



Command line user interface

Revision 1.5 - 26th Aug 2011
www.touch-base.com/documentation \General

[TButils](#)

[Tbcilib](#)

[Contact](#)

TButils user interface

Starting with updd version 4.1.10 a new command line user interface utility is available called tbutils and replaces a number of functions previously located in tbcilib. For users of earlier UPDD versions see [tbcilib](#) user interface below.

The user interface program exports this interface using the following syntax:

```
Windows    C:\program file\updd\TButils {parameter}
           Note: Entering the commands from a Windows command line would be tbutils "{parameters}"
Mac OS X   /tbupddmx/tbutils.app/Contents/tbutils {parameter}
Linux      /opt/tbupddlx/tbutils {parameter}
           This command may need to be run prior to calling tbutils:
           export LD_LIBRARY_PATH=/opt/tbupddlx:$LD_LIBRARY_PATH'
```

Notes:

1. Output is directed to the console (stderr / stdout) allowing for scripted automation.
2. Success is indicated by rc=0, error is rc= -1.
3. When an error occurs a meaningful message is directed to the console (stderr).
4. Passed parameters are checked for correct syntax and values.
5. If running the utility outside the UPDD application folder you may need to add the UPDD path to the system library path, such that the utility can locate the ACE library, as per this Linux example,: export LD_LIBRARY_PATH=/opt/tbupddlx:\$LD_LIBRARY_PATH.

Parameters Description

{None} Lists the command syntax
list Lists UPDD device information. Lists for each device:
Internal UPDD handle / device name / desktop segment
This example shows a system with 3 touch devices >>>>>>>

A later version of tbutils also lists the device state as below;
This example shows a system with 2 devices in various states >>

NOK = device not connected
OK = device connected
!!! = Driver interface not available (tbupddwu has been stopped)

Version Shows the driver's release information, being;
Version number / build id / Production system id

[<device selector>] Selects the UPDD device against which to perform request. Only required in multi-device environment

Syntax Description

nodevice The "nodevice" option allows actions on [general \(non-device specific\) parameters](#)
segment Perform request on the updd device associated with the updd desktop segment identifier.
<name> E.g. Tbutils Segment 'Monitor 2' disable - would disable the updd device associated with Monitor 2.
device Perform request on the specified UPDD device. Handle is the device handle of the device as held by UPDD and shown in the list command above.
<handle> If no device selector is specified and one is required the first installed device is chosen

UPDD specific requests

These requests relate specifically to UPDD functions

disable	Disable the device
enable	Enable the device
pointeroff	Disable the driver mouse pointer interface (system wide - all devices)
pointeron	Enable the driver mouse pointer interface (system wide - all devices)
reinit	Reinitialise the controller and re-establish a link
reload	Force the driver to re-read settings (not necessary when using this interface to change a setting)
soundoff	Turn sound off for the device
soundon	Turn sound on for the device
togglesound	Toggle sound setting for the device
toolbaroff	Disable a named toolbar
<toolbarname>	
toolbaron	Enable a named toolbar
<toolbarname>	
unload	Instruct registered applications to terminate (TBApiSendUnloadMessage)
version	Returns the UPDD 3 part build number to stdout e.g.04:01:06R / 1221 / G11951. For backward compatibility this can be redirected to file version.txt if required.
dump4tba	This option is used to create default calibration data from a calibrated system. If no file path is defined the calibration data is written to file tbcilib.tba in the current folder or the file defined. Defining a path e.g. dump4tba c:\users\gary\tba.dat is useful if the current working folder is not writeable.
<file path>	
	The data is written in a format suitable for embedding in our software generation system such that the installation utilizes the default calibration data in the UPDD settings files. In this example a system has been calibrated with a 1 percent margin, 8 calibration points, 10 second timeout: Normal,1,8,10,0,0,15790,1223,15768,15642,885,1132,942,15756,12178,4929,12146,12180,4649,4959,4658,12040 The file containing the captured calibration data should be sent to Touch-Base for processing.
record	Records touch co-ordinate input to a file. Recording terminates after about 10 seconds of inactivity. Ensure that

```
c:\program files\updd>tbutils list
1      BonXeon           Monitor 1
2      Dell, ST2220T     Monitor 2
3      Quanta Computer, Dual Monitor 3
c:\program files\updd>_

C:\Program Files\UPDD>tbutils list
1 NOK Zytronic, ZXV100 Whole
2 OK Zytronic, ZXV100(2) Whole
C:\Program Files\UPDD>net stop tbupddwu
The tbupddwu service is stopping.
The tbupddwu service was stopped successfully.
C:\Program Files\UPDD>tbutils list
Unable to open connection to driver
1 !!! Zytronic, ZXV100 Whole
2 !!! Zytronic, ZXV100(2) Whole
c:\program files\updd>tbutils version
04.01.10R / 2257 / D16195
c:\program files\updd>_
```

<filename> file name is in a folder that has write access.

The file is a csv format:

x, y, stylus, z, reserved-for-future-use

e.g. 600,253,0,-1,0

Z is not recorded, this has to be manually added. (-1 is "no value"). Only works if the Z (pressure) axis is both supported by the device and in UPDD's configuration for the device.

playback Used to playback previously captured touch co-ordinate data using the 'record' function above. Events are played back one every 20ms or so, so the playback might run at a slightly different speed to the recording.

<filename>

Periods of no input are identified by null records thus:

-1,-1,-1,-1,0

Use the following options to change arbitrary UPDD settings file entries. UPDD settings are documented in the [UPDD settings](#) file. By default the setting changes are applied to Device 1 - this equates to the settings file branch [[UPDD/Parameters/1](#)]. To change settings in the general branch (not related to a device) use 'desktop selector' = nodevice - this equates to the settings file branch [[UPDD/Parameters](#)].

Note: Take care when updating the driver settings as setting an invalid setting could result in unpredictable behaviour or a crashed driver! You have been warned.

setting <dw> Set the DWORD setting *name* to the hex numeric *value*

<name>

<value>

<sz> Set the String setting *name* to *value*

<name>

<value>

Note: If the setting *name* has a space then it must be quoted, e.g. tbutils nodevice dw "calibration beeps" 0

Layout

Shows system's monitor layout as seen by UPDD software. UPDD 4.1.10 and above utilises QT monitor matrix ([QDesktopWidget](#) class) in all OS to retrieve monitor layout information and for monitor handling. For successful UPDD multi-monitor operation the QT monitor properties must see either different monitor ids or co-ordinates. This option lists the monitor configuration as seen by UPDD, as per the examples below:

Windows system with 6 monitors

```
Administrator: Visual Studio 2005 CMD
C:\Program Files\UPDD>tbutils layout
Qt reports the following monitor layout
information
found 6 monitors
primary monitor is 0
Monitor 0:
  Left: 0
  Top: 0
  Right: 1919
  Bottom: 1079
Monitor 1:
  Left: 0
  Top: -1080
  Right: 1919
  Bottom: -1
Monitor 2:
  Left: 3840
  Top: -1080
  Right: 5759
  Bottom: -1
Monitor 3:
  Left: 3840
  Top: 0
  Right: 5759
  Bottom: 1079
Monitor 4:
  Left: 1920
  Top: 0
  Right: 3839
  Bottom: 1079
Monitor 5:
  Left: 1920
  Top: -1080
  Right: 3839
  Bottom: -1
C:\Program Files\UPDD>
```

Linux system with 2 monitors

```
leaton@localhost:~
File Edit View Terminal Tabs Help
[leaton@localhost ~]$ ls 145
/opt/tbupddlx/tbutils layout
Qt reports the following monitor layout information
found 2 monitors
primary monitor is 0
Monitor 0:
  Left: 0
  Top: 0
  Right: 1359
  Bottom: 767
Monitor 1:
  Left: 0
  Top: 0
  Right: 1023
  Bottom: 767
[leaton@localhost ~]$
```

Controller specific requests

Only shown if the software is configured to support the controller.

Zytronic - The following four options are for use with the Zytronic X-Y controllers and **will not work** with the new (Apr 2010) ZY100 controller. These functions are useful in OSes (Windows CE etc) where the UPDD Console, firmware dialog is not available to make the settings.

zysensitivity=nn Set touch sensitivity in Zytronic X-Y controllers. Range 0 to 50.

zyavframes=n Set number of frames for X / Y averaging in Zytronic X-Y controllers. Range 0 to 9.

zyglasstype=n Set the glass thickness in Zytronic X-Y controllers The controller can be adjusted using this setting to operate through various overlay thicknesses. Available options are Thin, Medium and Thick. The Medium setting is the default. These settings operate on time averaging of captured data from the sensor, hence the thicker the overlay, the sensor response time is reduced due to the greater time interval of data captured. These options should be used in conjunction with the Threshold (Sensitivity) setting adjustment to obtain optimum operation when using various thicknesses of overlays. Range 0 (Thin), 1 (Medium) and 2 (Thick).

zynormalisation Initiates a normalisation of the sensor array wire levels in Zytronic X-Y controllers.

zysensitivity=nn Set touch sensitivity in Zytronic X-Y controllers. Range 0 to 50.

TRS - The following four requests are for use with the TRS Star controllers.

trsset Further information available from TRS

touch_threshold

<val>

trsget Further information available from TRS

trsrecalibrate Further information available from TRS

trsrestart Further information available from TRS

ELO - The following two requests are for use with ELO Smartset controllers.

smtwrite:nnnnnn Set the serial number to nnnnnn on the ELO Smartset controller.

Note: Changing the serial number causes UPDD to see a new device, so an additional device will be listed in the UPDD Console device list when the controller reports its serial number (this appears to be after rescan of devices, such as a replug or a reboot).

Equivalent API call

BOOL TBAPI TBApiWriteSmartsetUSBSerialNumber(HTBDEVICE aDevice, const TCHAR* aBuffer, DWORD aSize);

smtread Read the serial number from the ELO Smartset controller and send to stdout. (Only use one ELO Smartset controllers, otherwise the behavior is undefined). For backward compatibility this can be redirected to file smtread.txt if required.

Equivalent API call

BOOL TBAPI TBApiReadSmartsetUSBSerialNumber(HTBDEVICE aDevice, TCHAR* aBuffer, DWORD aSize);

Hampshire - The following two requests are for use with Hampshire/Microchip tsharc controllers.

tsharcwrite:n Write the serial number n to the Hampshire TSHARC controller. (Only use one controller, otherwise the behavior is undefined).

Equivalent API call

BOOL TBAPI TBApiWriteTSHARCUSBSerialNumber(HTBDEVICE aDevice, const TCHAR* aBuffer, DWORD aSize);

tsharcread Read the serial number from eeprom on the Hampshire TSHARC controller and send to stdout (Only use one controller, otherwise the behavior is undefined). For backward compatibility this can be redirected to file tsharcread.txt if required.

Equivalent API call

BOOL TBAPI TBApiReadTSHARCUSBSerialNumber(HTBDEVICE aDevice, TCHAR* aBuffer, DWORD aSize);

Tbcalib user interface

In UPDD versions up to and including 4.1.8 user interface calls were held in Tbcilib. With 4.1.10 and above they were relocated to a new utility program, tbutils, as documented above.

The calibration program exports this interface using the following syntax:

```
Windows    Tbcilib {parameter}
           Note: Entering the commands from a Windows command line would be tbcilib "{parameters}"
Mac OS X   /tbupddmx/tbcilib.app/Contents/MacOS/tbcilib {parameter}
Linux      /opt/tbupddlx/upddcalib {parameter}
           or
           /opt/tbupddlx/tbcilib {parameter}
           This command may need to be run prior to calling tbcilib:
           export LD_LIBRARY_PATH=/opt/tbupddlx:$LD_LIBRARY_PATH'
```

Notes:

- 1) If the parameter affected has a space then the parameter value must be quoted, e.g. tbcilib Device=0 "/setting:calibration beeps=0". **In some cases we have seen " ignored and ' have worked! Please try ' if " cause issues.**
- 2) **Win7 file write issue:** Some of the user interface calls create files in the UPDD application folder and under Windows 7 this folder may not have correct write permissions to allow for files to be created. In this case you may see an error or you may not find the file (it will be remapped elsewhere). When using functions that create files ensure you have administration rights.

Please note parameters are case sensitive and must be defined as shown below.

Calibration parameters	Used with the calibration procedure
<i>None passed</i>	will calibrate the first active device on the system.
Device=n	perform request on the specified updd device and, if calibrating, the currently selected calibration style, default first in list. Will also calibrate any defined toolbars unless 'Toolbar=ABogusValue' is used to disable toolbar processing. Normally used by calling programs to perform a given function against a specific device, such as the UPDD Console device calibration option. N=the device handle of the device as held by UPDD. This option is used by UPDD SDK based programs tilizing the UPDD API to determine the device handle using related API calls such as TBAPIGetRelativeDevice.
Device=connected	Perform request on the first connected device.
"Segment=segment id" (UPDD ver 4.1.3 and above)	Perform request on the updd device associated with the updd desktop segment identifier and, if calibrating, the currently selected calibration style, default first in list. Will also calibrate any defined toolbars unless 'Toolbar=ABogusValue' is used to disable toolbar processing Normally used by calling programs to perform a given function against a specific device, such as the UPDD Console device calibration option. e.g. Tbcilib "Segment=Monitor 2" /disable - would disable the updd device associated with Monitor 2.
User Interface Calls	When Tbcilib is invoked with a user interface parameter only the function associated with the parameter is performed. As would be expected, calibration is not invoked.
Device=n	See above definition.
Segment=" segment id"	See above definition.
/reinit	Reinitialise the controller and re-establish a link
Equivalent API call	TBApiReinit(passedDeviceNumber);
/reload	Force the driver to re-read settings (not necessary when using this interface to change a setting)
Equivalent API call	TBApiReloadNoApply();
/toolbaroff:toolbarname	Disable a named toolbar
Equivalent API call	TBApiEnableToolbar
/toolbaron:toolbarname	Enable a named toolbar
Equivalent API call	TBApiEnableToolbar
/toggltouch	Toggle the device enabled state
Equivalent API call	DWORD dw; TBApiGetSettingDWORD(passedDeviceNumber,_T("Enabled"),&dw); dw ^= 1; TBApiSetSettingDWORD(passedDeviceNumber,_T("Enabled"),dw);
/enable	Enable the device
Equivalent API call	TBApiSetSettingDWORD(passedDeviceNumber,_T("Enabled"),1);
/disable	Disable the device
Equivalent API call	TBApiSetSettingDWORD(passedDeviceNumber,_T("Enabled"),0);
/pointeroff	Disable the driver mouse pointer interface (system wide - all devices)
Equivalent API call	TBApiMousePortInterfaceEnable(false);
/pointeron	Enable the driver mouse pointer interface (system wide - all devices)
Equivalent API call	TBApiMousePortInterfaceEnable(true);

<code>/soundon</code>	Turn sound on for the device
Equivalent API call	<code>TBApiSetSettingDWORD(passedDeviceNumber,_T("Sound"),1);</code>
<code>/soundoff</code>	Turn sound on for the device
Equivalent API call	<code>TBApiSetSettingDWORD(passedDeviceNumber,_T("Sound"),0);</code>
<code>/togglesound</code>	Toggle sound setting for the device
Equivalent API call	<code>DWORD dw; TBApiGetSettingDWORD(passedDeviceNumber,_T("Sound",&dw); dw ^=1; TBApiSetSettingDWORD(passedDeviceNumber,_T("Sound"),dw); TBApiApply();</code>
<code>/screenresupdate</code>	MAC OS X only – Requests the driver to recalculate calibration mapping based one the current screen resolution. To be used where a system is calibrated in one resolution but uses other resolutions (especially useful where applications are changing resolution)
Equivalent API call	<code>DWORD nDevices; TBApiGetSettingDWORD(0, _T("Number Of Devices"), &nDevices); for(unsigned j = 0; j < nDevices; ++j) { int dev=0; dev = TBApiGetRelativeDevice(j); if(!dev) { continue; } else { SetupForMultiMonitor(dev,this); } }</code>
Use the following options to change arbitrary UPDD settings file entry. UPDD settings are documented in the UPDD settings file. By default the setting changes are applied to Device 1 – this equates to the settings file branch [UPDD/Parameters/1]. To change settings in the general branch (not related to a device) use Device=0 – this equates to the settings file branch [UPDD/Parameters].	
<code>/setting:XXX=NNN</code>	Set the DWORD value XXX to the hex numeric value NNN
<code>/settingsz:XXX=ZZZ</code>	Set the string value XXX to the value ZZZ
Note: If the setting name has a space then the option must be quoted, e.g. <code>tbcalib Device=0 "/setting:calibration beeps=0"</code> . See note 1 above for important information!	
<i>The following four parameters are for use with the Zytronic X-Y controllers and will not work with the new (Apr 2010) ZY100 controller. These functions are useful in OSes (Windows CE etc) where the UPDD Console, firmware dialog is not available to make the settings</i>	
<code>zysensitivity=nn</code>	Set touch sensitivity in Zytronic X-Y controllers. Range 0 to 50.
<code>zyavframes=n</code>	Set number of frames for X / Y averaging in Zytronic X-Y controllers. Range 0 to 9.
<code>zyglasstype=n</code>	Set the glass thickness in Zytronic X-Y controllers The controller can be adjusted using this setting to operate through various overlay thicknesses. Available options are Thin, Medium and Thick. The Medium setting is the default. These settings operate on time averaging of captured data from the sensor, hence the thicker the overlay, the sensor response time is reduced due to the greater time interval of data captured. These options should be used in conjunction with the Threshold (Sensitivity) setting adjustment to obtain optimum operation when using various thicknesses of overlays. Range 0 (Thin), 1 (Medium) and 2 (Thick).
<code>zynormalisation</code>	Initiates a normalisation of the sensor array wire levels in Zytronic X-Y controllers.
<code>/smtwrite:nnnnnn</code>	Set the serial number to nnnnnn on the ELO Smartset controller. Note: Changing the serial number causes UPDD to see a new device, so an additional device will be listed in the UPDD Console device list when the controller reports its serial number (this appears to be after rescan of devices, such as a replug or a reboot).
Equivalent API call	<code>BOOL TBAPI TBApiWriteSmartsetUSBSerialNumber(HTBDEVICE aDevice, const TCHAR* aBuffer, DWORD aSize);</code>
<code>/smtread</code> (Win 7? – See note)	Read the serial number from the ELO Smartset controller and dump to the file smtread.txt (Only use one ELO Smartset controllers, otherwise the behavior is undefined).
Equivalent API call	<code>BOOL TBAPI TBApiReadSmartsetUSBSerialNumber(HTBDEVICE aDevice, TCHAR* aBuffer, DWORD aSize);</code>
<code>/tsharcwrite:n</code>	Write the serial number n to the Hampshire TSHARC controller. (Only use one controller, otherwise the behavior is undefined).
Equivalent API call	<code>BOOL TBAPI TBApiWriteTSHARCUSBSerialNumber(HTBDEVICE aDevice, const TCHAR* aBuffer, DWORD aSize);</code>
<code>/tsharcread</code> (Win 7? – See note)	Read the serial number from eeprom on the Hampshire TSHARC controller and dump to the file tsharcread.txt (Only use one controller, otherwise the behavior is undefined).
Equivalent API call	<code>BOOL TBAPI TBApiReadTSHARCUSBSerialNumber(HTBDEVICE aDevice, TCHAR* aBuffer, DWORD aSize);</code>
<code>/version</code> (Win 7? – See note)	Available in 4.1.6, (build 1221 and above), returns the UPDD version number in a text file called version.txt as a 3 part build number: e.g.04:01:06R / 1221 / G11951

Tbcalib return codes

These are the return codes from TBcalib and access to the code will be specific to the launch method used:

0	Success
4	Syntax error passing parameter
5	Failure to open API
6	Couldn't find a desktop segment
7	ZY Value passed for sensitivity setting out of range
8	ZY Set sensitivity failed
9	ZY Value passed for glasstype setting out of range
10	ZY Set glasstype failed
11	ZY Value passed for average frames setting out of range
12	ZY Set averaged frames failed
13	ZY Set normalisation failed

Contact

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