This document, API Features, for want of a better title, merely holds useful information 'loosely' related to API utilisation that has no better home to be listed other than features described here are likely to be cross referenced from other documents:

Packet Disposition

The driver can be in two modes of operation, one where it is processing the data being sent from the touch device and one where it is merely being used as a mechanism to pass all received data to a calling program. This is referred to as raw mode of operation.

If an application is wanting to receive data in this raw mode then starting with version 5.1.0 it is recommend that client programs should be aware of the "packetdisposition" setting, this offers a more flexible and in some cases more efficient method to deal with raw data.

New feature

"packet disposition" setting

Starting with UPDD version 5.1.0 the "packet disposition" setting is a controller setting that provides additional control over how the driver will deal with "raw data". Raw data is one of the following:

1) data defined in a UPDD packet definition that has only sync or NOP definitions.

2) Any data received from a device when the "raw data mode" is set with the TBApiRawDataMode api.

3) data from a source identified by this "packet disposition" setting.

This setting is a standard updd setting and can be set

1) as a default controller setting by Touch-base

2) using tbutils

3) Using the UPDD API.

Example

using tbutils

```
tbutils setting sz "packet disposition" "1,0,81,t;1,1,82,d"
```

using the api

```
TBApiSettingSZ(device,"packet disposition", "1,0,81,t;1,1,82,d");
TBAPIApply();
```

The packet disposition string is a comma separated list of 4 items.

Multiple dispositions can be joined with a semi-colon separator.

The meaning of the tokens is shown below, with zero being the first.

0: Placeholder for USB configuration (the C in C:I:E) and currently only 1 is supported

1: a USB interface index. For non USB devices this value is ignored.

2: a USB endpoint address. The address of a host to client endpoint. For non USB devices this value is ignored.

3: The actual disposition as follows

'n': the data is discarded

's': deliver all received information to serial device associated with UPDD handle 1. See note 1 below.

'd': deliver all received information to api, for client programs to process as raw data (_ReadDataTypeData).

't': process incoming items (part from raw packets as defined above) as touch data.

'b': implements both options 't' and 'd' above.

Examples

Send all data from a serial device to client callbacks registered with _ReadDataTypeData

(equivalent to TBAPIRawDataMode(true));

```
TBApiSettingSZ(device,"packet disposition", "0,0,0,d");
```

Process all non raw packet data from a serial device as touch data (standard default behaviour and (equivalent to TBAPIRawDataMode(false));

(equivalent to TBAPIRawDataMode(true));

```
TBApiSettingSZ(device,"packet disposition", "0,0,0,t");
```

Send all data from a simple USB device to client callbacks registered with _ReadDataTypeData

(equivalent to TBAPIRawDataMode(true));

This assumes that the USB device uses configuration 1, interface 0, endpoint 81, which is most often the

case.

TBApiSettingSZ(device,"packet disposition", "1,0,81,d");

Process data from different interfaces / endpoints differently.

TBApiSettingSZ(device,"packet disposition", "1,0,81,t;1,1,82,d");

This example treats data from configuration 1, interface 0, endpoint 81 as touch data and data from configuration 1, interface 1, endpoint 82 is sent to client callbacks registered with _ReadDataTypeData.

Using TBApiRawDataModeBlockSize with "packet disposition".

If TBApiRawDataModeBlockSize is not called or is called with a block size value of zero, data delivered to API callbacks for USB devices will be delivered in blocks of the same size of data received from the USB device. This allows for more efficient transfers even when mixed packet sizes are used. This overcomes the limitations of the TBAPIRawMode api which mean that a block size of 1 is normally required.

Similarly for non-USB devices if TBApiRawDataModeBlockSize is not called or is called with a block size value of zero the data is sent when the driver identifies a complete data packet. NB this does means that if raw data is used that does not match a packet definition then for the most reliable delivery a block size of one must still be set.

Notes:

1. Implemented for a project whereby a multi-touch usb device was required to be used on an OS that could not support USB devices. The device was connected to a Windows system whereby UPDD was configured for 2 devices, one serial and one USB. The usb device received the touches and they were processed and posted to the serial device. The other system (in this case a Solaris 8 system) utilised UPDD Solaris driver configured for a serial device. A serial cross over cable was used between the 2 systems.

   In theory a PCI Single board computer could have been plugged into the Solaris 8 PCI slot so that the windows system was contained within the Solaris 8 system.

Contact

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