Support Procedures

This document describes the procedures to follow if you are experiencing a problem with our pointer device driver UPDD.

Please note that in most cases the advice in this document applies to our version 4.x.x drivers but specific references relate to UPDD 4.1.1 and above. Please contact Touch-Base if you need specific support assistance with 4.0.x version of the driver.

As the developers of UPDD we want to ensure as best as possible that UPDD will cause as few problems as possible. Unfortunately, given all the environments in which drivers need to operate, UPDD will occasionally fail to function as expected. It is at this point we are contacted to help and you will appreciate that we need as much information as possible to help identify the problem so we can best advise or try and reproduce the problem to allow us to investigate further. Reporting that a problem occurs on a particular operating system doesn't really help, we will need specific details!

Please read carefully and supply as much of the information requested. On receipt of this information we will endeavor to investigate your problem and report back ASAP. Thank you in advance for your co-operation.

Obvious checks

Prior to supplying this information, check the obvious such as cabling and power issues. Try and prove the hardware is working without our driver.

Serial

If the driver is installed set it to access a different serial port so that the touch screen port can be used by another application. You can employ the following methods to see if there is data being received from the serial port

- Windows – Use HyperTerminal or the free Touch-Base software data scope at ftp://scope:scope@ftp2.touch-base.com (Windows 2000/XP/ Vista users only) or a commercial data scope program such as Docklight (http://www.docklight.de)
- Linux - Open a terminal window and type cat < /dev/ttySN and use the serial device.
- Mac - Open a terminal window and type cat < /dev/[port] and use the serial device.

Serial port testing is further described in the knowledge base article here.

Once data is seen you need to ensure UPDD is configured for the correct serial port. UPDD will then see the data. If it sees the data expected, the mouse should move when touching the screen. If it moves, but incorrectly, try calibrating the screen.

If the data is incorrect, or the wrong UPDD driver is installed (for a different touchscreen), there will be no mouse movement but there will be sync errors in the UPDD Console Status dialog.

USB

With USB devices ensure the driver is loaded and the USB is recognized by the operation system.

Windows	**Device Manager**	

Check that UPDD is loaded in the Device manager along with the supported controller. Any icon with an indicator, such as a red X, exclamation mark, down arrow etc will indicate an issue.

E.g. Under Vista this is shown with a small down arrow on the icon

The driver will take control of devices configured in the package. However, if another 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. This is normally shown in the Device Manager with a yellow exclamation mark, as shown:
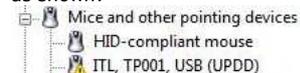
To rectify this problem remove the device created in the device manager by the other driver and reinstall UPDD.

A yellow exclamation will also be shown if there is an error on the system preventing access to the device.

Graphical viewer

Use a graphical USB Viewer available on the web (Google 'usb view download'), such as the one at http://www.microsoft.com/whdc/device/stream/vidcap/UVCView.mspx (this is the same as USB viewer)
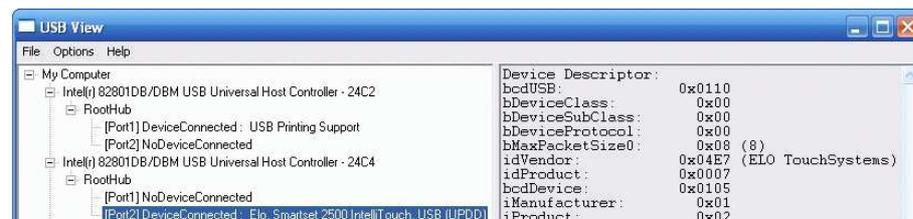
MacOSX   System Profiler

Use the system profiler to check the pointer device is recognised

Linux

Cat Command

Open a terminal window and type cat /proc/bus/usb/devices. This is best performed with all other USB devices unplugged as sometimes it is unclear which device is which, especially if the description string has not been implemented in the device.

```
T:  Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  3 Spd=1.5 MxCh= 0
D:  Ver= 1.00 Cls=ff(vend.) Sub=00 Prot=ff MxPS= 8 #Cfgs=  1
P:  Vendor=0afa ProdID=03e8 Rev= 1.10
S:  Manufacturer=TSC-10 DMC
C:* #Ifs= 1 Cfg#= 1 Atr=a0 MxPwr=100mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=ff(vend.) Sub=ff Prot=ff Driver=(none)
E:  Ad=81(I) Atr=03(Int.) MxPS=   5 Ivl= 10ms
```

Graphical viewer

Use a graphical USB Viewer available on the web (Google 'usb view download'), such as the one at http://www.kroah.com/linux/usb/ . When run, this will show the USB devices:

If the device is seen in Linux, and because we use the USB file system to access the port, it is possible to see if USB data is being received from the USB device as follows:
The sequence of steps to determine whether we are controlling the touch device are:-

-Open a terminal
-Type "cat /proc/bus/usb/devices"

A list of the USB devices plugged in will be printed. An example of a device would be:

```
T:  Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  2 Spd=12  MxCh= 0
D:  Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=14c8 ProdID=0003 Rev= 1.00
S:  Manufacturer=Zytronic Displays Limited
S:  Product=Zytronic, x-y, USB
S:  SerialNumber=0000006561
C:* #Ifs= 1 Cfg#= 1 Atr=c0 MxPwr=  2mA
I:  If#= 0 Alt= 0 #EPs= 2 Cls=00(>ifc ) Sub=00 Prot=00 Driver=(none)
E:  Ad=81(I) Atr=03(Int.) MxPS=  64 Ivl=1ms
E:  Ad=01(O) Atr=02(Bulk) MxPS=  64 Ivl=0ms
```

The information we have highlighted in red allows you to identify which device is the touch controller you are interested in. The information we have highlighted in blue will read "usbfs" if we have control of it.

If "usbfs" is in control and the controller is not outputting data then please do the following:-

-Open a terminal
-Type "sudo init 3" and enter your password when prompted, to stop X Windows from running.
-You will be presented with a text mode login screen. Log in.
-Type "cat /opt/tbupddlx/comReadPipe"
-Touch the screen

At this point you should see data being printed on the screen.

that TBUPDDWU is running:

For Wind you can use one of the USB utilit avai from http or http to prov the cont is func

Is the mou poin mov but un -calib In the case does calib work If not, it coul be that the HID drive or anot drive is in cont

In Wind does the UPD Cons Stat show som statu as n/a? This wou impl the user mod elem of the drive (tbu is not runn Che the Serv or Task Man to conf
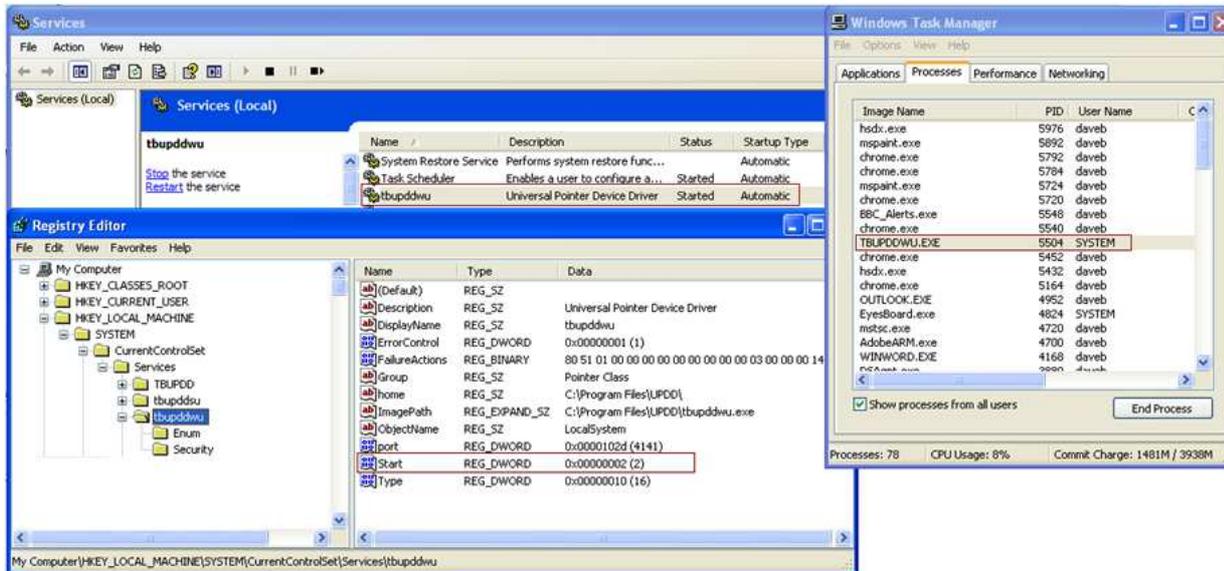
TBUPDDWU is the user mode portion of the driver and must be running in order for the touch device to work. It runs as a service and is set to run automatically (Registry key 'Start' = 2). Should, for any reason, the service fail or terminate it should 'in theory' restart automatically.

If it is believed that the hardware is functioning as expected re-enable the UPDD driver and run the test program from the UPDD Console, Status page and select the 'Direct' option. In this mode the test program uses the UPDD API to receive co-ordinate data rather than mouse emulation. If drawing is seen this implies the controller is functioning but for some reason the UPDD system mouse interface is not working.

If the device is correctly connected and functioning please answer the following questions as best you can (use N/A or Unknown where appropriate)

Support Information

Free Format
(Describe as best as you can what the problem is - list any error messages you see etc)

Operating System
- Detail the OS in use (Windows, Vista32, Vista64, CE4, 5 or 6, XPe, Linux, Mac OS X etc)
- For Linux indicate the distribution, kernel, Windows manager etc
- Indicate the patch status
- Indicate the locale (language)
- Any other features to be considered

Hardware related
- Is there anything unusual about the system in use?
- Is there any specialised hardware/configuration in use? E.g. Multiple monitors,
- Is the pointer device connected to unusual hardware? E.g. Serial to USB (to serial) converter.
- Are there other pointer devices installed and supported by other drivers?
- Is the any other driver that could be trying to claim the same hardware port/system resources in use by UPDD? (e.g. Serial printer/Modem driver connected to the same port)
- Do the ports in use by the UPDD driver show up in the system device manager or system profile?
- What pointer device controller and hardware port are you using?

Install problems
- Did the UPDD software install correctly?
- During install did you specifically select the hardware port to use or was the auto-detect check box available and enabled?
- Did you install as a result of a PnP message or did you run the setup program.

If the install appeared to be successful.....

Functionality
- In the UPDD Console, Hardware dialog is the correct port selected for the device?
- Can the device be calibrated? On the calibration screen do the points react to being touched (next point is displayed)?
- Does the cursor move when the device is in use?
- Does it work for a short while then fail? (e.g. after power management has hibernated the system or following a reboot?).
- Does it fail in some very specific situation?
- If there is no cursor movement does the device starts working after you the reinitialize or reload option in the UPDD Console, Status dialog?

Sequence of events leading to the problem
If the fault is obscure and is only seen after a sequence of events please describe the events that need to be performed in order to reproduce the fault.

Information request

Please supply the following information

1. In the UPDD Console, About dialog, please list company name and driver version number.
2. In the UPDD Console, About dialog, Support option, please list SCN (Software Support Number).
3. In the UPDD Console, Status please list the information shown.
4. In the UPDD Console, Status dialog, please select the 'Dump settings' option and attach it to this email.
5. In the UPDD Console, please list the name of the controller causing problems
6. If using USB, identify the vendor and product id of the device.
7. If using a serial device and the serial port connection appears not be working there are a number of procedures to follow to help identify the problem as described in the knowledge base article here.
8. If using a
9. Please list the devices shown in UPDD Console device list, are they shown in red or black (red indicating unplugged/not available)?

10. For Windows, send a screen print of the Device Manager 'Mouse/Mice and other pointing devices' branch. The easiest way to the Device Manager is to right click the My Computer icon or start menu entry, select Manage and select Device Manager:
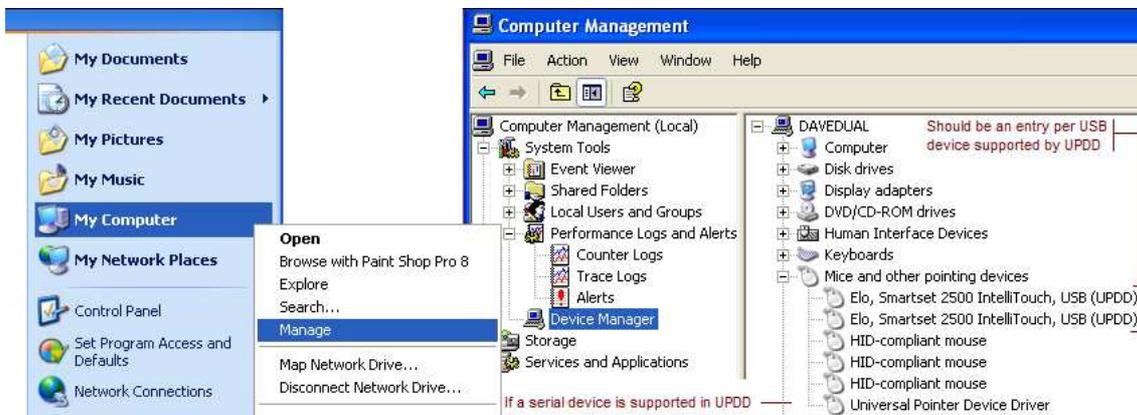
11. Please indicate the OS in use and its locale (e.g. English, French, Japanese etc)
12. If requested, please supply a debug trace as described below in the debug section.
13. Under Windows, if the error is causing a Blue Screen Crash – commonly known as a BSOD - we will need a system memory dump at the time the crash occurs. The dump is requested as follows: in the control panel -> system -> advanced -> startup and recovery. Select small memory dump. After a BSOD the dump file will be located in SystemRoot%\Minidump (or whatever was set in the control panel). Send the file related to the crash using the timestamp of the file to determine the correct file.
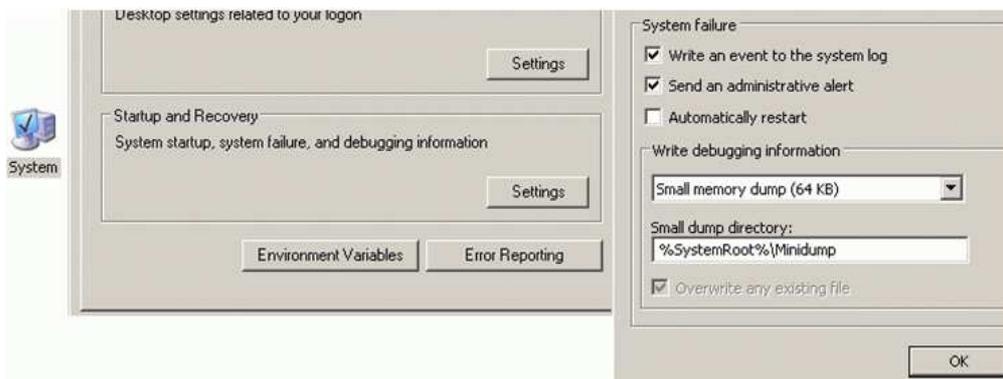
14. UPDD program symbols:
    These symbols are used in conjunction with identifying crashes within any UPDD modules and are mainly for internal use. However, occasionally technical users have asked for access to these symbols for their own debugging purposes. To this end we have made these symbols externally available on our symbol server at http://symbols.touch-base.com/Symbols.

    The symbol server holds symbols for all version 4.1.x release / production builds since 6 Nov 2008.

15. Under Linux please supply the following information:
    1. The outcome of typing "ps ax | grep tb" into a terminal.

```
                    andrew@ab12:~
File  Edit  View  Terminal  Tabs  Help
[andrew@ab12 ~]$ ps ax | grep tb
 2362 ?        S1     0:00 /tbupddlx/tbupddlx
 3213 pts/0    R+     0:00 grep tb
[andrew@ab12 ~]$
```

2. For USB controllers, the outcome of typing "cat /proc/bus/usb/devices".

```
                    andrew@ab12:~
File  Edit  View  Terminal  Tabs  Help
S:   SerialNumber=0000:00:07.2
C:*  #Ifs= 1 Cfg#= 1 Atr=c0 MxPwr=  0mA
I:   If#= 0 Alt= 0 #EPs= 1 Cls=09(hub  ) Sub=00 Prot=00 Driver=hub
E:   Ad=81(I) Atr=03(Int.) MxPS=    2 Ivl=255ms

T:   Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  2 Spd=12  MxCh= 0
D:   Ver= 1.10 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:   Vendor=0596 ProdID=0001 Rev= 2.90
S:   Manufacturer=MicroTouch Systems, Inc.
S:   Product=MicroTouch USB Touchscreen - EX II
C:*  #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=100mA
I:   If#= 0 Alt= 0 #EPs= 1 Cls=ff(vend.) Sub=00 Prot=00 Driver=usbfs
E:   Ad=81(I) Atr=03(Int.) MxPS=  32 Ivl=2ms

T:   Bus=01 Lev=01 Prnt=01 Port=01 Cnt=02 Dev#=  3 Spd=1.5 MxCh= 0
D:   Ver= 1.10 Cls=00(>ifc ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:   Vendor=00c8 ProdID=0003 Rev= 0.40
S:   Manufacturer=EELY Corporation
S:   Product=EELY elTouch
S:   SerialNumber=EWW-4WR-U
C:*  #Ifs= 1 Cfg#= 1 Atr=a0 MxPwr= 50mA
I:   If#= 0 Alt= 0 #EPs= 1 Cls=03(HID  ) Sub=00 Prot=00 Driver=usbhid
E:   Ad=81(I) Atr=03(Int.) MxPS=    5 Ivl=10ms
[andrew@ab12 ~]$
```

Note: On SuSE 10.1 systems (and possibly others) the USB file system isn't enabled by default. This should be enabled by editing the file "/etc/fstab" and change the line that says:

usbfs /proc/bus/usbfs noauto 00
to
usbfs /proc/bus/usbfs auto 00 and then reboot your system.

This is required to get the output of "cat /proc/bus/usb/devices"

Support Procedure

Based on the above information we will advise if we know of a fix or we will investigate further. We may require a debug log (see below) to be captured on the failing system. If we do not know what the problem is and cannot reproduce we may ask for the system to be sent to us if this is possible.

Debug Procedures

Difficult to reproduce or to identify UPDD problems sometimes require a debug trace to be captured. The debug trace is enabled within the UPDD driver and the debug output is sent to the 'standard out' port on the OS.

With UPDD versions up to 4.1.10, build 2113 debug messages were either On (DebugLevel setting = 5) or Off (DebugLevel setting = 0). However, starting with UPDD version 4.1.10, build 2114 the user mode component has 4 settings, 0,1,2 and 3 (and 5 to be backward compatible).

0 – no debug messags
1 – non-time critical messages
2 – non-time critical messages
3/5 - all messages including debugging in the interrupt path

Non-time critical messages will be allocated to either level 1 or 2 such as often repeated message, e.g. in polling functions.

Critical messages that can affect performance and critical path functionality, such as hardware interface interrupt procedures.

Note: The Windows Kernel mode component is still either on (5) or off (0).

Debug preparation

Windows

Under Windows we use a kernel debugger (normally DebugView) to capture the debug messages - software that is freely available on the web. Last time we looked it was available here. If not, Google DebugView.

There are two parts to the Windows driver, a kernel mode element – TBUPDDSU and a user mode element – TBUPDDWU. Both elements will output debug information if the Debug Level is set to 5. The kernel mode driver references the registry for it's debug setting and the user mode references the UPDD settings file for its debug setting, as follows;

Kernel mode debug setting - TBUPDDSU
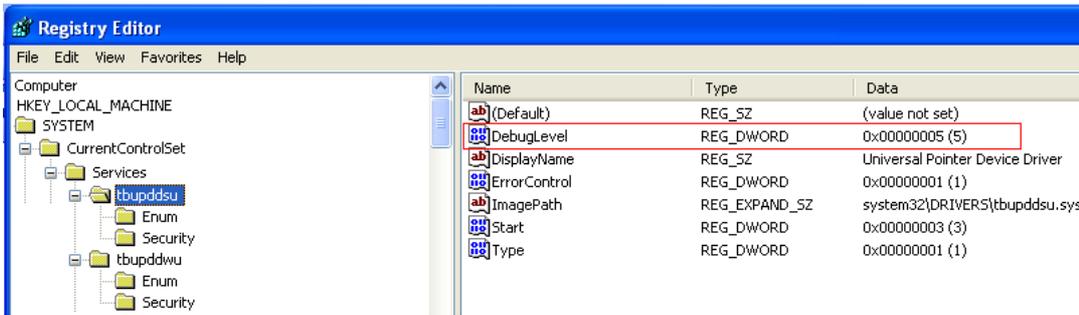
If the DebugLevel entry is missing, Right Click in the window pane, select New, DWORD value and rename the new key to DebugLevel:
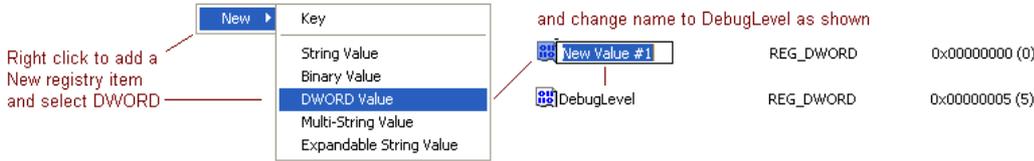
User mode debug setting – TBUPDDWU
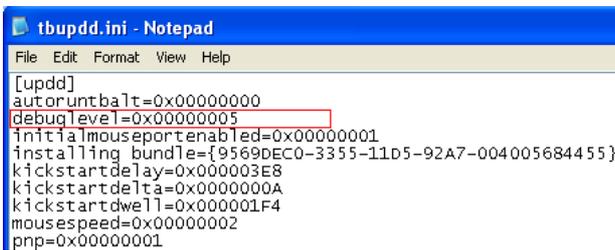
To **enable debugging**:

If requested to capture kernel mode debug messages use Regedit to set HKLM/System/CurrentControlSet/Services/tbupddsu/debuglevel setting to 5 and reboot the system (as shown above).

If requested to capture user mode debug messages use the following procedure whilst logged on as a user with administrator rights

1) If using UPDD 5.1.0 it is only necessary to run the command "tbutils global setting dw debuglevel 5". The driver will be reloaded and in debug mode.

   If not running 5.1.0 then

2) Stop the driver as follows:

- With UPDD 4.1.6, build 1200 and above (Start, Run) Net Stop TBUPDDWU
  or

- With earlier builds to **stop the user mode driver** you need to kill the TBUPDDWU process either via the Task Manager, end process:
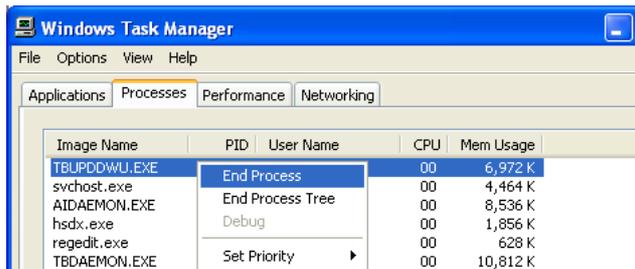
or, if the above does not work, change directory to c:\program files\updd and "kill /f tbupddwu"
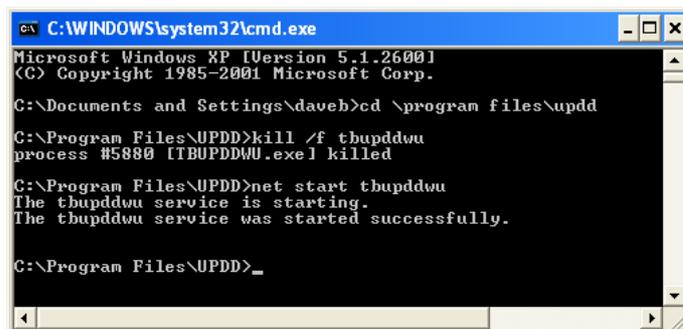
2) Use Notepad to set TBUPDD.INI file debuglevel setting to 1,2,3 or 5 as requested.

3) Reboot the system or **start the user mode driver** with the command "Net Start TBUPDDWU" as shown above.

Mac OS X

If requested to capture user mode debug messages use the following procedure:

For Mac OS X you can run the driver from a terminal (console) and view the debug log messages or redirect them to a file.

If using UPDD 5.1.0 it is only necessary to run the command "tbutils global setting dw debuglevel 5". The driver will be reloaded and in debug mode

If not running 5.1.0 then:

To **stop the driver** run up a terminal window (sometimes known as a Console) and type 'sudo killall tbupddwu'.

To **enable debugging**:

Open up a terminal window
cd /tbupddmx
cp tbupdd.ini ~/tbupdd.ini
open -a textedit ~/tbupdd.ini
Use the text editor to make the change, save and close
sudo cp ~/tbupdd.ini tbupdd.ini

To **start the driver** change to the UPDD folder (type cd \tbupddmx in the terminal window) and type 'sudo ./tbupddwu to see the debug log in the terminal window or 'sudo ./tbupddwu >updddebuglog' to redirect the console output to a log file.

Linux

If requested to capture user mode debug messages use the following procedure:

For Linux you can run the driver from a terminal (console) and view the debug log messages or redirect them to a file.

If using UPDD 5.1.0 it is only necessary to run the command "tbutils global setting dw debuglevel 5". The driver will be reloaded and in debug mode

If not running 5.1.0 then:

To **stop the driver** run up a terminal window (sometimes known as a Console) and type

'sudo killall startupdd'

'sudo killall tbupddwu'

To **enable debugging**:

Open up a terminal window
cd /opt/tbupddlx
cp tbupdd.ini ~/tbupdd.ini
open -a textedit ~/tbupdd.ini or gedit ~/tbupdd.ini
Use the text editor to make the change, save and close
sudo cp ~/tbupdd.ini tbupdd.ini

To **start the driver** change to the UPDD folder (type cd /opt/tbupddlx in the terminal window) and type

export LD_LIBRARY_PATH=/opt/tbupddlx:$LD_LIBRARY_PATH

sudo ./tbupddwu

to see the debug log in the terminal window or 'sudo ./tbupddwu >updddebuglog' to redirect the console output to a log file.

Debug message capture

Once debug is enabled, run up the kernel debugger under Windows or start the driver in a terminal Window under Linux/Mac OS X and recreate the problem and send the debug log to Touch-Base for analysis.

If capturing a kernel log and using DebugView please ensure the kernel capture setting is enabled:
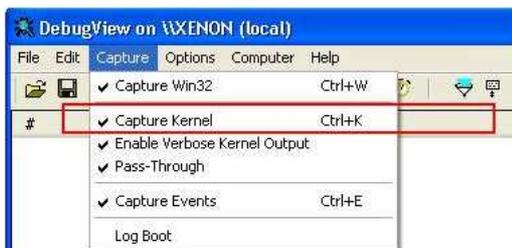
If the driver is failing to load or the load sequence is required then run up the UPDD Console and select reload from the status screen. When the Reload option is selected in UPDD it effectively reinitialises the driver and hardware port, outputting debug messages at each stage. These messages are displayed in the debug dialog and should be saved and sent to Touch-Base for analysis.

Example:

Windows

User reports that touch screen is taking excessive time to become active after system has been sleeping:

1) Set both debug levels to 5 (kernel and user) and reboot system. Check touch is active.
2) Invoke the kernel debugger (e.g. locate and run DebugView application)
3) Place system in sleep (or standby / hibernate) and return from sleep
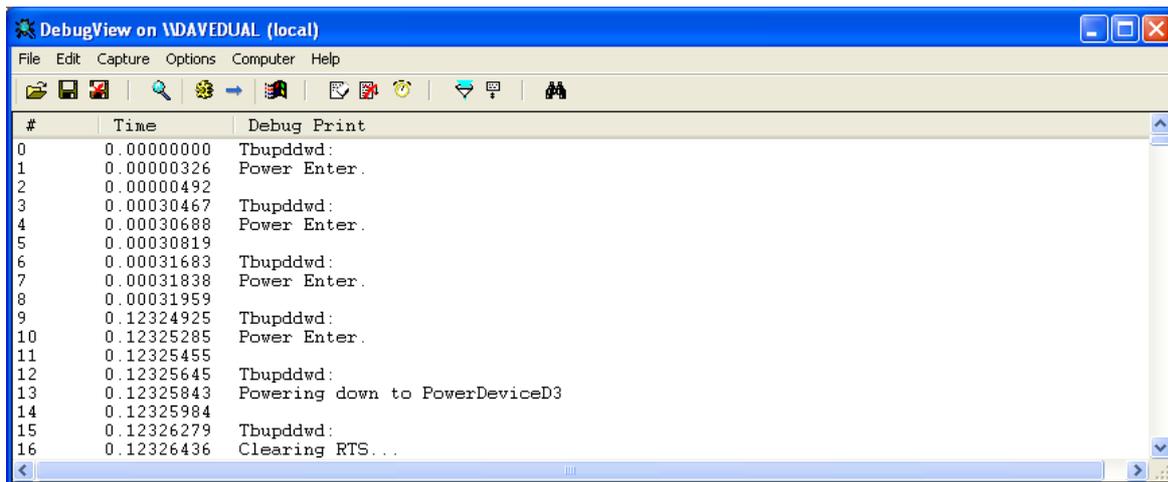
4) Use DebugView, File, Save As to save debug log.

Capturing debug over a reboot (Windows)

In some circumstances we may need to see the debug log over a system boot rather than when the driver is reloaded. In this instance the kernel debugger may have an option to log kernel debug output over a reboot as shown below in the DebugView debugger:

Selecting this option in DebugView acknowledges that data will be captured at next reboot:
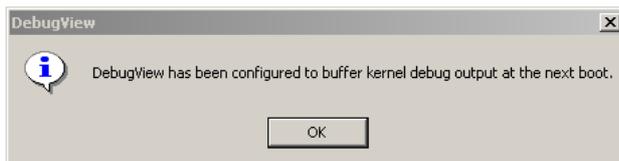
The debugger will capture all the debug messages produced by the driver during the reboot load.

Controller data capture

Occasionally we would like to view the raw data received from the controller. This can be achieved in one of two ways:

Tbcalib test mode

Tbcalib program has a test mode to log single point touch data to a file.

Tbcalib diag mode

Tbcalib program has a diagnostic mode to log trace data pattern to a file.

Remote Access

In some cases it is desirable to be able to remotely connect to a failing touch system to help identify the problem. We prefer to utilize a remote access product called Crossloop, available for free from http://www.crossloop.com/ipage.htm?id=predownload (the free, non professional version should be available). To make the system available for remote control, ensure it is physically connected to the internet, install and run the crossloop application (you do not need to register for an account) which will issue a unique session identifier which you need to pass on to the Touch-Base technician (phone or via on-line chat using Skype or Google Talk). Using this code the technician can connect to and take control of the system, as shown below:
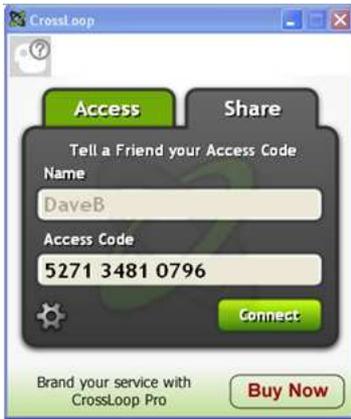
Run crossloop and select the Share option. Pass on the Access Code to the technician via phone or chat session.

When instructed by the technician hit connect.

Using the supplied access code, the technician will try and connect which will need to be approved. You can now watch whilst the system is controlled remotely and to be at hand to help if required.

Uploads

Any files requested to be sent to Touch-Base can be sent as email attachments or placed directly on our upload server at http://upload.touch-base.com – useful for very large attachments. Click on the link or enter this URL in any browser and you will see a drop zone in the browser to which you can drag and drop the file(s).

Contact

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