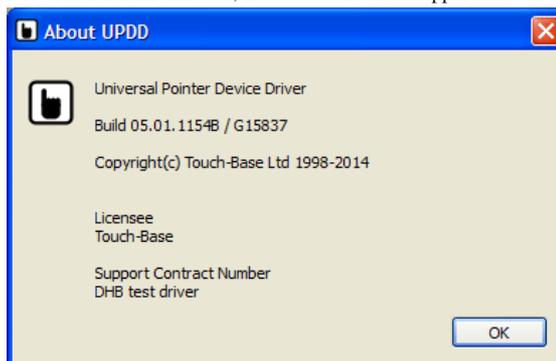
Invokes the touch screen calibration program, UPDD Calibration

Invokes the driver settings program, UPDD Console

Invokes the driver test program, UPDD Draw

Invokes the help system

Driver version information, Licensee name and Support Contract Number

About UPDD

Universal Pointer Device Driver

Build 05.01.1154B / G15837

Copyright(c) Touch-Base Ltd 1998-2014

Licensee
Touch-Base

Support Contract Number
DHB test driver

OK

Calibration

Calibration is a procedure used to align the pointer device with the graphically display area or desktop segment. When using the pointer device the mouse cursor should normally position itself under the stylus when it is in contact with the pointer device. If this is not the case then calibration will be required and this is described in full in the Calibration document. The UPDD driver also supports Toolbars, which also require calibrating, and this is covered in full in a separate Toolbar document.

Console

The driver and device settings can be adjusted with the UPDD Console program and is described in full in the UPDD Console documentation.
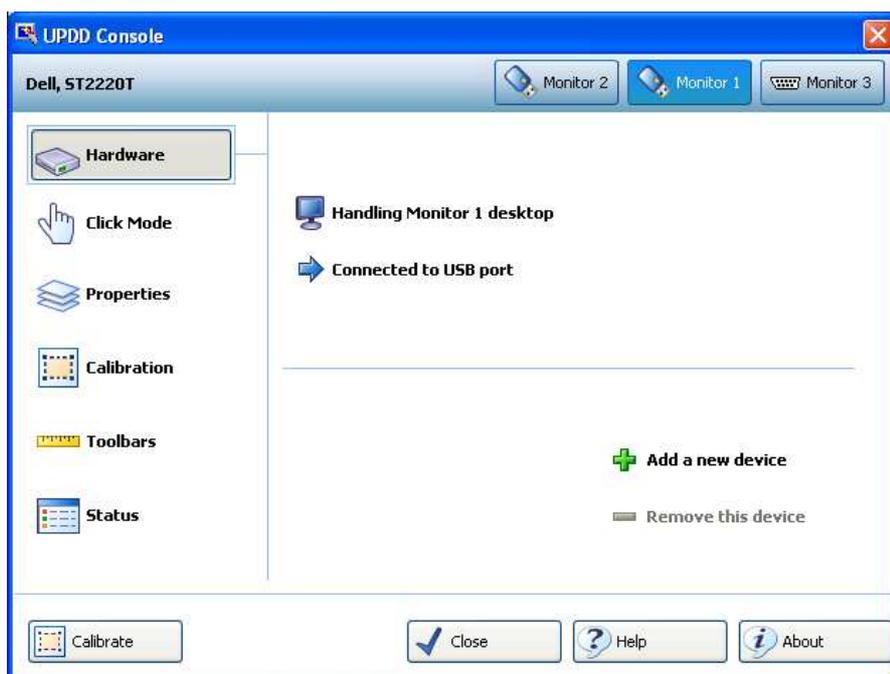
Following a successful install the UPDD Console will list the configured and connected devices and allow changes to the driver settings as required.

In the above example a serial device has been manual selected (during install or using the 'Add a new device' option) and the driver's PnP manager has detected two USB devices. The three touch devices have been configured to control individual monitors.

Test program

A simple test program is supplied to test the performance and accuracy of the touch device as handled by the UPDD driver. It can be used to view stylus input from single and multi-touch devices.

Command Line interface

A command line utility is supplied that can be used to perform certain configuration or device type functions.

Useful driver settings

There are a number of useful driver features that may be disabled by default by nevertheless are useful in some cases as highlighted here.

Edge Acceleration

It may be required to be able to push the cursor to the extreme edge of the screen, especially to invoke features that are associated with the cursor reaching the extreme edge such as with Win 7 and Win 8 extended touch mode. To cater for the need to be able to place the cursor to the extreme edges there is an advanced setting in UPDD (UPDD Console, Properties, Advanced) to accelerate the cursor toward the edge of the screen as the stylus approaches the edge. The values shown below work well:
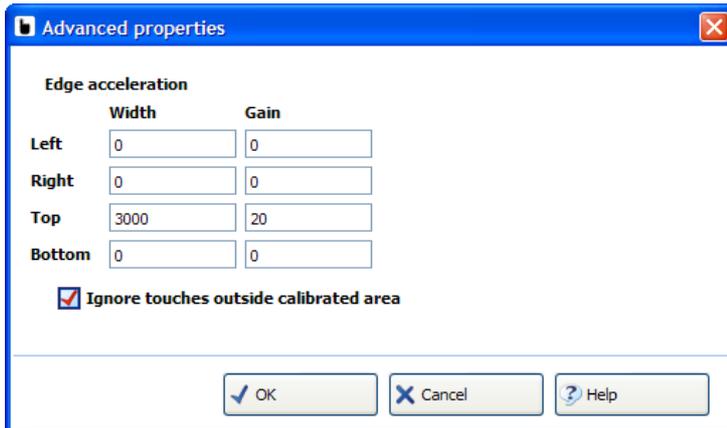
These settings are useful to invoke any application or function that is invoked by placing the system point to the edge of the display.

A collection of other general driver settings are described here.

Serial port issues

Change serial port connection

The UPDD Console - Hardware tab allows the COM port name to be reassigned after installation.

Serial to USB adaptors

If using a serial to USB adaptor, select the appropriate name in the COM port selector.

Serial port testing

Should the serial port connection not be working there are a number of procedures to follow to help identify the problem as described in the knowledge base article here.
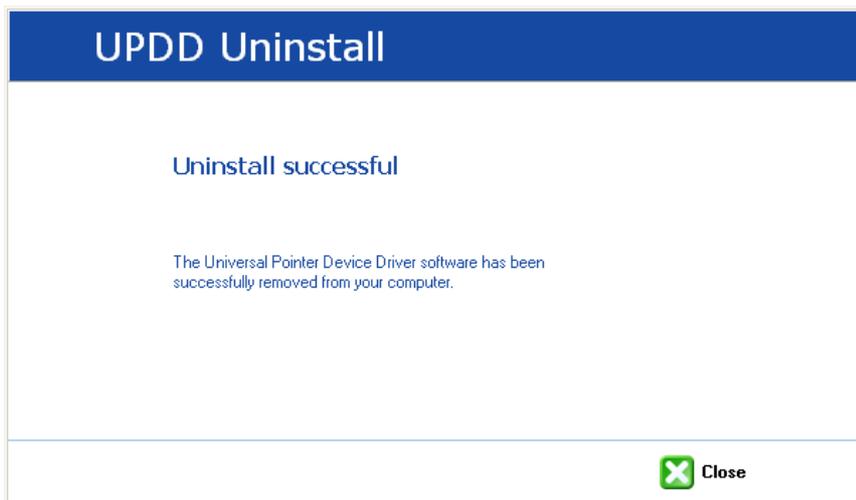
Uninstall

The driver can be removed via the Windows Control Panel, 'Add/Remove Programs' or 'Programs and Features – Change/Remove' option:
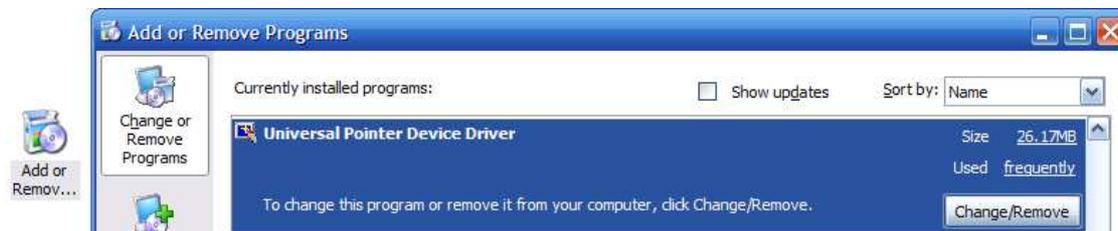
Select the 'Universal Pointer Device Driver' entry in the Add/Remove Programs dialog.

The automatic uninstall program can also be invoked by calling the program …\updd\uninstall.exe. For Silent uninstall, with no user dialogs being shown, pass the parameter UNINSTALLS i.e …\updd\uninstall.exe UNINSTALLS. Note – earlier versions of the uninstall program needed the parameter UNINSTALL to be passed to perform the normal uninstall procedure. Running an earlier version without any parameters will invoke an install – don't ask!

Please note: The uninstall process *does not have* any progress bar and can take upwards of 1 minute to complete, depending on system speed. On completion the following dialog is shown:

Manual

To manually uninstall the driver components you need to:

- Delete the UPDD application folder
  e.g. c:\program files\UPDD

- Remove any registry branches at HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ starting with TBUPDD (i.e. tbupdd, tbupddwu, tbupddsu)

- Delete any UPDD files in c:\[windows]\system32\drivers where [windows] is the main windows folder.
  (any files starting with tbupdd or updd i.e. tbupddsu.sys, upddvh.sys etc)

- Delete the UPDD Program Manager entry (Start, All Programs, right click UPDD entry, Delete)

- Delete any UPDD related files in c:\[windows]\inf where [windows] is the main windows folder.
  (any files starting with UPD i.e. upddv3mo.inf/cat/pnf, upddhid.inf/cat, upddhf.inf, upd6v3mo.inf/cat/pnf)

Following the above a reboot is required to remove the driver and daemon processes.

General Notes

Driver Components

The standard install procedure installs the following components:

Description	Name	Location
Application folder	UPDD	%program files% - typically C:\program files
Driver - kernel mode component	Tbupddsu.sys	Windows driver folder. E.g. %WINDIR%\system32 \drivers
Virtual HID Driver (Vista/Win 7/ Win 8)	upddvh.sys	Windows driver folder. E.g. %WINDIR%\system32 \drivers
HID touch interface used by UPDD extended touch features		
Driver - user mode component.	Tbupddwu.exe	%program files%\UPDD
Runs as a Windows service ("tbupddwu –i 4141" – installs the driver as a Windows service)		
UPDD Settings file	Tbupdd.ini	%program files%\UPDD
UPDD settings file backups	Tbupdd.bak.*.ini	%program files%\UPDD
UPDD Console	DCU.exe	%program files%\UPDD
Calibration	Tbcalib.exe	%program files%\UPDD
Test program	Updddraw.exe	%program files%\UPDD
Utilities program	TButils.exe	%program files%\UPDD
Windows background process	Tbdaemon.exe	%program files%\UPDD
No longer loaded since 5.1.1143		
Cross platform background processes	Aidaemon.exe	%program files%\UPDD
Win 8.1 touch interface daemon process	Tidaemon	%program files%\UPDD
Onscreen annotation	Activedw.exe	%program files%\UPDD
Event Selector application	Tbalt.exe	%program files%\UPDD
Setup Information Files	*.INF (and *.PNF)	%program files%\UPDD
Application Programming Interface	TBAPI.*	%program files%\UPDD

Language files	*.QM	%program files%\UPDD
Uninstaller	Uninstall.exe / .dat	%program files%\UPDD
ACE inter process communications lib	ACExxxxxxx.DLL	%program files%\UPDD
QT graphics library	QT-xxxxx.DLL	%program files%\UPDD
Windows driver Framework	Wdfcoinstaller*.DLL	%program files%\UPDD
Needed to install upddvh.sys		
Updd coinstaller	Upddcoin.dll	%program files%\UPDD
Launches setupxx.exe if using an INF based installer		
Monitors low power state and screensaver mode (so first touch/click can be ignored in this state as you are touching a blank screen - has to register as a global hook and as such will appear in other processes)	Tbhooknn.dll nn = 32 / 64	%program files%\UPDD
Pointer Device control panel	Tbctlpnl.cpl	Windows folder E.g. %WINDIR%\system32
Cursor utility	Tbcursor.exe	%program files%\UPDD
Cursor files	*.CUR	%program files%\UPDD
Help file viewer	Helpviewer.exe	%program files%\UPDD
UPDD Console Help files	*.HTM	%program files%\UPDD
Help file Images folder	IMAGES	%program files%\UPDD
Toolbar folder	TOOLBARS	%program files%\UPDD

The following registry entries are created (Windows will create others as a result of installing UPDD) :

HKEY_LOCAL_MACHINE\SOFTWARE\Touch-Base	
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tbupddsu	Kernel mode settings
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tbupddwu	User mode settings
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\aidaemon.exe	Invoke background process
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\tbdaemon.exe	Invoke background process

The driver uses TCP/IP port 4141 to communicate between components using the API.

Processes

Name	Description	Start	Stop
tbupddsu	The kernel mode driver	Tbutils upddmode	Tbutils hidmode
tbupddwu	The user mode service	Net stop tbupddwu	Net start tbupddwu

Device Manager entries

Branch	Item	Notes
Human Interface Devices	UPDD Virtual HID (UPDDVH.SYS)	One entry per discovered monitor
Mice and other pointing devices	[Controller Name], (UPDD)	One per PnP touch device (e.g. USB)
Mice and other pointing devices	Universal Pointer Device Driver	To support non PnP device (e.g. Serial)

In the example below the Win 7 system has two monitors, hence two UPDD Virtual HID entries - one for each monitor, one USB touch device and support for serial devices (which would be connected to a Com port if it was configured)

Processes and Device Manager notes

Some OEM customers did not want the driver processes and Device Manager entries to be created if no devices are discovered (announced) within a certain period of time. To accommodate this request, under version 5.1.1107 and above the installer no longer installs the "Universal Pointer Device Driver" (non pnp) device entry as shown above. This is now performed by User Mode driver (tbupddwu) at the same time as the UPDD Virtual HID (upddvh) installation.

The installation of the non pnp instance and upddvh is executed only if the setting startupdetectionseconds is zero or does not exist OR one or more devices have been announced to the user mode driver. This setting is in [updd\parameters] and would normally be set in a custom build.

If startupdetectionseconds seconds elapse from the time the first api based program connects without a device being announced the driver stops issues an UNLOAD message to tell all api programs to terminate and 5 seconds later the driver stops itself with a service stop command.

This approach means that if no device is announced then no updd drivers or programs are loaded.

NB once the software detects a device this operation is not reversed if the device is unplugged (although by deleting the upddvh and non pnp entries

in the device manager this can be achieved).

NB a serial device is "announced" at startup if it is selected as an installed device.

If using 'tbutils nodevice' command line to set this value after install all Device Manager entries would need to be manually deleted and the system rebooted.

This option is mainly for OEM systems that install UPDD on a wide number of system but in some cases a UPDD supported device is not in use and in this case they do not want unnecessary entries in the Device Manager or Processes running.

Daemon tasks

The daemon tasks run in the background and perform a number of functions related to the driver and user interface. The driver will still function for normal touch usage without the daemon task running but may not work as expected when using some of the more advanced features.

Multi-monitor and multi-device support

Multi-monitor and multi pointer devices are supported with this driver and this functionality is covered in full in the multi monitor and device document, Windows section.

Display rotation considerations

Under Windows there are various methods for rotating the desktop and UPDD will work with most rotate implementations as explained in detail in the separate rotate documentation.
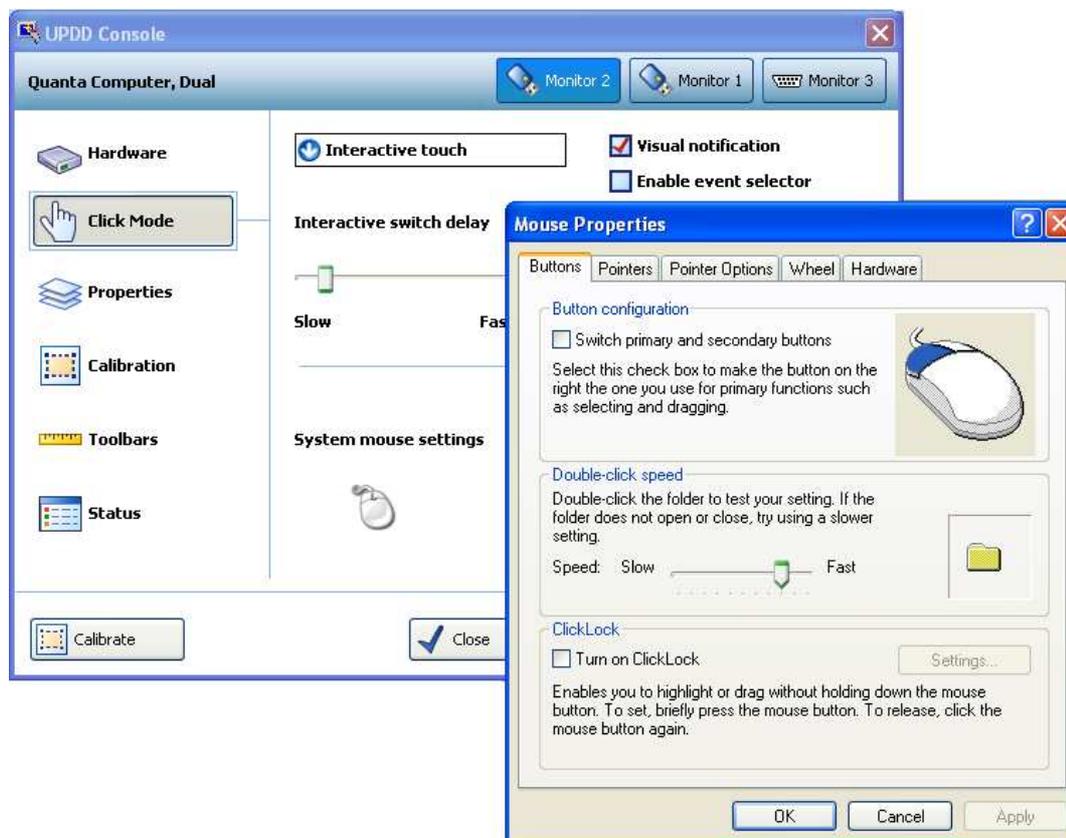
System Setting

Certain system settings may affect the functionality of the pointer device. This section highlights some of the settings that may need to be changed as required.

Mouse settings

Double click capabilities are affected by the system's Mouse settings. To achieve a double click using the pointer device these settings need to cater for the type of device in use. A touch screen may well require different settings to that required by a mouse. The main setting that affects the ability to double click is the double click speed. If this is set too fast it may be impossible to produce a double click. Ensure this is set to an appropriate value in the mouse settings to allow for double clicks via a stylus.

The UPDD Console, Click Mode dialog, System Mouse settings will invoke the Windows Mouse settings as shown in the following example:


More information on double click settings are available here.

Remote Wakeup settings

Some touch devices can be used to bring a system out of sleep/standby/hibernate. These sleep levels are referred to as power states and can range from S1 to S4, whereby S1 is the lightest and S4 the deepest sleep. Our driver supports power states S1 through S3. S4 normally requires the power button to be pressed to resume so external

devices are not capable of returning the system from this state.   If the device is capable of Remote Wakeup you also need to specify in the Device Manager (Start, Control Panel, System, Hardware tab, Device Manager) that the device can be used for this purpose, as shown below:
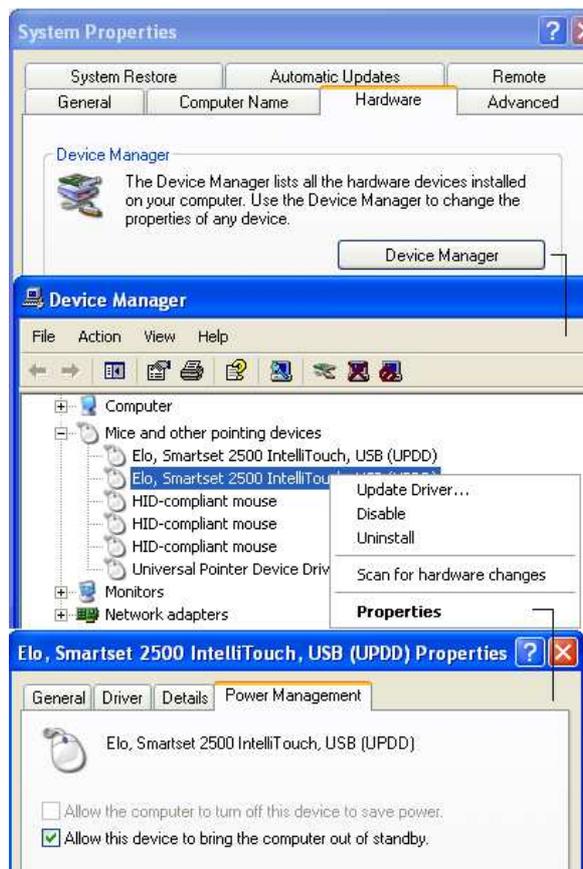
In the Device Manager, identify the UPDD device entry, right click to bring up the device menu, select Properties, Power Management and set accordingly.

To see the devices able to wake up the system use the command 'powercfg –devicequery wake_armed' as below:-

Remote Wakeup notes

1) The system must be configured to allow the device to bring the computer out of standby.
2) The device must be capable of interfacing with the system to wake it up. Not all touch screen controllers are able to perform this function
3) We have found that some environmental settings can affect the ability of certain devices to return a system from standby. In our tests under Vista we found that the ability of some devices can be affected by the "hybrid sleep" setting whereby the device can be used to return a system from standby, sleep or hibernation but not the new 'hybrid sleep' state.

To disable Hybrid Sleep, open up the Power Options setting from the control panel and select the "Change plan settings" links below the selected power plan:
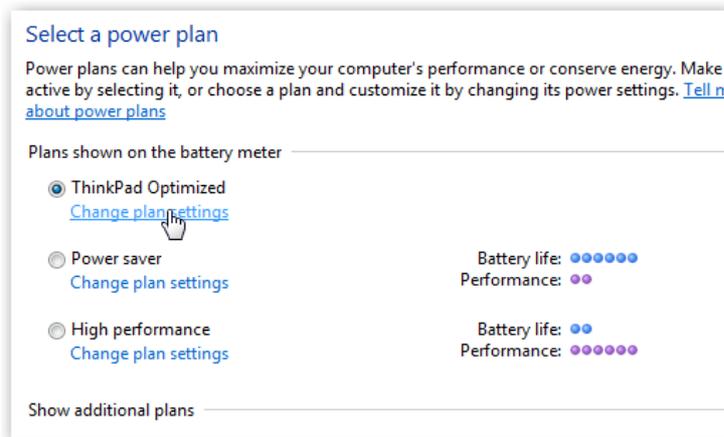
Once in the resulting dialog, choose "Change advanced power settings" and scroll through the options to find the "Sleep" node. Expand this and then the "Allow hybrid sleep" node below it. Set both the "On battery" and "Plugged in" to off

Touch Utilities

Virtual Keyboards

A number of Virtual keyboards are available on the Web for Windows as detailed in the UPDD Virtual keyboard documentation.

Mouse Cursor

A Windows cursor scheme settings program TBcursor is shipped with UPDD and installed in the UPDD Application folder.

Troubleshooting

Following successful installation and reboot of the system (if requested/required) the device should respond to UPDD calibration (a moving cursor does not always indicate UPDD driver is in control of the touch device). If this is not the case then follow these procedures to help identify the fault:

Check the device is seen by the driver

The UPDD Console will either show no device or list a selected device in Red text if the device is not found on the system or is being handled by another 3rd party driver:

Only when the device is listed in black text is the driver connected to the device. No / red listing can be because the device is not plugged in or if the device plugged in is not supported by the driver or the device is being handled by another 3rd party driver, such as the standard HID driver.

Ensure the device is plugged in and listed in the Windows Device Manager is that expected by the driver and also identify if the device listed (by Manufacturer, USB Vendor id and Product id) is that expected by the driver.

Uninstall any other 3rd party driver.

If you have just installed, try rebooting, especially if no name is listed in the console.

Very important BIOS settings issue
A hardware related issue has been reported on some systems with the BIOS settings of UPC and PLD. This has been seen on some systems running Windows 7 that have been reverted to XP. When these systems hibernate or standby they do not return as expected and the system appears totally dead. Investigation revealed that the driver handles and passes on a IRP request (as expected in power management processing) which is never completed and the driver 'hangs' waiting for the completion notice. We suspect this may be a BIOS issue on these particular systems. However, we have experimented with not passing on this IRP request and in tests this did not have any ill effect and the system resumed correctly. Since 4.1.6, build 1389 the following DWORD registry entry can be used to block the driver's IRP forward processing:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tbupddsu\NoForwardPower=1

A .reg file to set this setting is available here.

We further discovered on some systems that when returning from hibernate (deep sleep) the USB device was not powered up if the IRP was not processed. The driver can detect that the USB device is not powered up and force a stop / start of the device to initiate power up such that the touch continues to work.

Note: We would consider this function to be a hack to overcome a hardware issue and should be used with caution.

Contact

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