



# UFCOM - USB Flexible Virtual COM driver user guide

Added by 陈军, last edited by 陈军 on Mar 24, 2017

- [1. UFCOM Introduction](#)
- [2. UFCOM and VCOM management](#)
  - [2.1. UFCOM install and remove](#)
  - [2.2. UFCOM working behavior configuration](#)
    - [2.2.1. VCOM Lifemode](#)
    - [2.2.2. Modify port number for VCOM](#)
    - [2.2.3. Know which application has opened the VCOM](#)
    - [2.2.4. Cleanup stale VCOMs to release port number](#)
    - [2.2.5. Port number Bindmode](#)
  - [2.3. Troubleshooting](#)
    - [2.3.1. Driver package installation troubleshooting](#)
    - [2.3.2. VCOM device is not created \(in Device Manager\)](#)
    - [2.3.3. Diagnose USB data communication error](#)
- [3. VCOM device API behavior](#)
- [4. Document history](#)

This document applies to UFCOM 1.2.0 .

## 1. UFCOM Introduction

---

UFCOM(abbreviation of USB Flexible COM Port) is a brand new Virtual COM Port driver developed by Newland Auto-ID. We provide this driver to our customer since year 2017. This driver works with our barcode scanner hardware; it enables bidirectional communication between your Windows applications and our scanners.

UFCOM provides similar functionality as that of Windows stock usbser.sys and acts as a replacement for usbser.sys. UFCOM not only fixes many ugly behaviors of usbser.sys, but also much more feature rich. Highlights of UFCOM is listed below:

### 1. Multiple barcode scanners support.

UFCOM works with many USB devices(i.e. scanners) at the same time, at least eight. Each scanner will correspond to a separate Virtual COM Port device(VCOM) in your Windows system.

### 2. VCOM Lifemode .

By choosing a VCOM Lifemode, user can control whether the VCOM is still present after the scanner is unplugged from your system. In many cases, we intend the VCOM to remain present after scanner is unplugged(accidentally or purposely), so that when the scanner is plugged in again, the data flow resumes, and the application works smoothly – without experiencing ineffective device handles in the midway.

### 3. Port number Bindmode .

By choosing a Bindmode, user can decide whether VCOM port number association behavior is determined by scanner device-type or by device-serial-number.

### 4. Dynamic device display name .

In Windows Device Manager, VCOM device display-name dynamically reflects device working state. For example, a normal working scanner will have [online] prefix, while an unplugged scanner will have [offline] prefix.

### 5. VCOM Settings UI

Check into Device Manager's property dialog, there is a VCOM Settings UI section, where users can conveniently adjust various properties of VCOM devices, e.g., viewing and changing Lifemode, Bindmode and port number. You can even check USB link layer errors there.

### Some background information for our existing customers

Before UFCOM, we used to provide two sets of Virtual COM Port drivers to our customers.

- A so-called USB Datapipe driver: You need this when scanner USB mode is set to "USB Datapipe" or "com0com virtual COM port".
- CDC Virtual COM driver: You need this when scanner USB mode is set to "USB-CDC" and Windows is 8.1 or prior. This driver actually refers Windows stock usbser.sys . This means, you will not be able to upgrade this driver even if Microsoft has new versions available. For example, You can use new usbser.sys from Windows 10 onto your Windows 7.

Now, UFCOM is a thorough replacement of the two. That is, whether you set your scanner USB mode to any one above, UFCOM will create virtual COM ports for them.

UFCOM supports legacy Datapipe API, so legacy applications that use udp\_op.dll can work with UFCOM. But note: some applications may be buggy and new version is required to work with UFCOM.

## 2. UFCOM and VCOM management

### 2.1. UFCOM install and remove

#### 【 background information 】

For our customer to be bright on Windows system driver installation, I'd like to give some key introduction on this topic. "Driver installation" actually consists of three steps.

1. Copy a **driver package** to a system folder called **DriverStore**. On Windows 7 and above, this folder is `C:\Windows\System32\DriverStore\FileRepository` . A driver package normally contains files types of .inf, .sys and .cat , and possibly some dll and exe The inf file is the key, you can think of an inf file representing a driver package.
2. Windows finds a driver package match for a detected device. It means, when the device(scanner) hardware is detected by Windows, the device reports its hardware-id to Windows, Windows then searches all inf files inside DriverStore for matches. It is possible to find multiple matches for a single hardware-id, but Windows at the same moment will pick only one to drive the device. Normally, Windows will pick the best one, typically, the one with digital signature and with newer date.
3. Load the driver code(.sys) into memory and execute the code. From this time one, hardware starts to exchange data with Windows.

When Windows has determined a driver match, this matching is recorded into registry, and stay there even if the device is unplugged. So, when the device is plugged in a second time, Windows can skip the driver match step, and use the previously recorded one as driver to load into memory. But if you Uninstall a device from Device Manager , the matching information is deleted from registry, so next time device plugging-in will trigger driver match searching again.

#### 【 UFCOM driver package installation 】

UFCOM driver package is provided as zip file. After extracting, we see an install.bat , run it and the installation is done. Install.bat calls Microsoft provided DPInst.exe to achieve its goal. Install.bat does the following:

- Copy UFCOM driver package to DriverStore .
- If Windows has existing devices that have been matched with UFCOM driver, these devices will be updated to match UFCOM driver package(if it's a better match).
- If old driver file cannot be unloaded from memory(COM port being opened for example), Windows will prompt you to reboot the whole system.

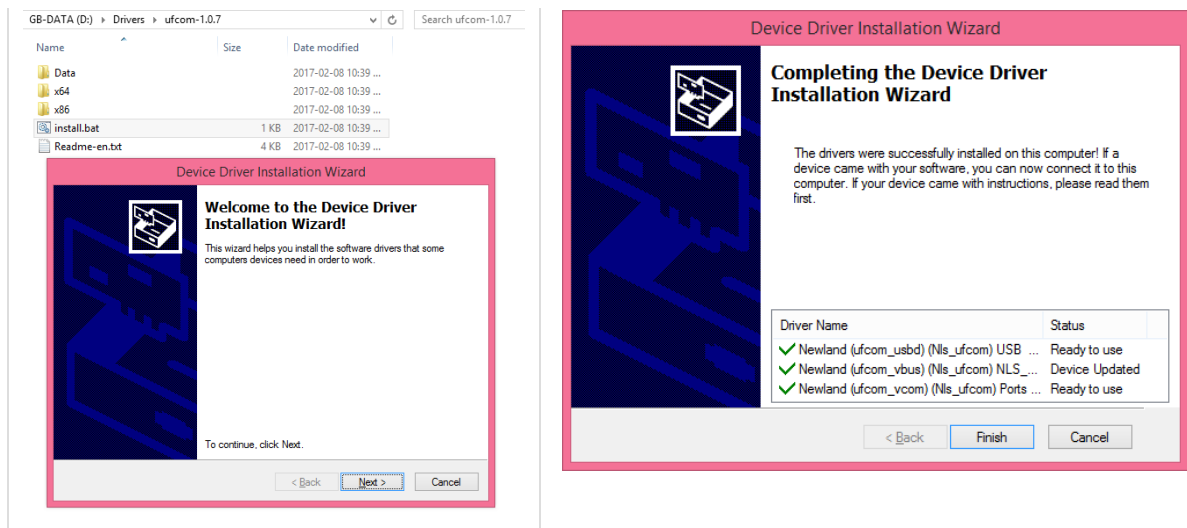
For first time installation, it does not matter you plug in scanner first or run installation package first. Both are OK.

We suggest you close all applications operating the scanner before running installation, this can reduce the chance you need to reboot Windows.

On installation wizard dialog, just click Next, wait some moment, you will be told the result. If you see three green ticks, it means success.

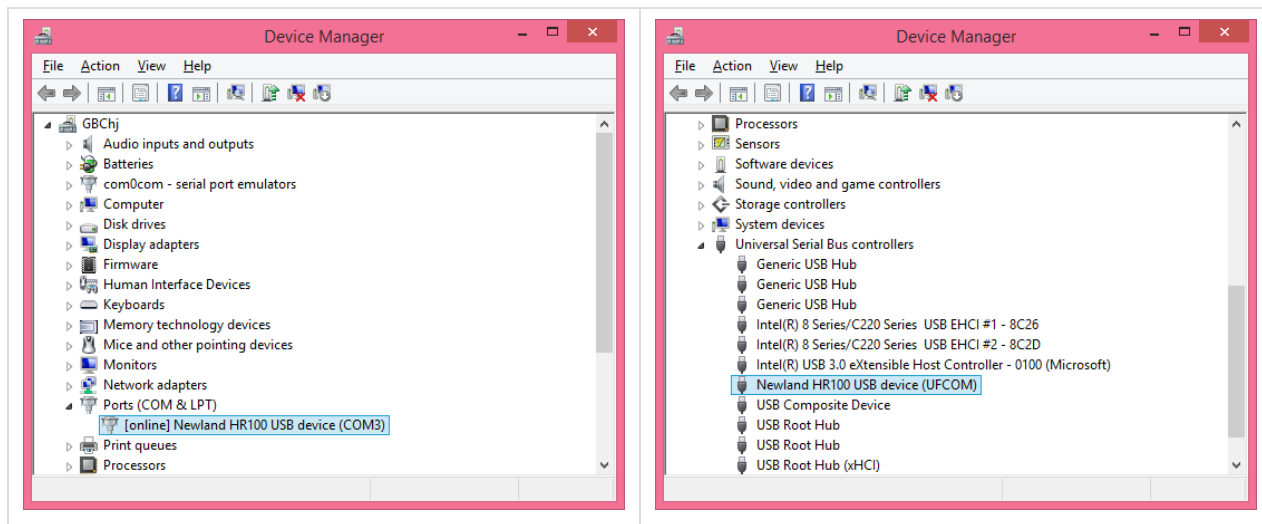
If Windows prompts for reboot, do so to ensure installation completes.





Now plug in your USB scanner, Windows now searches and loads driver for it. For first scanner plugged in, it costs several to tens of seconds, depending on your Windows system performance.

To confirm that scanner is driven successfully by UFCOM, you can check it with Device Manager. In Device Manager, we actually see two new devices for a physical scanner. One is the VCOM device, another is your real device(or call it physical device).

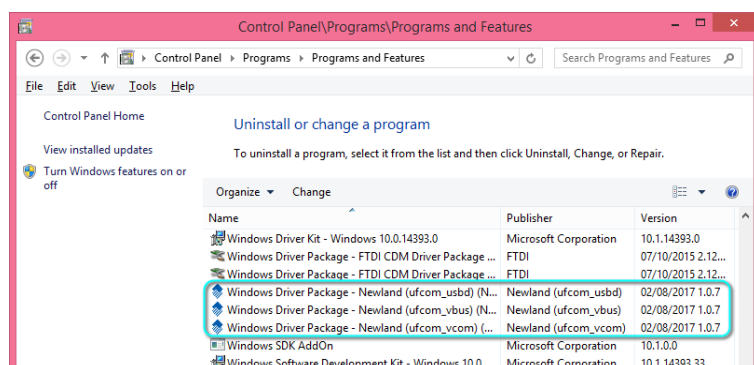


If the scanner is unplugged, the physical device node will definitely disappear(unplugged state), but the VCOM device node does not necessarily disappear. Whether VCOM should disappear is determined by VCOM Lifemode which you can control. VCOM Lifemode is explained later.

## 【 Uninstalling UFCOM 】

From Control Panel, you can uninstall UFCOM driver package.

For each version of UFCOM, you get three items in Programs and Feature, with "ufcom" in their names.



If really required, you can uninstall them. Un-installation here means:

- UFCOM Driver package is removed from DriverStore .

- Device nodes previously driven by UFCOM will receive new driver matching, which may get a match of older driver or a stock Windows driver.

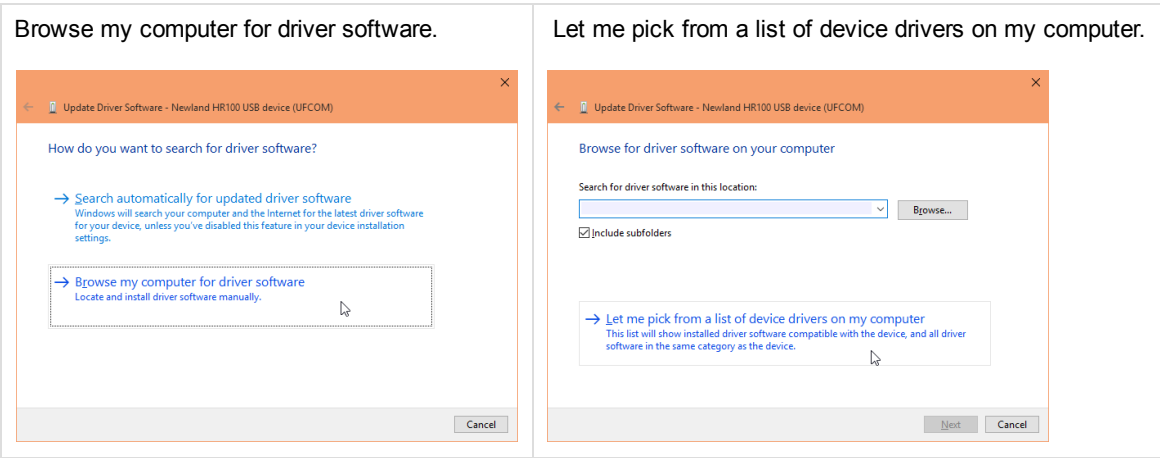
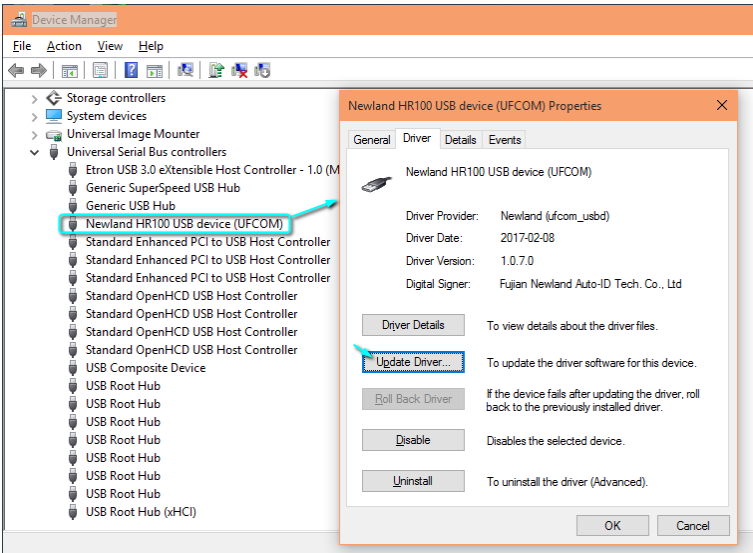
Note: For each UFCOM driver package installation, we do not automatically remove old drivers. This should not be a problem, because Windows always consider the latest driver(by timestamp) as best match.

【 Manually choosing a driver package for USB device 】

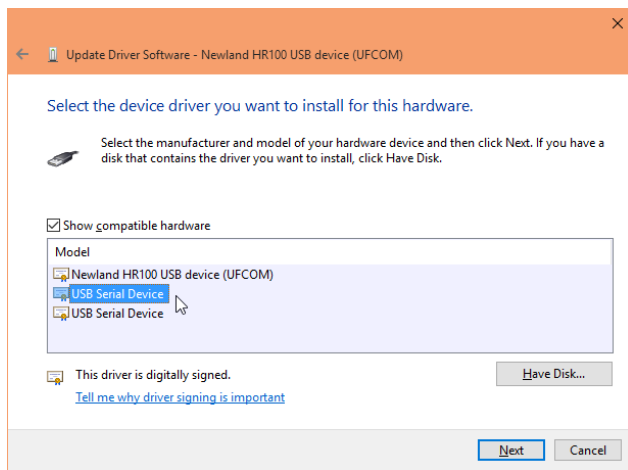
If for any reason you need to go back to the old Datapipe driver, to an older UFCOM version, or Windows stock driver, you don't have to uninstall UFCOM, because there is a simpler way provided by Windows itself.

Use Windows 10 as example, do it as follows:

Find the physical device node in Device Manager(NOT the VCOM node), open its properties, click Update Driver.



In Show compatible hardware list box, pick a package to apply. Here, a driver-package is identified by "device model name" inside its corresponding inf file. In the example illustrated below, "Newland HR100 USB device(UFCOM)" represents UFCOM driver package, and "USB Serial Device" represents Windows stock usbser.sys .



Click Next a final time. To confirm your selection.

Of course, you can switch back to UFCOM driver by the same way.

## 2.2. UFCOM working behavior configuration

For UFCOM generated VCOM device, you can open it with any COM port communication software, operate it with standard Windows COM port API(or called Serial port API). Any data you read/write comes from/goes to USB device itself. On Windows XP, you can use HyperTerminal; on Windows 7 and above, no HyperTerminal any more, so you can use free [PuTTY](#) . For some advanced configuration of your scanner, we provide EzSet software.

Due to the fact the Windows COM port API is designed with physical RS-232 serial port, it does not make consideration about USB device's special traits, like unpluggability at any time, power management. So UFCOM needs to design special way to control these special traits.

### 2.2.1. VCOM Lifemode

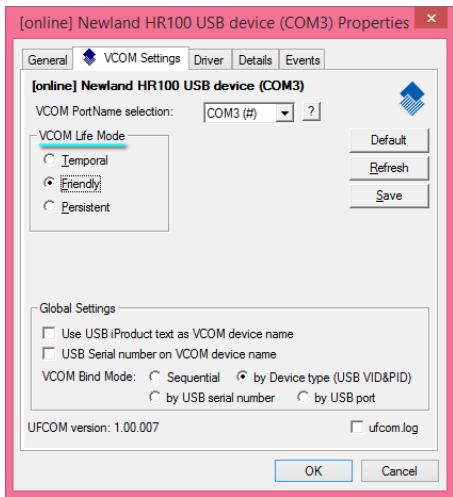
VCOM Lifemode determines when should VCOM appear(be present), when should it disappear. "Disappear" or called "non-present" means: You will not see it in Device Manager – unless you tell Device Manager to display hidden devices.

Lifemode	VCOM display prefix	Conditions that VCOM present
Temporal	<online>	When scanner plugs in, VCOM appears; when scanner unplugs, VCOM disappears immediately.
Friendly (default)	[online] [offline]	When scanner is plugged in, and there is application using this VCOM(has a opened handle of this VCOM), VCOM will not disappear, and the already opened handle remains valid. Later when the scanner re-plugs in, the communication channel is re-established. That means: an application holding a VCOM handle will not be interfered with temporary plug-in/unplug of the physical device.  When scanner-unplugging and VCOM handle closing are both true, VCOM will disappear.
Persistent	[[online]] [[offline]]	VCOM exists forever.  Persistent mode makes a VCOM look almost like a on-board physical COM port. It always exists no matter scanner is plugged-in or unplugged.

Meaning of VCOM display name prefix "online, offline":

- "online" means scanner is plugged in and the driver works normally.
- "offline" means scanner is physically unplugged.

Method to set Lifemode: In Device Manager, locate a VCOM device node(not physical device node), open its properties dialog. Switch to "VCOM Settings" tab, where you can set Lifemode for this VCOM.

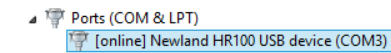


NOTE: If you are using two scanners, there will be two VCOM devices, and their Lifemode is configured separately.

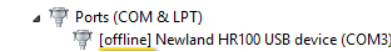
You need administrative rights on Windows to make changes to VCOM Settings.

【 Observing VCOM device "online/offline" prefix change 】

Plug a scanner into Windows , the corresponding VCOM device display name will have a "online" prefix.



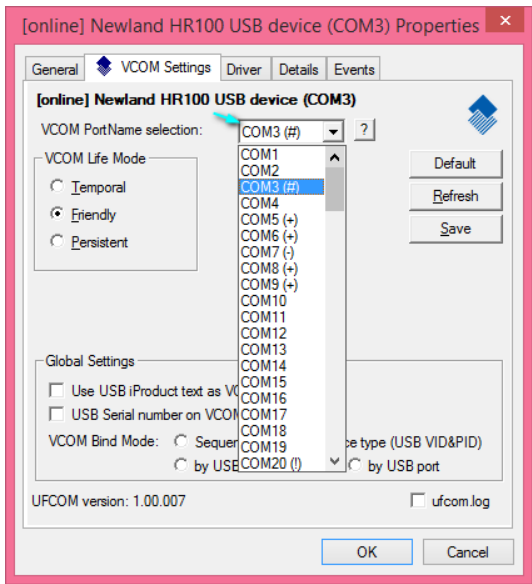
Unplug the scanner, and, if the VCOM should remain present according to its Lifemode rule, the prefix becomes "offline".



### 2.2.2. Modify port number for VCOM

This is a widely used feature.

When Windows creates a VCOM device for us, the Windows system picks a free(not yet reserved) port number for this VCOM, normally picking the one with lowest number available(from COM3 and above). If we are unsatisfied with the auto-picked one, we can change it. With VCOM Settings UI, you can do it without hassle.



Before picking a new port number(precisely called PortName), please be aware of the status mark alongside each PortName entry. These marks have special meanings:

Mark	Meaning
------	---------

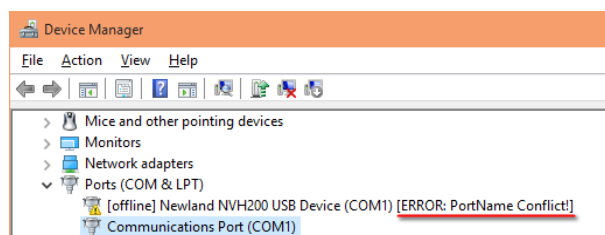
#	This port number is being used by the VCOM itself.
!	This port number is being occupied by another COM port device, and this another COM port device now truly exists on the system(plugged-in state).
+	Some other UFCOM VCOM device has reserved this port number, and this other device is currently unplugged.
-	Windows ComDB database has reserved this port number for some other device, which is also currently unplugged.

No need to memorize those symbol marks. Clicking the small question mark button at right side, you get these explanations.

We suggest selecting a port number with no mark, which means not occupied or reserved, so you reduce the chance to interfere with other devices on the system.

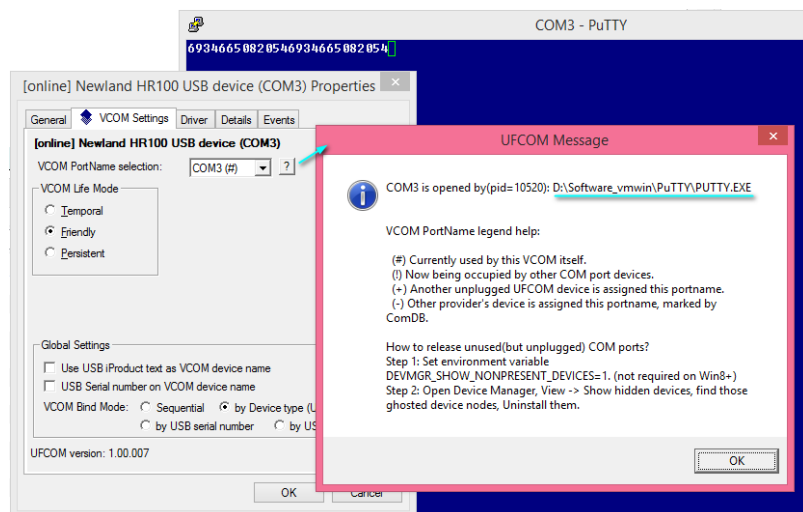
- If you select a port number with (!) mark, it will definitely cause error.
- Selecting (+) (-) mark may be OK, COM7 from the image for example. But be aware, when the original COM7 device(maybe a Virtual COM Port from another vendor) is re-plugged in, that one may or may not work normally. The final behavior is determined by that driver's code logic.

If you choose a port number with (!) mark, UFCOM will indicate the error reason on device display name. For example: When on-board COM1 exists and you force changing a VCOM to using COM1, Device Manager will show this VCOM as working abnormally and the device display name will have prompt text "[ERROR: PortName Conflict!]".



### 2.2.3. Know which application has opened the VCOM

Click the small question mark button beside PortName list box, we can know whether the VCOM is opened by some application. The example image below shows: COM3 is being opened by PUTTY.EXE .



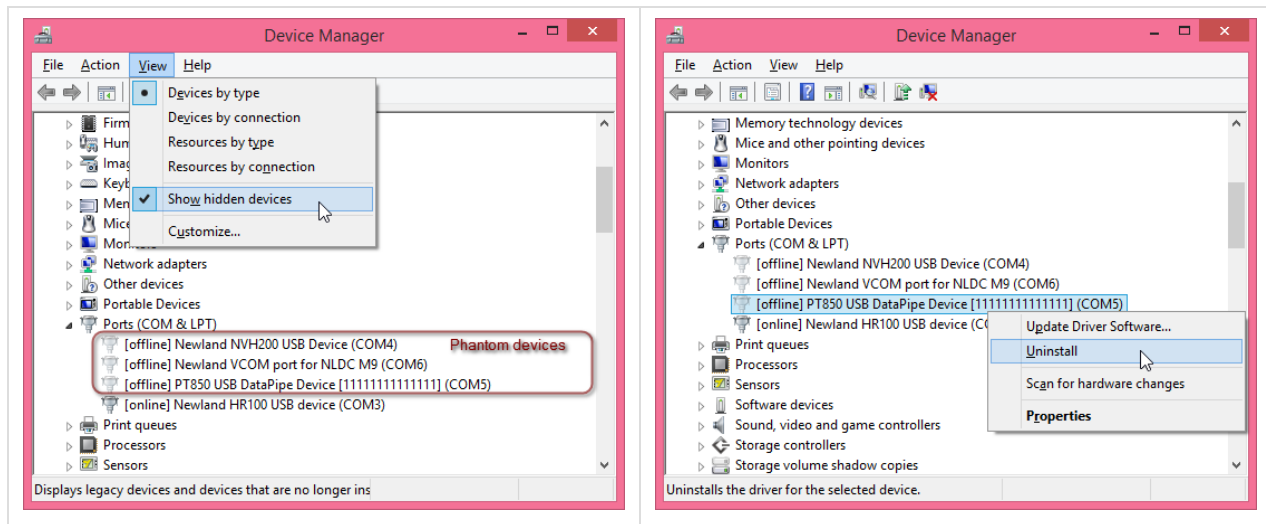
### 2.2.4. Cleanup stale VCOMs to release port number

After we unplug a scanner and close its VCOM handle, we see corresponding VCOM device node disappearing from Device Manager, but actually, it is only hidden. The VCOM related settings are still recorded in Windows registry. These settings include: VCOM port number, Lifemode etc. When the same VCOM device instance appears again, Windows will re-apply those settings. This is reasonable design. Tech note: a VCOM device instance is identified by the so-called "Device instance path", which can be queried from device properties, Details tab.

After many scanners have been plugged and unplugged, many port numbers would have been reserved. If we want to reclaim those port numbers so that they becomes free again for others, the best way is to "Uninstall" the old device nodes.

Way to do that: Open Device Manager, enable menu item View → Show hidden devices. Now we can see some more devices with dimmed icons, each of which represents an unplugged but still registered device. We call them phantom devices. Right click and Uninstall a phantom device, the corresponding port number is reclaimed.





**NOTE:** On Windows XP~Windows 7, in order to see phantom devices, we have to do an extra step, that is, add an environment variable `DEVMGR_SHOW_NONPRESENT_DEVICES=1` then restart Device Manager.

No need to do this since Windows 8.

## 2.2.5. Port number Bindmode

COM Port Bindmode is a global setting. It governs the rule how a physical scanner gets associated to a COM port number.

UFCOM provides four Bindmodes, You need some imagination to understand them.

Bindmode	Meaning
Sequential	<p>UFCOM considers all scanners equal. The result is, port number is allocated one by one, typically sequentially.</p> <p>Example: If we have two scanners, plugging in scanner A, we get COM3; plugging in scanner B at the same time, we get COM4. Later we will see, we unplug both scanners and plug in either one, the one will always get COM3 .</p>
by Device type (default)	<p>Simply speaking, UFCOM takes consideration of scanner device type. Different device types will get different COM port numbers. Scanner device type is identified by VID,PID tuple from USB device descriptor.</p> <p>Example: Assume we have an HR100 and an HR200 (different types). First time plugging in HR100, we get COM5; first time plugging in HR200, we get COM6. Later we will see: Unplug and replug the two scanners, HR100 will always get COM5, HR200 will always get COM6, regardless of whether the other scanner is present or not. So , COM5 is bound to HR100 and COM6 is bound to HR200. This binding relation persists until you Uninstall corresponding VCOM devices.</p>
by USB serial number	<p>Devices with different serial numbers will get different COM port numbers. For a USB device, the serial number is determined by iSerialNumber text reported in USB device descriptor.</p> <p>When we need to bind two same-type scanners to their own VCOM, we can use this Bindmode. But note: This can be achieved only if both scanners has serial numbers burned-in. For scanners with no serial number, Bindmode behavior will fall back to be "by device type".</p>
by USB port	<p>Devices at different USB ports will get different COM port numbers. So the device type and serial number is not considered.</p> <p>This Bindmode is available only in Windows 7 and above.</p>

Extra notes:

- Currently, switching Bindmode will cause all VCOM opened handles to become ineffective. Applications have to reopen VCOM to get new handles.
- When switching Bindmode. The port numbers used by previous bindmode is not reclaimed by system. So, switching bindmode usually causes the same scanner to receive a different port number. If you want the former port number, just manually change it.

## 2.3. Troubleshooting

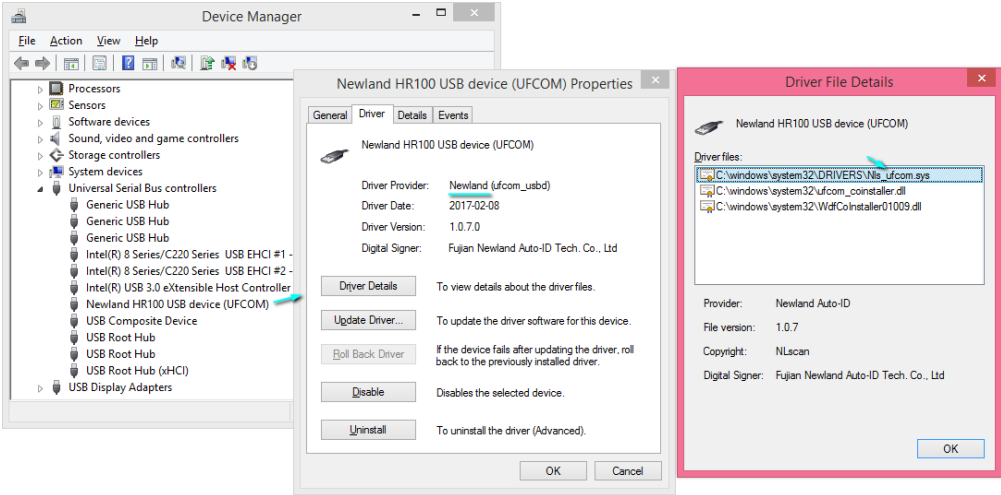
### 2.3.1. Driver package installation troubleshooting

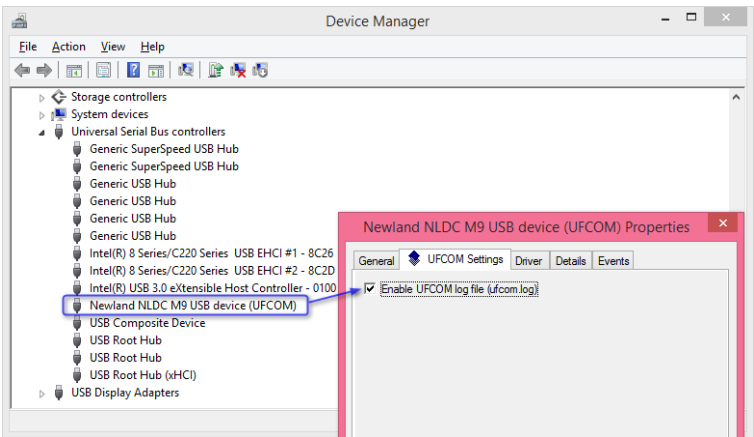


None yet.

### 2.3.2. VCOM device is not created (in Device Manager)

There are many causes.

Category	Symptoms, cause and solutions
USB physical device is not detected by Windows	<p>Symptom: In Device Manager, in "Universal Serial Bus controllers" category, you do not see new device node. You do not see any new device node in other device categories either.</p> <p>Possible reasons:</p> <ul style="list-style-type: none"><li>• Scanner is not actually plugged into Windows machine.</li><li>• Scanner has not enabled USB function.</li><li>• Scanner has hardware problem.</li><li>• Windows does not have correct USB system drivers(USB host controller drivers) installed.</li></ul> <p>Checking physical device presence is done by Windows itself, not a matter of UFCOM.</p>
USB scanner is working as a keyboard	<p>Symptom: Scanner works, and scanned barcodes go into text editor(Notepad for example).</p> <p>This means: The scanner is working in USB keyboard mode(typical our factory default). Your scanner needs to scan a special setting code to make it transform into a USB mode that can work with UFCOM(USB-CDC for example).</p>
Windows detects USB physical device, but uses another driver	<p>Symptom: USB physical device display name does not end with "(UFCOM)".</p> <p>To confirm: Double click USB physical device node to check its driver properties. There should be Newland text in it, and Driver Details section refers to driver file Nls_ufcom.sys .</p>  <p>If you find that another driver is taking control of your scanner(Windows stock usbser.sys or the old Datapipe driver), please switch the driver to be UFCOM, as told in former section "Manually choosing a driver package for USB device".</p>
USB physical device is detected, but UFCOM driver does not work correctly	<p>Symptom: USB device icon has a yellow exclamation mark on it.</p> <p>First, confirm that the scanner has been assigned UFCOM as its driver; if not, do it first. If the yellow exclamation still exist, consider the following reasons:</p> <ul style="list-style-type: none"><li>• Scanner's firmware is too old to support UFCOM. Please contact Newland to get a new firmware version.</li><li>• USB hardware problem. The problem may reside in PC, USB cable, or scanner itself. Check device properties dialog for simple error reason, and contact us for help.</li></ul>

Category	Symptoms, cause and solutions
USB physical device looks OK in Device Manager, but VCOM does not appear.	<p>This is rare. If you encounters this, you can help us diagnose it by sending use UFCOM log file. Enable log file, reproduce the problem, send the log file to us.</p>  <p>Log file location is C:\Windows\Temp\ufcom.log . This file will be cleared and regenerated every time UFCOM driver is loading(e.g., after Windows reboots). So it won't bloat forever.</p>
VCOM device looks OK, but application does not communicates correctly.	<p>The reasons can be complicated. See next section.</p>

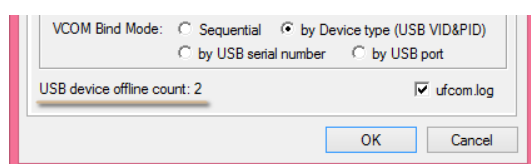
### 2.3.3. Diagnose USB data communication error

Sometimes, we may encounter data communication error with USB hardware, for example, not able to receive barcode data from VCOM, configuration app(EasySet etc) not able to detect attached scanners, Easyset failing to retrieve captured images etc. Before contacting support staff, we can try to identify the reason. In common sense, communication error may arise in the following segments:

- (A) Hardware/software inside USB Scanner itself.
- (B) Physical link between USB scanner and your PC.
- (C) PC side USB hardware and USB host controller driver(this driver is provided by Windows or host controller manufacturer).
- (D) PC side client driver, i.e. UFCOM.
- (E) The application that uses VCOM.

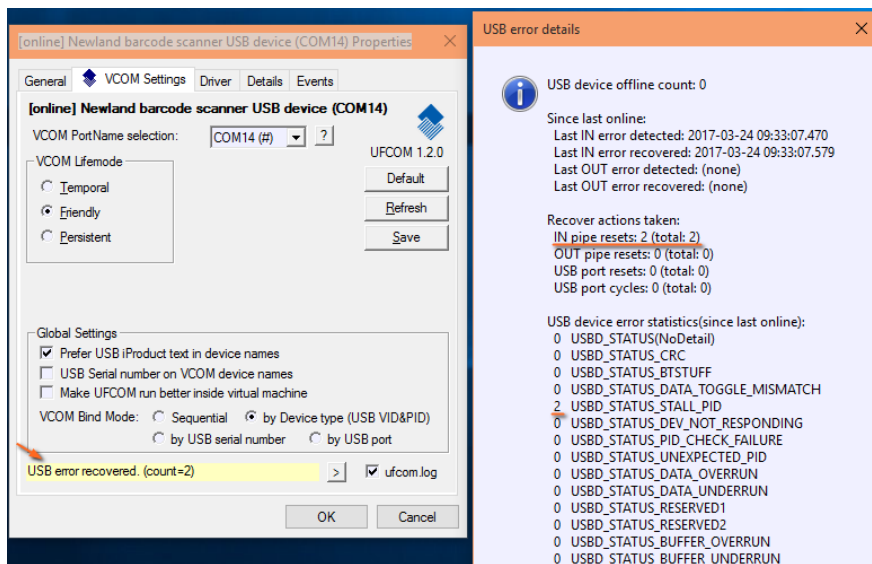
For average users, determine the faulting segment can be hard. Luckily, UFCOM can help you identify whether B is. When using poor quality cable, using buggy USB hub, receiving not enough voltage supply, or encountering electromagnetic interference, we can get faulting condition at B.

VCOM Settings UI's bottom status bar can tell you USB devices offline count. We know that manually unplugging the USB device will increase its count; but if you get increased count while in normal using conditions, it probably means there is problem with USB signals, which is a sign of faulting hardware.



In other circumstances, USB transfer error does not cause "offline", but exhibits a UsbdStatus error code reported to USB host controller driver, and Windows reports that error to UFCOM, UFCOM in turn presents such error to end user through VCOM Settings UI. In a well behaved operation environment, we do not expect to see even a single such error. If we see such an error, it implies some problem in the transfer system, probably in hardware. Sporadical appearing of such error can be recovered by UFCOM automatically, and the VCOM application has a good chance to work smoothly without being interfered; but if it arises too frequently, the application may not work as expected, and we have to go with further troubleshooting.

When UFCOM detects UsbdStatus error, VCOM Settings UI will report it to us. In the following figure, UFCOM detects two such errors, and they have been recovered. Click a small arrow at right-side, we get more error details. We can see that they were two USBD\_STATUS\_STALL\_PID errors.



Be aware, in case the USB link is every unstable, the recovery action may not guarantee the link can become normal, even if UFCOM says "USB error recovered".

Note: The error statistics is per VCOM, that is, each VCOM has its own statistics. That statistics data persist until the VCOM device node is unplugged(disappeared from Device Manager).

If the problem exists, you may need to contact us for help.

### 3. VCOM device API behavior

UFCOM implements Windows COM port API(serial port API) quite thoroughly.

If you program with Windows COM port API, you need to know information in this section.

API category	Implementation note
RS-232 parameter simulation	<p>No simulation for this. User can set any baudrate, but the value is simply ignored. Actually communication speed is determined by the USB system.</p> <p>Data bits, stop bits, parity settings are all ignored.</p>
Modem signal simulation	<p>Output signals: RTS, DTR setting is ignored.</p> <p>Input signals: GetCommModemStatus always report RING and RLSD off . DSR and CTS is simulated.</p> <p>DSR: When scanner is plugged in, DSR=On ; when unplugged, DSR=Off .</p> <p>CTS: When scanner is plugged in and down stream is not blocked, CTS=On ; when scanner is unplugged or down stream is blocked, CTS=Off .</p>
hardware flow control	<p>UFCOM does not support upstream(input) flow control with simulated modem signal. Upstream flow control is entirely determined by USB system.</p> <p>UFCOM supports downstream(output) flow control.</p> <ul style="list-style-type: none"> <li>When VCOM is first opened by application, flow control is not enabled;</li> <li>After application calls SetCommState() with DCB . fOutxCtsFlow=1 , the downstream flowing can be controlled, e.g., If CTS=Off, WriteFile will be blocked.</li> </ul>
software flow control	XON/XOFF flow control is not supported.
WriteFile behavior	UFCOM VCOM's WriteFile accepts user data atomically. That is, WriteFile will complete with <u>all requested data</u> or <u>no data</u> ; you will not see partial data accepted.

API category	Implementation note										
COM port timeouts	<p>Implement COMMTIMEOUTS semantic completely.</p> <p>There are some special designs:</p> <ul style="list-style-type: none"> <li>When WriteTotalTimeoutMultiplier=0 , WriteTotalTimeoutConstant=0 , and <u>output flow is not enabled</u>. UFCOM guarantees WriteFile will not block forever. If scanner is unplugged or downstream flow blocked, WriteFile will <b>complete with success</b> after there is no downstream flow for continuous 1 second. This is to simulate that WriteFile to a on-board physical port will not block forever.</li> <li>When WriteTotalTimeoutMultiplier=0 , WriteTotalTimeoutConstant=0 , and <u>output flow is enabled</u>, WriteFile will block before all data is written out. Of course, user can call Canceled to cancel WriteFile prematurely.</li> <li>When WriteTotalTimeoutMultiplier or WriteTotalTimeoutConstant is non-zero, API-assigned timeouts will be effective, regardless whether flow control is enabled.</li> </ul>										
CommEvents	<p>UFCOM implements four notification events.</p> <table> <tr> <th>Event</th><th>When to notify</th></tr> <tr> <td>EV_RXCHAR</td><td>When there is bytes in input buffer, this event arises. It's level-triggered, that is, as long as there is any byte in input stream, WaitCommEvent will always report this event.</td></tr> <tr> <td>EV_TXEMPTY</td><td>When output buffer becomes empty, this event arises. It's edge-triggered, that is, after EV_TXEMPTY is reported once, waiting this event again will not report a second EV_TXEMPTY; it is reported again only after some bytes are filled into output buffer by WriteFile and becomes empty again.</td></tr> <tr> <td>EV_DSR</td><td>EV_DSR is edge-triggered, representing simulated DSR change.</td></tr> <tr> <td>EV_CTS</td><td>EV_CTS is edge-triggered, representing simulated CTS change.</td></tr> </table> <p>UFCOM allows multithreaded operation. You can call WaitCommEvent in one thread, and ReadFile/WriteFile in another thread.</p>	Event	When to notify	EV_RXCHAR	When there is bytes in input buffer, this event arises. It's level-triggered, that is, as long as there is any byte in input stream, WaitCommEvent will always report this event.	EV_TXEMPTY	When output buffer becomes empty, this event arises. It's edge-triggered, that is, after EV_TXEMPTY is reported once, waiting this event again will not report a second EV_TXEMPTY; it is reported again only after some bytes are filled into output buffer by WriteFile and becomes empty again.	EV_DSR	EV_DSR is edge-triggered, representing simulated DSR change.	EV_CTS	EV_CTS is edge-triggered, representing simulated CTS change.
Event	When to notify										
EV_RXCHAR	When there is bytes in input buffer, this event arises. It's level-triggered, that is, as long as there is any byte in input stream, WaitCommEvent will always report this event.										
EV_TXEMPTY	When output buffer becomes empty, this event arises. It's edge-triggered, that is, after EV_TXEMPTY is reported once, waiting this event again will not report a second EV_TXEMPTY; it is reported again only after some bytes are filled into output buffer by WriteFile and becomes empty again.										
EV_DSR	EV_DSR is edge-triggered, representing simulated DSR change.										
EV_CTS	EV_CTS is edge-triggered, representing simulated CTS change.										
Canceled	<p>Fully supported. User can Canceled ReadFile, WriteFile, WaitCommEvent.</p> <p>Canceled will succeed immediately, no blocking.</p> <p>Special note: If VCOM has received some bytes when ReadFile is cancelled, ReadFile will complete with success and report those already received bytes. This behavior is better than that of serial.sys and usbser.sys. The serial.sys and usbser.sys merely reports error result 995(ERROR_OPERATION_ABORTED) for ReadFile on cancelling.</p>										

#### 【 Special note for Temporal Lifemode 】

For Temporal Lifemode VCOM, corresponding USB physical device unplugging will render the VCOM handle ineffective, which means, calling ReadFile, WriteFile, WaitCommEvent on such ineffective handle will report error immediately – even if USB device has been re-plugged in. In order to recover communication with the USB device, you have to close the old handle and call CreateFile to get a new one.

## 4. Document history

UFCOM version	Document updates brief
1.2.0	USB error detection and recovery.
1.1.0	UFCOM log file can be enabled from USB device Properties dialog. Supports Datapipe API.
1.0.7	Initial release.