

HID Feature Request API

Two API call's have been specifically created to cater for HID Get and Set feature requests.

TBApiHidGetFeature

This API is used to issue an HID get feature request to a UPDD controlled device.

It is intended for use by programmers familiar with and as a replacement for a direct HID API call in cases where such a call is not possible due to UPDD being in control of the device.

As such a detailed description of the HID report format is not given here.

The document [here](#) describes this in the context of the native Windows API.

Definition

```
TBBOOLTBAPI TBApiHidGetFeature(  
    HTBDEVICE aDevice,  
    int aInterface,  
    void* aReportBuffer,  
    uint32_t aReportBufferLength  
);
```

Parameters

Device: the UPDD device handle representing the device instance to be queried.

aInterface: the interface number to be queried

aReportBuffer:

A block of memory allocated by the caller

When calling the API the caller sets up the HID feature report data to be sent to the device.

Typically this is simply the report id of the feature in byte zero.

HID Feature Request API

`aReportBufferLength`: The length of the allocated block. This should be long enough to contain the expected response from the controller including the leading report id byte.

Notes: in normal HID mode a programmer will use an open handle to an HID collection.

Using the UPDD API; use standard UPDD APIs to derive the device handle and consult with the hardware supplier and / or examine USB traffic in HID mode to derive the interface number. The interface number is defined in the USB report descriptors and as such is constant for a given device in most cases. In many cases this value is simply zero.

TBApiHidSetFeature

This API is used to issue an HID get feature request to a UPDD controlled device.

It is intended for use by programmers familiar with and as a replacement for a direct HID API call in cases where such a call is not possible due to UPDD being in control of the device.

As such a detailed description of the HID report format is not given here.

The document [here](#) describes this in the context of the native Windows API.

Definition

```
TBBOOLTBAPI TBApiHidSetFeature(  
  
    HTBDEVICEaHandle,  
    intaInterface,  
    constvoid* aReportBuffer,  
    uint32_t aReportBufferLength  
);
```

Parameters

HID Feature Request API

aDevice: the UPDD device handle representing the device instance to be queried.

aInterface: the interface number to be queried

aReportBuffer:

A block of memory allocated by the caller

When calling the API the caller sets up the HID feature report data to be sent to the device.

Typically this is simply the report id of the feature in byte zero followed by the feature data.

aReportBufferLength: The length of the allocated block.

Notes: in normal HID mode a programmer will use an open handle to an HID collection.

Using the UPDD API; use standard UPDD APIs to derive the device handle and consult with the hardware supplier and / or examine USB traffic in HID mode to derive the interface number. The interface number is defined in the USB report descriptors and as such is constant for a given device in most cases. In many cases this value is simply zero.

Test interface for these APIs.

The UPDD [Command Line Interface](#) has two options which can be used to test the use of these APIs with a device under test

These are intended for use by programmers familiar with making direct API calls and as such a detailed description of formulating an HID feature get / set request is not be given here.

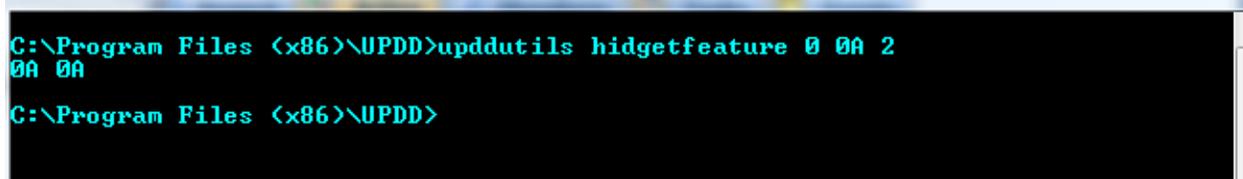
HidGetFeature

HID Feature Request API

Syntax

```
upddutils hidgetfeature <interface> <report_id> <length>
```

Example



```
C:\Program Files <x86>\UPDD>upddutils hidgetfeature 0 0A 2  
0A 0A  
C:\Program Files <x86>\UPDD>
```

In this example get feature report 0A is send to the device.

For this device feature 0x0a is the max count HID feature.

The report ID must be entered as a 2 character ASCII representation of the hex value.

The report length is 2 bytes.

The response indicates that feature 0x0a has a value of 0x0a.

HidSetFeature

Syntax

```
upddutils hidsetfeature <interface> <reportdata>
```

Example

HID Feature Request API

In this example set feature report 05 is sent to the device with report data of 0x02,00

For this device feature 0x05 is the device mode feature

The data must be entered as a series 2 character ASCII representations of the hex values, with no spaces.

Touch-Base Support

<http://support.touch-base.com/Documentation/50409/HID-Feature-Request-API>