Verifone®

Windows Unified Driver
User Guide

Revision 0.1.1

TABLE OF CONTENTS

DOCUMENT REVISION HISTORY

DATE	REVISION	DESCRIPTION
2014-07-04	0.0.1	Initial revision.
2015-05-07	0.0.2	Add RNDIS, batch file, releases details
2015-06-02	0.0.3	Update for v 5.0.0.6
2016-02-23	0.0.4	Update for v 5.0.0.8
2016-05-17	0.0.5	Updated inf file
2016-08-31	0.0.6	Added new fix in 5.0.0.8
2016-09-29	0.0.7	Update for v 5.0.1.0
2016-10-25	0.0.8	Update for v 5.0.1.1
2016-12-16	0.0.9	Update for v 5.0.2.1
2017-01-20	0.0.10	Update for v 5.0.3.0
2017-02-17	0.1.0	Added supported OS and devices
2017-04-05	0.1.1	Update for v 5.0.4.0

1. INTRODUCTION

1.1. Description

This document describes how to use the Windows drivers, installer, associated tools, and details the version changes.

It also describes the logging that is available, and some common issues found with USB installations.

1.2. Abbreviations

USB	Universal Serial Bus
NDIS	Network Device Interface Specification
RNDIS	Remote NDIS

1.3. References

No.	VDN	Title	Revision

2.   OVERVIEW

The Unified driver consists of a Serial driver and an RNDIS driver. The Serial driver consists of an inf file, and sys files, provided by Verifone. The RNDIS driver consists of an inf file, provided by Verifone. The sys files are provided by Microsoft and are already on the system. On XP systems however there is a bug in them so a hotfix, detailed later in the RNDIS install section, must be applied.

This document is divided into sections describing the design rational, installation, and trouble-shooting procedures. It also describes the ancillary tools and their usage as well as the logging available in the driver. It also describes the USB enumeration process, what the stages are, what the common faults are, and what they mean when they occur.

Included in this document are the changes at each version of driver release, and the GUID values used in each version of the msi that might be used in an installation batch file.

3. DRIVER SUPPORTS

Windows platforms :

Win 10	32/64 bit
Win 8.1	32/64 bit
Win 8	32/64 bit
Win 7	32/64 bit
Win Vista	32/64 bit
Win XP	32/64 bit

Terminal Series :

Vx (both Verix V and eVo).

VX 510 Ethernet Terminal

VX 510 GPRS Terminal

VX 610 Terminal

VX 570 Terminal

VX 670 Terminal

VX 700 Terminal

VX 800 PIN Pad

VX 810 PIN Pad

VX 680 Terminal

VX 520 Terminal

VX 820 PIN pad

VX 805 Terminal

VX 825 Terminal

VX 520 GPRS Terminal

VX 680 CDMA Terminal

FD60 Terminal

V5/VMT USB UART Device

VX 680 BlueTooth Terminal

VX 680 WiFi Terminal

VX 520 Sprocket Printer Terminal

VX 675 Terminal

VX 680 3G Terminal

VX GEN3 Terminal

VX 675 3G Terminal

VX 820 DUET base Terminal

VX 805 DUET base Terminal

VX 690 3G Terminal

VX 685 Terminal

VSP100 Terminal

VX 600 payware mobile reader

VX 690 BASE

PP1000 Next Gen(V3)

C520H Terminal

C680 Terminal

Mx and UX

Mx8 Terminal

Mx9 Terminal

Ux100 Terminal

Ux200 Terminal

Ux300 Terminal

Engage Terminals

4. VERSIONS

5.0.0.0

Base release the latter data build has additional tracing

5.0.0.1

Has 2 second timeout on sent URBs. This avoids app hangs if the pinpad isnt responding

5.0.0.2

1. 2 second timeout on URBs removed, uses timeout from   write comm timeout now.

2. write timeouts fixed

3. g_OutputDispatchEntry   change to g_OutputDispatchEntryExit and controls both

4. URB Irp complete request no longer logged (doesnt need logging)

5. SYNCHRONIZE added to ZwCreateFile attributes

6. Set cancel routine of write irp returned at Disconnect to NULL.

7. Lock logging to help log entry synchronisation

8. Free log context to lookaside list after being written to file.

9. SingleDevice logic is now moved entirely to coinstaller, plugging the device in any com port results in the it being given the same com number

10. va functionality protected from dispatch level use

11. g_ndevice not being decremented

5.0.0.3

1. Fix BSOD when FileLoggingOff is set to 1

2. 5 second default write timeout used if none is set

3. KRecycleCompletionRoutine now traces URB IRP Completion.

4. g_ndevice decremt removed, the name of the device this numeric value is used in is irrelevant

5. Removed ExRaiseStatus() as they cause the KLock destructor to be skipped, and raise the level to DISPATCH causing BSODs

6. DriverUnload waits for all logs to be written from queue

5.0.0.4

1. Fix log overwrite bug

2. Add PID_aaaa   (multifunction serial) device id

3. Fix Read Interval timer bug

4. Improve logging clarity

5. Fix large memory utilization bug

6. User mode components now statically linked to avoid missing VC80CRT error

7. Windows 8 and 8.1 signed

5.0.0.5

1. Fix read bug resulting in occasional received packet loss/wrong data (buffer) being sent up to app.

2. Installs RNDIS inf and drivers.

3. For win7 64 bit you will need this OS patch, because MSFT changed the driver signing hash recently https://technet.microsoft.com/en-us/library/security/3033929

4. Co-installer adds entry to HKLM\Software\Microsoft\Windows NT\CurrentVersion\Ports, 9600, n, 8, 1   as default port data (on some systems this has been absent)

5. RNDIS sys files are self-signed for XP.

5.0.0.6

1. Fix read bug when the device is opened, closed, and opened in quick succession. Read requests were not being returned.

2. Remove 5 second default read timeout. Applications must set this themselves.

3. Remove installing RNDIS sys files on XP systems, POSReady 2009 should have them already.

5.0.0.8

1. Add Exar ID to inf file and driver

2. Add Raptor ID to inf file

3. Add SetLineCoding for Exar

4. Fix uninstall-install bug for XP and win7 64 where RNDIS component wasn't getting installed.

5. Updated inf file with supported Verifone devices.

6. Fix provided for installing the driver in Windows 10 OS.

5.0.0.10

1. Delay has been provide in open/close calls.

5.0.1.0

1. Added generic read/write permissions in .inf file.

2. Enhanced the setline coding to avoid XACT error.

5.0.1.1

1. Refined error recovery mechanism to overcome enumeration issue on multiple download/ terminal restart.

5.0.2.1

1. Added support for PP1000V3.

2. Fix for BSOD with parameter –

   DRIVER_UNLOADED_WITHOUT_CANCELLING_PENDING_OPERATIONS.

3. Enhanced support by providing custom port retaining feature.

4. Fix for windows 10 port hopping issue.

5.0.3.0

1. Fix for 100% disk usage issue due to excessive logging.

2. Driver logging to file is disabled by default. This can be enabled using registry values as described in LOGGING section.

5.0.4.0

1.   Vx690 Device Friendly Name Change From 'Vx690 BASE' to 'VX 690 BASE'.

2.   Default logging file size changed from 30KB to 30 MB.

5. DESIGN RATIONALE

5.1. Serial driver

The Unified Serial Driver is designed to make the allocation of communication port names much easier, and with its associated software, enable easier management of them.

The Unified Driver also works on all Verifone devices, VX and MX, and manages them without distinction.

The design rationale behind port management enhancement is to principally allow the use of custom port names, so rather than having to use the Windows dictated COMn format, any three letter text combination can be used such as ABCn for example.   This allows an installation to avoid clashing with existing COMn ports on the machine. A custom name like this is specified in the same way a COMn name in the call to CreateFile(), used by an application to open the port.

The management of the associated number, n, is also automated so that subsequent installations of devices on a single machine cause n to be incremented. So for example if XYZ and 10 were chosen during install, the first new device connected to the machine would be given the port name XYZ10 and a second new device XYZ11.

These port names are, as Windows dictates, tied to physical ports on the machine, so if both devices were disconnected, and the second device inserted again to the same port, it will be named XYZ11 again.

This provides a dependable and logical naming of ports which applications can exploit to direct data to specific devices.

The machine can also be specified as a Single Device System in which case installation of a second new device will be blocked. This allows tighter control of the operating environment by limiting cash register to using one specific com port. If the device is moved to another port it will be assigned the same port name and number.

Specific device can also be deleted from a system, whether the device is present or not, with the VFI Device Manager tool.

This tool also permits changing the three letter device name. This change does not affect devices already installed, only new installations after the change. For example, if the custom name is MXX, an MX9 device can be installed, resulting in a port name of MXX9, and

then the custom name changed via VFI Device Manager to VXX, an installation of a VX device would be assigned a port name of VXX10.

This allows applications to manage specific types of devices based on name which can permit a looser understanding in the application of the ports on the machine since it allows the application to search for a port by name regardless of it numerical extension.

5.2. RNDIS driver

The RNDIS component stems from the ability of the USB driver on the Linux system on the MX to be configured as an RNDIS device, and thus avoid having to run PPP over serial to mimic an Ethernet device on the host.   This allows direct IP communication with the device via a separate route to the serial data.

The RNDIS drivers themselves, and generic inf file, are provided by Microsoft, all we do is supply an inf file that specifies our device ID, and calls into this generic inf file to install these drivers on our device.

6. RECOMMENDATIONS

6.1. Disable USB Selective Suspend.

USB selective suspend is a mechanism built into Windows that allows a Hub to de power devices that can power down when inactive.   This is particularly useful on laptops where power saving is critical, but on desktops it is not. Ordinarily, this is not a problem, since our devices do not advertise to the USB system that they support selective suspend however not all hubs and drivers work as they ought to and they can go into a low power state themselves which kills communication with the device.   So it is recommended to always disable USB selective suspend.

Win 7:

Control Panel->**Power Options**->**Change Plan Settings** (for the active plan) **Change Advanced Power Settings**->**USB Settings**->**USB Selective Suspend Setting** and set the **Plugged In** option to **Disabled**.

Win XP:

Start Menu Right click on **My Computer** and select **Properties->Hardware tab->Device Manager**.

Scroll all the way down to the bottom and expand the + symbol by **Universal Serial Bus Controllers.**

Double click each branch and go to the **Power Management** tab and uncheck the **Allow the computer to turn off this device to save power** option.

6.2. Application timeouts

It is recommended to use a three second timeout when calling SetComTimeouts.
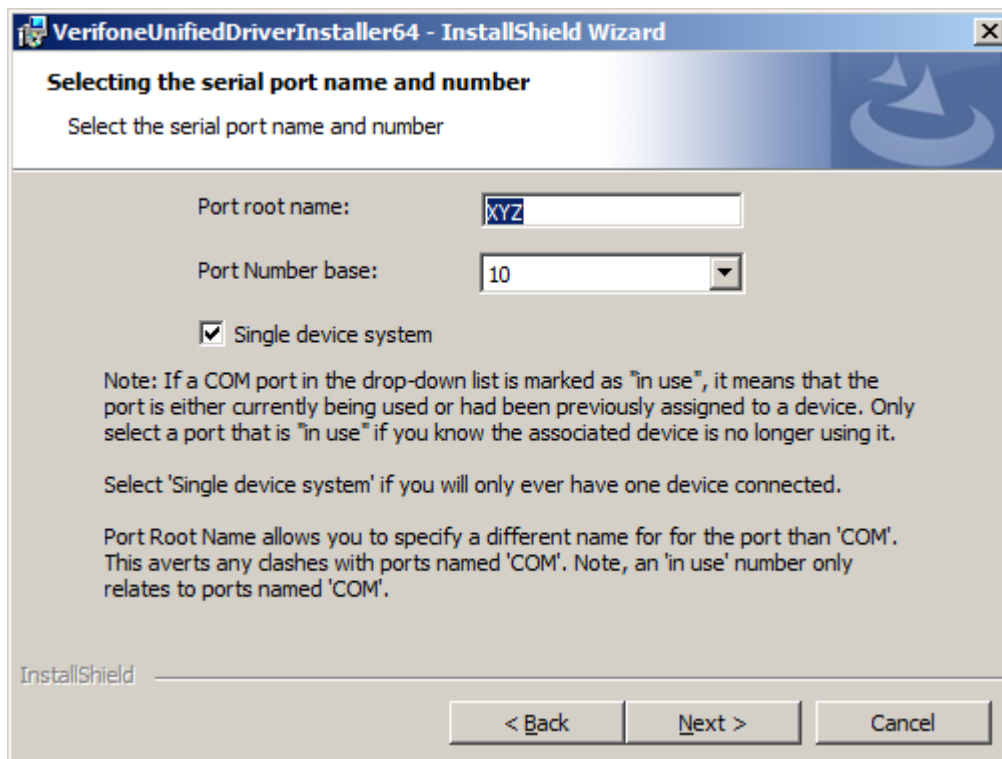
6.3. Windows 7 64

Microsoft changed the signing process in May 2015 which requires the downloading of this update   to   enable   the   drivers   to   load.   https://technet.microsoft.com/en-us/library/security/3033929

7.   INSTALLATION

Installation can be silent, or UI driven. If it is silent, then the requisite information must be specified in a batch file.

7.1. UI Install

7.2. Batch file install

These parameters must be specified when called from a batch file:

PORT=#   Specify port number (eg: "PORT=9", default is 9)

SINGLE_DEVICE_SYSTEM=n   Specify whether the system will have one or many devices attached. 1 for one, 0 for many.

PORT_ROOT_LINK_NAME=ABC   Specify a custom symbolic link name or 'COM'. Must be three letters long.

Sample batch file:

```
@echo off

REM push working dir onto stack so can use relative paths below

pushd "%~dp0"

REM remove previous version(s) first

start /w msiexec.exe /qn /x {<product code GUID>}

start /w msiexec.exe /qn /x {<product code GUID>}

start /w msiexec.exe /qn /x {<product code GUID>}

REM silently install the new version

Start /w msiexec /qn /i VerifoneUnifiedDriverInstaller.msi PORT=9 SINGLE_DEVICE_SYSTEM=1 PORT_ROOT_LINK_NAME=COM

popd
```

*Note, on 64 bit systems the installer is called VerifoneUnifiedDriverInstaller64.msi

*Note The msi will automatically uninstall previous versions on install, but the batch file can also do this explicitly via the Product Codes

7.2.1.   Product code GUIDs

7.2.1.1.   32 bit

From 5.0.1.0 onwards product code for this installer will remain unchanged.

5.0.1.0 Product Code {3DED2043-538D-4725-B7A2-5D8257229D6A}

5.0.0.10 Product Code {72702E87-3800-4B58-B130-7F1F1F274D61}

5.0.0.8 Product Code {72702E87-3800-4B58-B130-7F1F1F274D5F}

5.0.0.7 Product Code {72702E87-3800-4B58-B130-7F1F1F274D5E}

5.0.0.6 Product Code {72702E87-3800-4B58-B130-7F1F1F274D5D}

5.0.0.5 Product Code {72702E87-3800-4B58-B130-7F1F1F274D5C}

5.0.0.4 Product Code {72702E87-3800-4B58-B130-7F1F1F274D5B}

5.0.0.3 Product Code {72702E87-3800-4B58-B130-7F1F1F274D5B}

5.0.0.2 Product Code {72702E87-3800-4B58-B130-7F1F1F274D5A}

Oney Product Code {72702E87-3800-4B58-B130-7F1F1F274D54}

7.2.1.2.   64 bit

From 5.0.1.0 onwards product code for this installer will remain unchanged.

5.0.1.0 Product Code {C7BBB812-4B12-4B50-B8C2-57029675A993}

5.0.0.10 Product Code {F3E17471-B2FC-4911-9C5F-01E1C4FB8DD1}

5.0.0.8 Product Code {F3E17471-B2FC-4911-9C5F-01E1C4FB8DCF}

5.0.0.7 Product Code {F3E17471-B2FC-4911-9C5F-01E1C4FB8DCE}

5.0.0.6 Product Code {F3E17471-B2FC-4911-9C5F-01E1C4FB8DCD}

5.0.0.5 Product Code {F3E17471-B2FC-4911-9C5F-01E1C4FB8DCB}

5.0.0.4 Product Code {F3E17471-B2FC-4911-9C5F-01E1C4FB8DCA}

5.0.0.3 Product Code {F3E17471-B2FC-4911-9C5F-01E1C4FB8DCA}

5.0.0.2 Product Code {F3E17471-B2FC-4911-9C5F-01E1C4FB8DCA}

Oney Product Code {F3E17471-B2FC-4911-9C5F-01E1C4FB8DCC}

7.2.2. RNDIS Hotfix

On XP systems (POSReady 2009 as well), there is a bug in the Microsoft RNDIS driver so this hot fix must be installed:

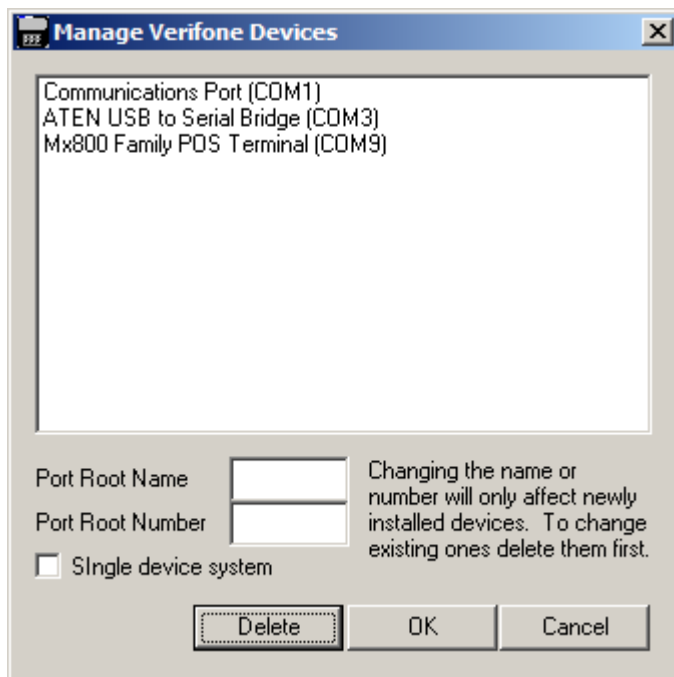https://support.microsoft.com/en-us/kb/959765

8. TOOLS

8.1. VFI Device Manager

Run Verifone Device Manager

This shows all the communication ports on the machine, whether the devices are connected or not.

Selecting one of them, and clicking Delete, removs that device from the system. If it is a Verifone device the number it is using is librated and will be reused by any new devices installed.

The root name and number can be changed here, and affects device installed in the future. Existing devices wil not change.

8.2. VfiCleanup.exe

This is a command line utility that deletes all Verifone devices from the system and deletes all Verifone inf files on the system.

9. LOGGING

9.1. Install and port assignment

During install of new devices, a log file is written to in c:\program files\Verifone\Verifone USB Driver Installer\log.txt

This file specifies the date and port name assigned to devices when installed as well as logging the base information specified during install of the product.

It also logs this information on uninstall.

9.2. Driver

The driver always logs to file, of which there are two, used alternately as they fill up. They are at c:\windows and are called VfiUniUSB0.log and VfiUniUSB1.log

In the driver service section of the registry (HKLM\System\CurrentControlSet\Services\VFIUNIUSB) these values control logging:

FileLoggingOff – setting to zero value turns ON file logging.
FileLoggingOff - setting to non-zero value turns OFF file logging. This is the default settings.
MaximumLogFileSizeInKB - default is 30 MB. Value of 0xFFFFFFFF means unlimited. Otherwise it is 1024 times this value

To interpret the logs read the section 'Loss of Communication with the Device'

10. TROUBLE SHOOTING

10.1. Enumeration

When a USB device is inserted into a host, PC for example, it shows up as a Full Speed device because its D+ line is pulled up to 3volts. To indicate it is a High Speed device, during the mandatory Reset, when the host pulls D+ and D- low, the device applies a lower voltage to the D- line for a period of time. This is called a Chirp-K. If the host is High Speed capable, it sees this, and replies with a stream of Chirp-Js and Chirp-Ks (alternating low voltage on D- and D+). The host copies the device in applying a different resistor to the D+ line resulting in a 500 millivolt differential which is High Speed.
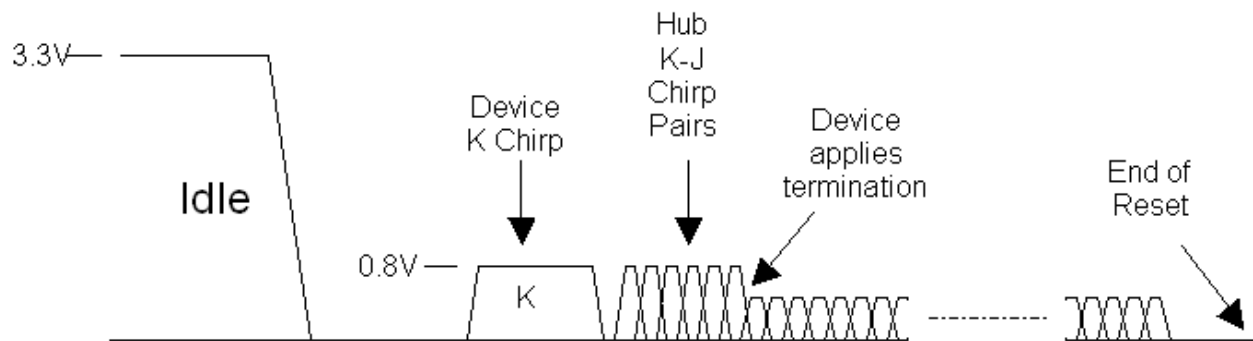
Once the bus communication has been established data can be exchanged. This consists of setup packets.

There are two principle ones, the String Descriptor and the Vendor/Product ID Descriptor.

Typically, the data returned for these are:

Verifone Mx Gadget Serial v2.4

0x11CA 0x0201   The first is the Vendor, and will always be 0x11CA, the second is the product and will vary depending on which product is inserted. These two are important because they give us the next stage of enumeration.

This is what Devicemanager will show if enumeration is successful:
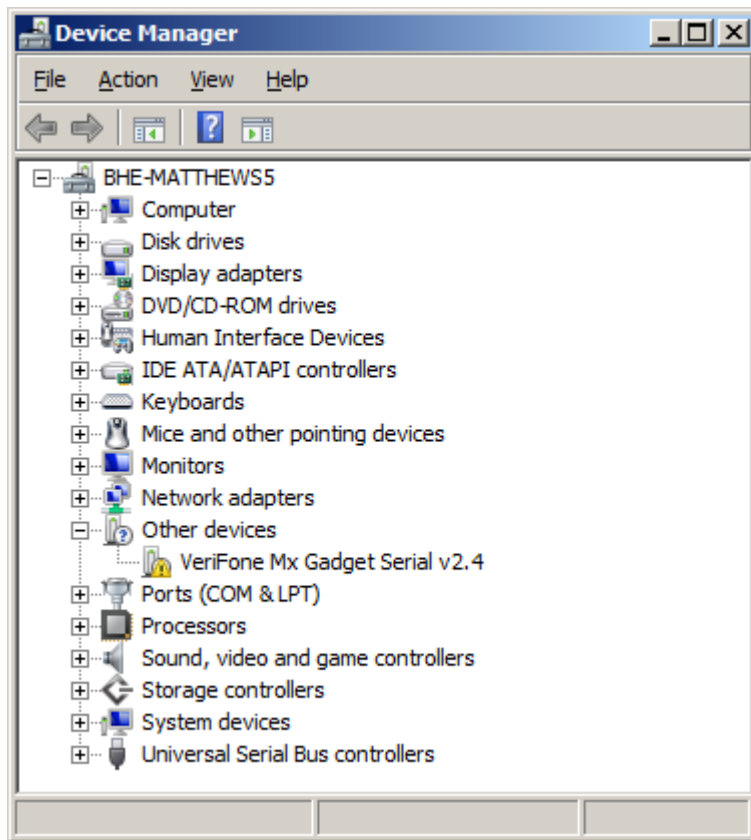
If it is not successful then 'Unrecognised Device' will be displayed instead of 'Verifone Mx Gadget v2.4. So, if you see 'Unrecognised Device' you know the failure was in the hardware enumeration.

If you do see 'Verifone Mx Gadget v2.4' then you know the hardware enumerated correctly, but that the drivers are not installed.

If the drivers are installed the system will use the values for the Vendor and Product IDs, 0x11CA and 0x0201 in this case, to locate the drivers inf file in c:\Windows\inf.

The inf file contains an entry for this hardware ID combination, an associated friendly name for the device, which gets used in place of 'Verifone Mx Gadget v2.4', and an associated service, which is the driver itself.

When the system has all this information, and has registered this hardware with the information in the inf file, Devicemanager will look like this:

VERIFONE, INC.

Verifone®
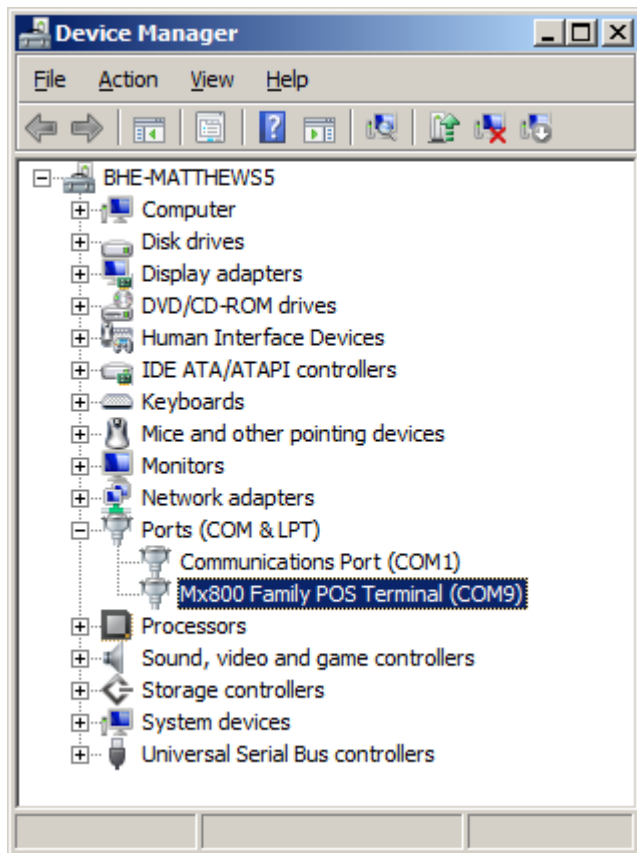
Depending of course on what device you are using, and what port name and base number was chosen during the install.

At this stage all is good and the application can open that port and begin comunicating with the device.

10.1.1.   What can go wrong

So, if Devicemanager shows 'Unrecognised Device' the failure occurred at a hardware level. The device failed to supply its Device Descriptors. A USB bus analyzer might shed light on the cause, but it generally indicates a problem with hardware.

If Devicemanager shows 'Verifone Mx Gadget v2.4' the system can not find the inf file.   The drivers need installing.

If Devicemanager shows a friendly name such as 'Mx800 Family POS' but with a yellow exclamation mark, banged out is a common term for this, then the inf file was found, but the driver failed to load. Right clicking and selecting properties will give more information on this.

10.1.2.   Devicemanager error codes

Code 1	This device is not configured correctly.	• Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly
Code 3	The driver for this device might be corrupted, or your system may be running low on memory or other resources.	• Close some open applications • Uninstall and reinstall the driver • Install additional RAM
Code 10	This device cannot start.	• Look at the installer log to make sure the device was installed correctly • Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly
Code 12	This device cannot find enough free resources that it can use. If you want to use this device, you will need to disable one of the other devices on this system.	• Disable the conflicting device
Code 14	This device cannot work properly until you restart your computer. (Code 14)	• Restart your computer

Code 16	Windows cannot identify all the resources this device uses.	• Assign additional resources to the device
Code 18	Reinstall the drivers for this device.	Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly
Code 19	Windows cannot start this hardware device because its configuration information (in the registry) is incomplete or damaged.	• Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly
Code 21	Windows is removing this device.	• Refresh the Device Manager view • Restart your computer
Code 22	This device is disabled.	• Enable the device
Code 24	This device is not present, is not working properly, or does not have all its drivers installed.	• Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly
Code 28	The drivers for this device are not installed.	• Install the device driver
Code 29	This device is disabled because the firmware of the device did not give it the required resources.	• Enable the device in the BIOS
Code 31	This device is not working properly because Windows cannot load the drivers required for this device.	• Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly
Code 32	A driver (service) for this device has been disabled. An alternate driver may be providing this functionality.	• Uninstall and reinstall the driver • Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly

Code 33	Windows cannot determine which resources are required for this device.	• Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly
Code 34	Windows cannot determine the settings for this device.	• Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly
Code 35	Your computer's system firmware does not include enough information to properly configure and use this device. To use this device, contact your computer manufacturer to obtain a firmware or BIOS update	• Contact the computer manufacturer to update the BIOS
Code 37	Windows cannot initialize the device driver for this hardware.	• Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly
Code 38	Windows cannot load the device driver for this hardware because a previous instance of the device driver is still in memory.	• Restart your computer • Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly • Make sure all applications that have the port open are closed
Code 39	Windows cannot load the device driver for this hardware. The driver may be corrupted or missing.	• Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly
Code 40	Windows cannot access this hardware because its service key information in the registry is missing or recorded incorrectly.	• Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly

Code 41	Windows successfully loaded the device driver for this hardware but cannot find the hardware device.	• Update the device driver • Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly
Code 42	Windows cannot load the device driver for this hardware because there is a duplicate device already running in the system.	• Restart your computer
Code 43	Windows has stopped this device because it has reported problems.	• Use VFIDevMgr to delete the device and reinsert it. Check the installer log to see it deleted and reinstalled correctly • Check the hardware
Code 44	An application or service has shut down this hardware device.	• Restart your computer
Code 45	Currently, this hardware device is not connected to the computer.	• Reconnect the device to your computer
Code 46	Windows cannot gain access to this hardware device because the operating system is in the process of shutting down.	• No resolution required
Code 47	Windows cannot use this hardware device because it has been prepared for safe removal, but it has not been removed from the computer.	• Reconnect the device to your computer • Restart your computer
Code 48	The software for this device has been blocked from starting because it is known to have problems with Windows. Contact the	• Update the device driver

	hardware vendor for a new driver.	
Code 49	Windows cannot start new hardware devices because the system hive is too large (exceeds the Registry Size Limit).	• Uninstall devices that you are no longer using
Code 52	Windows cannot verify the digital signature for the drivers required for this device.	• Reinstall the drivers • If the drivers are known to be Verifone and not Microsoft signed, then the Verifone certificate file will need adding to the system with CertMgr. The file will be found here: C:\Windows\System32\VFISECA.CER • Windows 7 64 bit, install the hot patch detailed earlier in this document

10.2. Loss of communication with device

If the device is correct in device manager, has the right port name, and reports no errors, and USB Selective Suspend is disabled, then the cause of the failure is most likely going to be at the application level (pin pad and PC); it is either failing to open the port or is not sending data to it, or the Pin Pad is not responding to that data.   To confirm this the driver logs can be examined.

What you will be looking for are events such as open/close and read/write, traced as IRP_ML_CREATE, IRP_MJ_CLOSE, IRP_MJ_READ and IRP_MJ_WRITE respectively. These are events initiated by the application on the PC, and correlate to CreateFile() CloseHandler() ReadFile() and WriteFile(). These will be traced twice, first, when sent by the application, which are normally pended, and when completed with a final status, success, error, timeout and so on.

Also look for URB READ and URB WRITE events.   These are requests sent to the USB subsystem (and to the pin pad eventually) and also have completion status' of success, failure, and so on.

Data transfer sizes are also logged.

You will also see many device IOCTLs traced for things like get serial state, clear rts and so on. These are part of com port set up and are normally of little interest.

If you see all these traced, and they show errors then they indicate what might be at fault. So if a URB write failed, then there is a problem on the device side.

If these are all traced with success then the issue will be at a protocol level between the device and the POS, ie, they are shipping data, but aren't understanding what they are receiving.   Or perhaps the application expects the pin pad device to be present on boot sooner than it is, and gives up trying to open it.

If you don't see the application side traces (the IRP_MJ ones), then the app isn't sending data or isn't opening the port.