

HOWTO

Cleaning the Windows USB Device Database and Installing Correct FTDI Drivers for TI mmWave Boards

A practical procedure for resolving residual USB/FTDI conflicts on Windows 10 Pro

Willie Chaffin | August 2026 | TRIDRONE.com

1. Problem Statement

A Windows 10 Pro system developed residual USB device entries that prevented a clean installation of FTDI drivers required by a Texas Instruments IWRL6432 (BP-IWRL6432WMOD) evaluation board. The FTDI USB port on the board should enumerate as four healthy COM ports named AR-DevPack-EVM-012. Instead, connecting the port and running TI Visualizer produced a Blue Screen of Death (BSOD).

The official FTDI uninstall utility failed to remove previous installations completely. Attempting to finish the install using raw INF files and generic FTDI packages only made the device database worse. The root documentation problem was that TI's user guide points to a tools\ftdi folder inside the mmWave L SDK that no longer exists in current releases.

Without giving the nasty details that precipitated this problem I can say it was encouraged by Texas Instruments error of omitting the proper drivers where their documentation said they would be after installing MMWAVE_L_SDK_05_05_04_02. They were NOT in the <install path>\tools\ftdi folder as their guide explained. This is what causes hunting, trial and error problems technologist dread.

2. Critical Background – Vendor IDs

Before any cleanup, identify the correct USB Vendor ID (VID) of the residual devices.

Key facts:

- FTDI's standard Vendor ID is **0403**.
- Texas Instruments' primary Vendor ID is **0451**.
- Many TI development boards (including the IWRL6432) use real FTDI silicon, but the EEPROM is programmed with TI's VID 0451 and custom Product IDs (fd03 for the four AR-DevPack channels, bef3 for the XDS110 debug probe). That is why no 0403 entries appear, yet FTDI driver components are still involved.

Public reference for assigned Vendor IDs: USB-IF "Valid USB Vendor ID Numbers" at <https://www.usb.org/developers>. A practical community database of VID+PID combinations is available at <http://www.linux-usb.org/usb-ids.html>.

3. Tools Required

- **USBDeview** (NirSoft) – current version used was 3.2. Run as Administrator.
- **Device Manager** with "Show hidden devices" enabled.
- **Registry Editor** (regedit) – for COM Name Arbiter cleanup.
- **mmWave Studio 04.01.00.06** (MMWAVE-STUDIO-3G) – the package that currently contains the correct TI-signed FTDI drivers when the SDK tools\ftdi folder is missing.

4. Cleanup Procedure

4.1 USBDeview – Remove Residual Devices

Launch USBDeview as Administrator. Enable display of disconnected devices.

1. Select all disconnected residual devices with Vendor ID **0451**. On the IWRL6432 these typically include:
 - a. Four AR-DevPack-EVM-012 A/B/C/D entries (PID fd03) – the FTDI USB side.
 - b. XDS110 Class Application/User UART, Debug Probe, Auxiliary Data Port (PID bef3).
 - c. Any associated USB Input Device (HID) entries under the same VID.
 - d. Parent USB Composite Device entries that still carry VID 0451.
2. Right-click → Uninstall Selected Devices → confirm.
3. Press F5 to refresh and verify the selected entries are gone.
4. Do **not** uninstall any device that shows “Yes” under Connected, nor any non-0451 devices (Logitech, Silicon Labs, CH340, root hubs, etc.).

4.2 Device Manager – Hidden Devices

Open Device Manager. Enable View → Show hidden devices.

1. Under Ports (COM & LPT) uninstall any remaining grayed-out AR-DevPack-EVM-012 entries. When the dialog appears, check “Delete the driver software for this device” if offered.
2. Check Other devices, Universal Serial Bus controllers, and Human Interface Devices for any remaining grayed-out entries that mention AR-DevPack, XDS110, USB Serial Converter, or VID 0451. Uninstall them the same way.
3. Leave legitimate devices alone (Intel host controllers, USB Root Hubs, Logitech camera, mass-storage devices, Silicon Labs, CH340, etc.).

4.3 COM Name Arbiter Registry Reset

(OPTIONAL) Residual COM port numbers remain reserved even after the devices themselves are removed. Reset the allocation bitmap.

1. Open Registry Editor and navigate to:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\COM Name Arbiter
2. **Backup first:** right-click the COM Name Arbiter key → Export → save the file.
3. Double-click the binary value named **ComDB**. Select all existing data and replace it with zeros (or clear the value).
4. Optionally, under the Devices subkey, delete any remaining REG_SZ values that still contain vid_0451. I deleted them all. But I knew I could reassign other device that had COM ports to correct them in applications. I wanted a fresh list of COM ports.
5. Close Registry Editor.

4.4 First Reboot (Board Disconnected)

Reboot the computer with the IWRL6432 board completely disconnected. After Windows starts, open Device Manager (Show hidden devices) and confirm that no residual AR-DevPack or 0451 entries have reappeared. Optionally re-assign Intel AMT SOL to a free high or low port (e.g., COM2) if it conflicts with other devices.

5. Installing the Correct Drivers

TI's documentation (SWRU620 and related guides) still points users to:

```
C:\ti\mmwave_sdk_<version>\tools\ftdi
```

In current releases of the MMWAVE_L_SDK (including 05_05_04_02) that folder does not exist. Generic FTDI drivers downloaded from ftdichip.com do not contain the TI VID/PID entries and therefore fail with “No driver found.”

When users report the missing folder on the TI E2E forum, TI support currently replies: “This is unusual. Can you try installing mmWave Studio? We have seen in the past this solves the issue.” Multiple users confirm that installing mmWave Studio installs the correct customized drivers.

5.1 Install mmWave Studio 3G

(NOTE: This step WOULD NOT be required if the tools\ftdi were provided where doc said they would be.)

Download and install:

MMWAVE-STUDIO-3G version 04.01.00.06

Direct page: <https://www.ti.com/tool/download/MMWAVE-STUDIO-3G/04.01.00.06>

Note: The download requires a quick TI export-approval form (normal for these tools). After installation the correct TI-signed FTDI drivers are present on the system.

5.2 Verify Driver Installation

Connect only the FTDI USB port of the IWR6432. The four ports should appear under Ports (COM & LPT) as:

- AR-DevPack-EVM-012 (COMx)
- AR-DevPack-EVM-012 (COMy)
- AR-DevPack-EVM-012 (COMz)
- AR-DevPack-EVM-012 (COMw)

Expected driver properties:

- Driver Provider: FTDI
- Driver Version: 2.12.28.0 (dated 2017)
- Digitally signed by Texas Instruments

Do not replace these drivers with the latest generic FTDI package from ftdichip.com. The generic drivers lack the TI VID/PID matching entries; installing them will rename the ports to generic “USB Serial Port” and usually or could break TI Visualizer compatibility.

6. Results Achieved

- All residual 0451 / AR-DevPack / XDS110 phantom devices removed.
- COM Name Arbiter reservation database reset.
- Correct TI-signed FTDI drivers installed and enumerating four healthy AR-DevPack-EVM-012 COM ports.
- System returned to a clean state ready for TI Visualizer testing.

7. Lessons Learned

1. **Documentation lag is real.** TI’s user guide still points to a tools\ftdi folder that no longer ships in the L SDK. When the documented path is missing, the practical solution TI support currently gives is to install an old version of mmWave Studio.

1. **Hacking around the gap creates worse problems.** Installing generic FTDI packages, hand-editing INF files, or forcing mismatched drivers leaves residual registry and device-database entries that are harder to clean later.
2. **Always identify the true Vendor ID first.** On TI boards the silicon may be FTDI but the VID is 0451. Cleanup tools and uninstallers must be aimed at the VID the hardware actually presents.
3. **Clean the residuals thoroughly before installing new drivers.** USBDeview + Device Manager hidden devices + COM Name Arbiter reset form a reliable three-layer cleanup.

8. Note on the XDS110 USB Port

The second Micro USB port on the IWRL6432 is labeled XDS110 USB. It requires DIP-switch changes and was never used during this cleanup. Residual entries for it (VID 0451, PID bef3) were safely removed. When the port is later needed, Windows will create a fresh instance; the standard TI XDS110 drivers supplied with Code Composer Studio / UniFlash normally handle it.

9. Closing Remarks

The sequence above restored a clean USB device database and installed the drivers that correctly match the TI-programmed FTDI silicon on the IWRL6432 board. The same layered approach (USBDeview → Device Manager hidden devices → COM Name Arbiter → correct vendor-supplied package) can be applied to other residual USB conflicts where a vendor has re-programmed FTDI (or similar) silicon under its own VID.

Hacking does not scale. Following the supported path—once the correct path is identified—does.

Document version: 1.0 | Based on a complete cleanup performed 24 August 2026 on system WS01 | Author retains rights to the troubleshooting narrative and conclusions.