

USB stall issues

USB connected devices can report a stall condition to the host. Typically this means that an error has occurred. The nature of and the recovery from these errors varies by device.

Some devices never report stalls; some report stalls but can be ignored; some require a re-enumeration (re-initialisation) after a stall.

When [driver debugging is enabled](#) or [UPDD diagnostics](#) is invoked a debug log will record USB activity. If a stall occurs it will likely be reported in the log file.

Stall issues during initialisation

We have found some USB controllers stall when the driver issues USB requests to the controller as discussed below.

This hardware stall can sometimes result in a lockup in the USB interface module we utilise in the driver to handle the USB interface.

The USB initialisation requests that UPDD issues are as follows:

Request	Description
Set alternate interface	This is a feature we have never seen used for a touchscreen, however we have seen devices where not explicitly setting alternate 0 caused a failure. For this reason the driver always explicitly sets alternate 0 for all utilised interfaces unless this option is disabled as described below.
Get max contacts	This is issued to determine the number of supported contacts if the HID Report Descriptor indicates a multi-touch device.
Set device mode	This is issued to set multi-touch mode if the HID Report Descriptor indicates a multi-touch device .

We have seen the Set Device Mode stall a TPK, Fusion 4 and one other 'unknown' device.

USB stall issues

When this happens the symptoms seen are:

1. 'No driver connection' is shown in UPDD Status.

If the Set Alternative USB request is the cause the debug log will show

2019/07/08-14:48:00: ERR: USB error: (setting alt=0) Other error

2019/07/08-14:48:00: ERR: Skipping this interface

2. The [bootstrap log](#) has 'Issuing set device mode' as last entry.

It is often the case that the device defaults to multi-touch mode so does not need or is not expecting the 'set device mode' request.

Disabling USB requests

If it is discovered that these requests are having a negative affect on the device you can set a corresponding disable setting using the [command line interface](#) 'setall' command. The settings are **disable_set_alt0**, **disable_get_max_contacts** and **disable_set_device_mode** respectively and should be set to 1 to disable the request, i.e.

```
Upddutils setall disable_set_alt0 1
```

```
Upddutils setall disable_set_device_mode 1
```

```
Upddutils setall disable_get_max_contacts 1
```

It is hoped that in some future release of the USB interface we will be able to detect the stall, retry the failing command and then continue.

Stall issues during device access

These stalls can result in UPDD disconnect [notifications](#) being issued (if notifications are enabled) and/or occasional loss of touch is experienced; this might be due to this type of device stall.

[Enable debug logging](#) and observe the log for the time of the error or [run diagnostics](#) (which also creates a debug log).

USB stall issues

A stall is identified with a message like that below.

```
2019/09/27-20:52:22: ***: USB: (LIBUSB_TRANSFER_ERROR)
```

```
2019/09/27-20:52:22: ***: USB: (LIBUSB_TRANSFER_ERROR closing device)
```

or

```
2019/09/27-20:52:29: ***: USB: (LIBUSB_TRANSFER_STALL)
```

```
2019/09/27-20:52:29: ***: USB: (LIBUSB_TRANSFER_STALL closing device)
```

Stall handling

The drivers stall behaviour is defined by the setting:

```
usb.handle_stall
```

This is a bootstrap setting and as such must be set with the upddutils [setall command](#) and the driver must be restarted after setting this value as in this Windows example:

```
C:\Program Files (x86)\UPDD>upddutils setall usb.handle_stall 1
C:\Program Files (x86)\UPDD>net stop updd
The UPDD Driver service was stopped successfully.
C:\Program Files (x86)\UPDD>net start updd
The UPDD Driver service is starting.
The UPDD Driver service was started successfully.
C:\Program Files (x86)\UPDD>
```

The valid values for this setting are

0 - the device is closed and the re-enumeration process invoked immediately

1 - the driver will issue a USB "clear halt" and retry.

USB stall issues

From build 516 onwards the following values are supported:

- 2 - the error is simply ignored, the next read is issued as though no error occurred
- 3 - the device is closed and the re-enumeration process invoked immediately, no disconnect / connect messages are issued

The best value will need to be determined by experiment.

Notes

1. If no value is specified for this setting a value of 0 is assumed.
2. This is the default. The behaviour for a value of 1 (clear halt and retry) is the theoretical correct action but we see a number of devices that do not implement this correctly so it is not used as the general default.
3. Some devices / system will react badly to certain settings and in the worst case it might be necessary to reinstall the software if a lockup occurs as a result of changing this setting.

Touch-Base Support

<http://support.touch-base.com/Documentation/50454/USB-stall-issues>