Qualcomm Technologies International, Ltd.

Confidential and Proprietary – Qualcomm Technologies International, Ltd.

(formerly known as Cambridge Silicon Radio Ltd.)

Qualcomm Technologies International, Ltd.
Churchill House
Cambridge Business Park
Cambridge, CB4 0WZ
United Kingdom

CSR

Push every boundary®

BlueCore®

USB Device Driver

User Guide

Issue 4

Document History

Revision	Date	History
1	21 JAN 11	Original publication of this document.
2	27 APR 11	Added new section Certifying the USB Driver
3	03 AUG 11	Added a new FAQ section
4	30 APR 15	Updated to latest CSR style

Contacts

General information	www.csr.com
Information on this product	sales@csr.com
Customer support for this product	www.csrsupport.com
More detail on compliance and standards	product.compliance@csr.com
Help with this document	comments@csr.com

Trademarks, Patents and Licences

Unless otherwise stated, words and logos marked with ™ or ® are trademarks registered or owned by CSR plc and/or its affiliates.

Bluetooth® and the Bluetooth logos are trademarks owned by Bluetooth SIG, Inc. and licensed to CSR.

Other products, services and names used in this document may have been trademarked by their respective owners.

The publication of this information does not imply that any licence is granted under any patent or other rights owned by CSR plc or its affiliates.

CSR reserves the right to make technical changes to its products as part of its development programme.

While every care has been taken to ensure the accuracy of the contents of this document, CSR cannot accept responsibility for any errors.

Life Support Policy and Use in Safety-critical Compliance

CSR's products are not authorised for use in life-support or safety-critical applications. Use in such applications is done at the sole discretion of the customer. CSR will not warrant the use of its devices in such applications.

Performance and Conformance

Refer to www.csrsupport.com for compliance and conformance to standards information.

Contents

Tables, Figures and Equations

1. Introduction

This user guide describes how to install and use the CSR USB driver to communicate with a CSR BlueCore device. This enables using the USB for various BlueSuite tools and the Device Firmware Upgrade (DFU) Wizard.

1.1. Operating System Support

Use this procedure with Microsoft Windows XP, Microsoft Vista or Microsoft Windows 7. Both 32-bit and 64-bit versions of the operating systems are supported.

1.2. Installing BlueSuite

If they are not already installed, download and install the latest version of BlueSuite from the **PC Software/Tools** section of www.csrsupport.com.

Note:

Installing the USB driver without BlueSuite is possible, but is outside the scope of this document.

1.3. Driver Signing

During the driver upgrade procedure (Windows XP Driver Procedure in section 3.1; Windows 7 and Vista Driver Procedure in section 3.2), Windows warns that the driver is unsigned if:

- The driver is from BlueSuite 2.3, or earlier
- The driver `.inf` information file is edited, see section 2

Note:

On 64-bit systems, you cannot install a driver unless it is signed. You can temporarily allow an unsigned driver to be installed by pressing **F8** on start-up and setting **Disable Driver Signing**. However, this is a driver development issue and beyond the scope of this document.

2. Modifying the CSR USB Driver Information File

The USB Vendor ID (VID) and Product ID (PID) must exist in the USB Driver information file (`.inf`). This section describes:

- How to check the file to determine if the device is supported
- How to add the device, if required

Note:

If the information file is changed, then Windows warns that the driver is no longer signed.

2.1. Verifying VID and PID Operating System Support

The CSR USB INF file is `CSRBlueCoreUSB.inf`, and is located by default in **Program Files\CSR\BlueSuite\drivers**.

1. Open the file with a text editor.
2. Search for the device VID and PID. This example is for a standard Dell Bluetooth device:

```
%DELL.DeviceDesc%=CSRBC.Inst.NTx86.5.1,
       USB\VID_413c&PID_8000 ; DELL USB Device VID&PID
```

If the device is supported, ignore section 2.2.

2.2. Adding the VID and PID

Create a backup copy of `CSRBlueCoreUSB.inf` before editing the file.

The example in this section adds the following device:

- VID = `0x1234`
- PID = `0x5678` (All PIDs and VIDs in an INF file are in hexadecimal.)
- Name = Generic Device.

Although you can only modify your particular operating system, CSR recommends that you add the information for all operating systems, for completeness.

The new lines added to the file are shown in **bold**. CSR recommends the device information is added after the final line in each section:

1. Add the device to the Windows 2000 section of the file:

```
;
; Windows 2000
;
[CSR]
...
%MOTION2DFU.DeviceDesc%=CSRBC.Inst,
      USB\VID_10ab&PID_1006 ; MOTION BC04 Device VID&PID DFU
%GENERIC.DeviceDesc%=CSRBC.Inst,
      USB\VID_1234&PID_5678 ; Generic Device VID&PID
```

2. Add the device to the 32-bit Windows XP and later section of the file:

```
;
; Windows XP and later
;
[CSR.NTx86.5.1]
...
%MOTION2DFU.DeviceDesc%=CSRBC.Inst.NTx86.5.1,
      USB\VID_10ab&PID_1006 ; MOTION BC04 Device VID&PID DFU
%GENERIC.DeviceDesc%=CSRBC.Inst.NTx86.5.1,
      USB\VID_1234&PID_5678 ; Generic Device VID&PID
```

3. Add the device to the 64-bit Windows XP and later section of the file:

```
;
; 64-bit Windows XP and later
;
[CSR.NTamd64...1]
...
%MOTION2DFU.DeviceDesc%=CSRBC.Inst.NTamd64...1,
USB\VID_10ab&PID_1006 ; MOTION BC04 Device VID&PID DFU
%GENERIC.DeviceDesc%=CSRBC.Inst.NTamd64...1,
USB\VID_1234&PID_5678 ; Generic Device VID&PID
```

4. Add the device name to the **strings** section of the file:

```
[Strings]
CSR="Cambridge Silicon Radio"
MfgName="CSR"
...
MOTION2DFU.DeviceDesc="MOTION BC04 in DFU - CSR Driver"
GENERIC.DeviceDesc="Generic Device - CSR Driver"
```

5. After adding the appropriate lines, save the file and exit the editor.

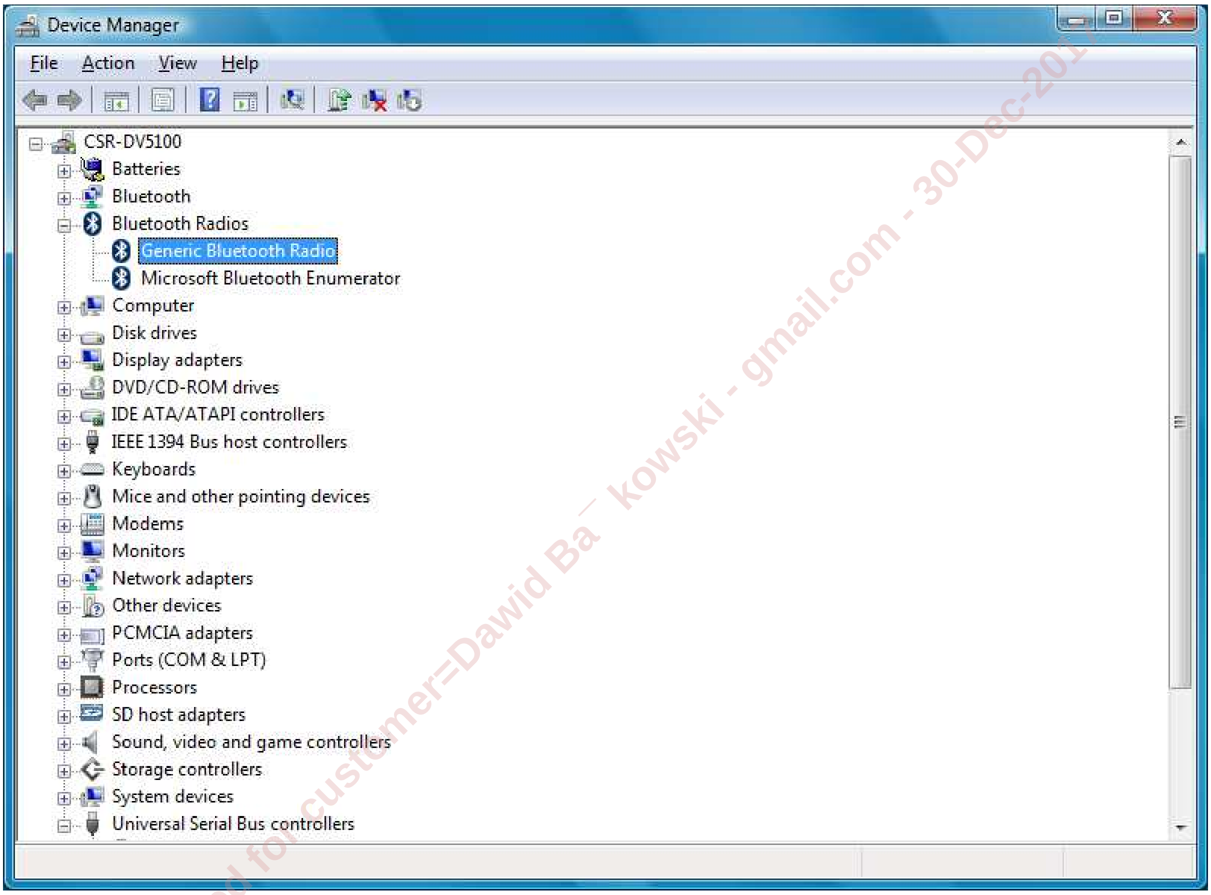
3. Loading the CSR USB Driver

To start the device manager:

1. Go to **Start/Run…**
2. Enter **devmgmt.msc**
3. Press **OK**.

Depending on the device VID and PID, it can appear in one of several categories in the Device Manager. Figure 3.1 to Figure 3.2 show some common locations for the device.

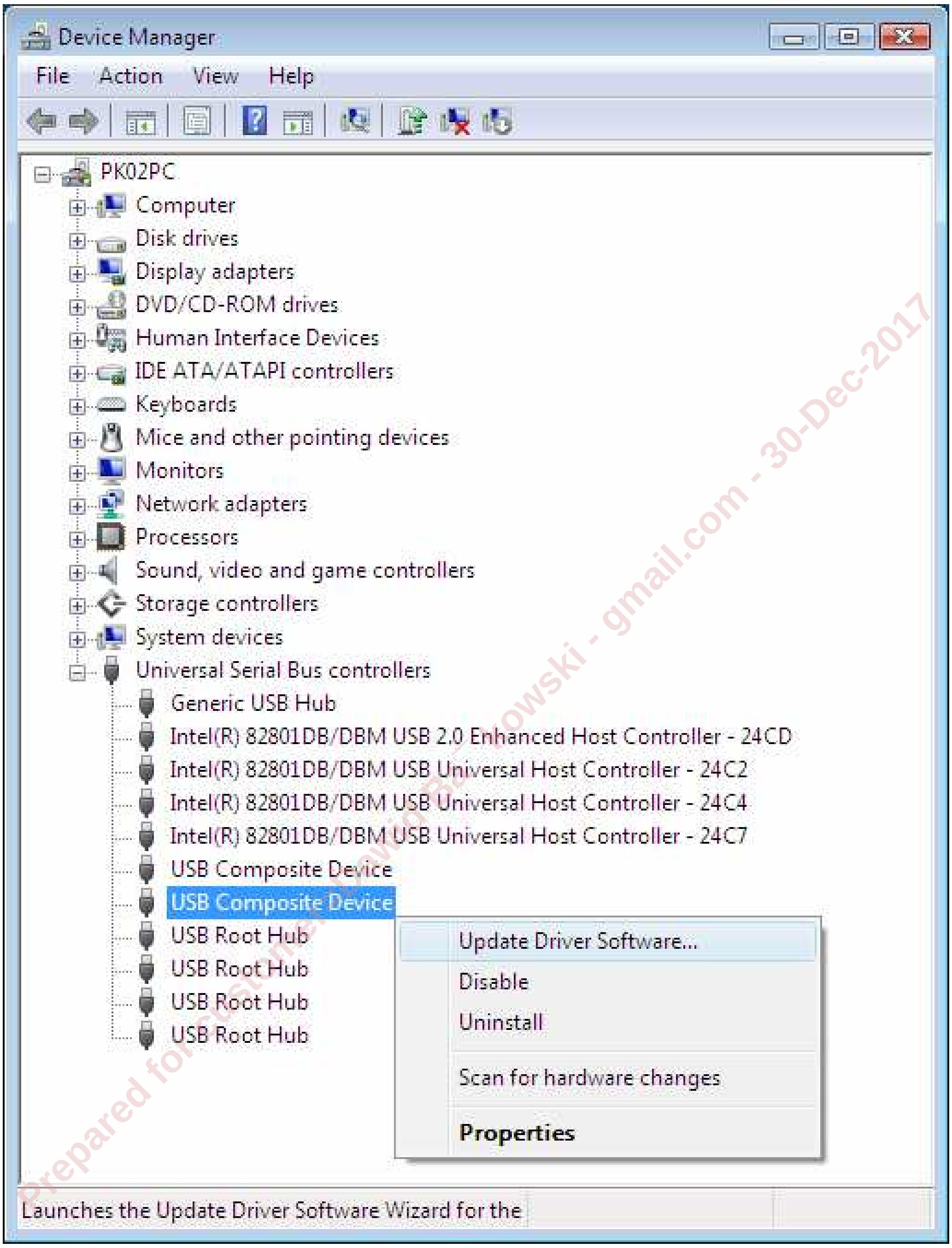
Figure 3.1 shows the device as a **Generic Bluetooth Radio**.



- CSR-DV5100
  - Batteries
  - Bluetooth
  - Bluetooth Radios
    - Generic Bluetooth Radio
    - Microsoft Bluetooth Enumerator
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - IDE ATA/ATAPI controllers
  - IEEE 1394 Bus host controllers
  - Keyboards
  - Mice and other pointing devices
  - Modems
  - Monitors
  - Network adapters
  - Other devices
  - PCMCIA adapters
  - Ports (COM & LPT)
  - Processors
  - SD host adapters
  - Sound, video and game controllers
  - Storage controllers
  - System devices
  - Universal Serial Bus controllers

Figure 3.1: Generic Bluetooth Radio in Bluetooth Radios Section

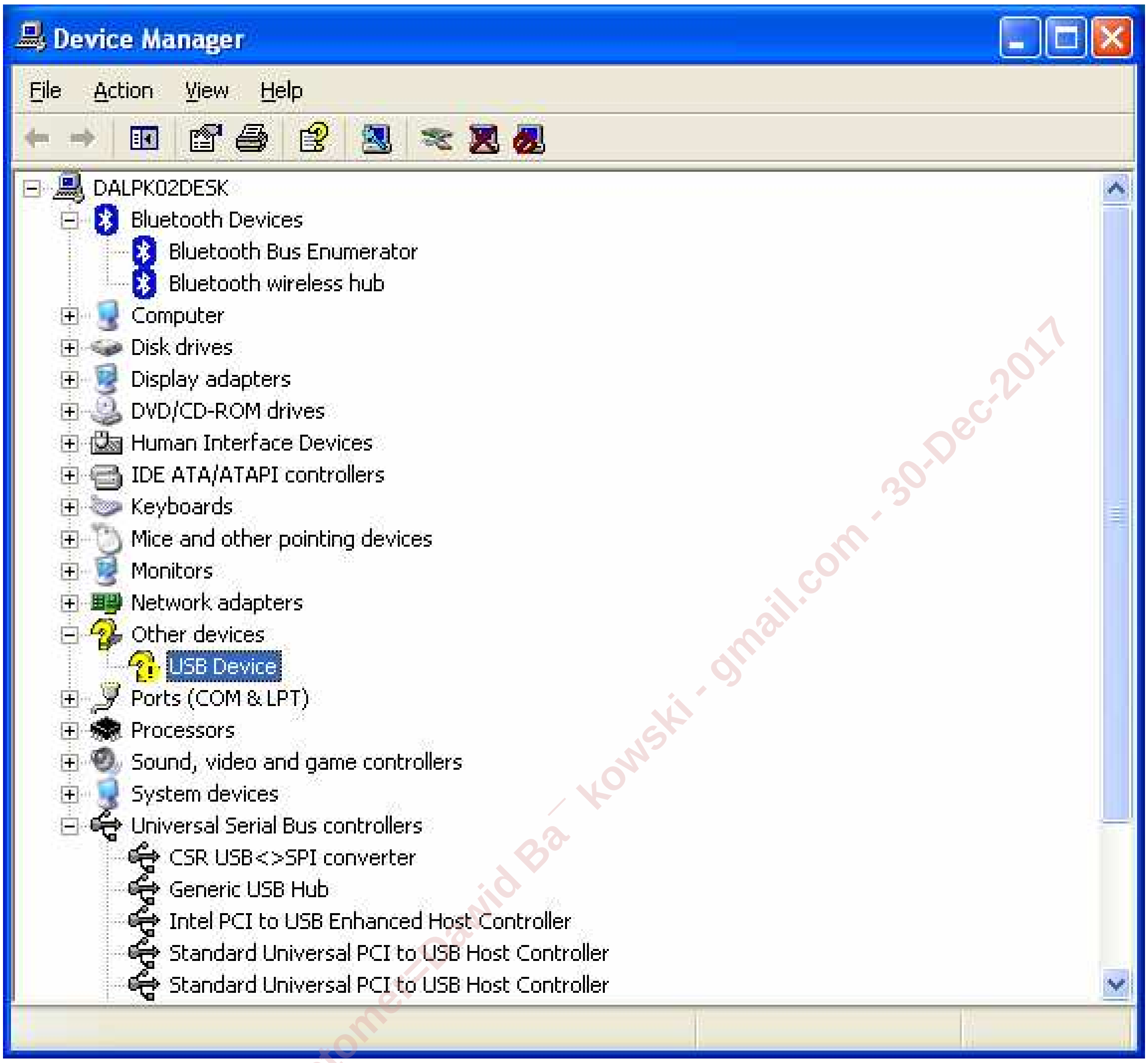
Figure 3.2 shows the device as a **USB Composite Device**.



Device Manager
File Action View Help
PK02PC
Computer
Disk drives
Display adapters
DVD/CD-ROM drives
Human Interface Devices
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Processors
Sound, video and game controllers
Storage controllers
System devices
Universal Serial Bus controllers
Generic USB Hub
Intel(R) 82801DB/DBM USB 2.0 Enhanced Host Controller - 24CD
Intel(R) 82801DB/DBM USB Universal Host Controller - 24C2
Intel(R) 82801DB/DBM USB Universal Host Controller - 24C4
Intel(R) 82801DB/DBM USB Universal Host Controller - 24C7
USB Composite Device
USB Composite Device
USB Root Hub
USB Root Hub
USB Root Hub
USB Root Hub
Update Driver Software...
Disable
Uninstall
Scan for hardware changes
Properties
Launches the Update Driver Software Wizard for the

Figure 3.2: USB Composite Device in Universal Serial Bus Controllers Section

Figure 3.3 shows the device as a **USB Device** in the **Other devices** category.



```
DALPK02DESK
    Bluetooth Devices
        Bluetooth Bus Enumerator
        Bluetooth wireless hub
    Computer
    Disk drives
    Display adapters
    DVD/CD-ROM drives
    Human Interface Devices
    IDE ATA/ATAPI controllers
    Keyboards
    Mice and other pointing devices
    Monitors
    Network adapters
    Other devices
        USB Device
    Ports (COM & LPT)
    Processors
    Sound, video and game controllers
    System devices
    Universal Serial Bus controllers
        CSR USB<>SPI converter
        Generic USB Hub
        Intel PCI to USB Enhanced Host Controller
        Standard Universal PCI to USB Host Controller
        Standard Universal PCI to USB Host Controller
```

Figure 3.3: USB Device in Other Devices Section

Right-click on the device, and select **Update Driver…**

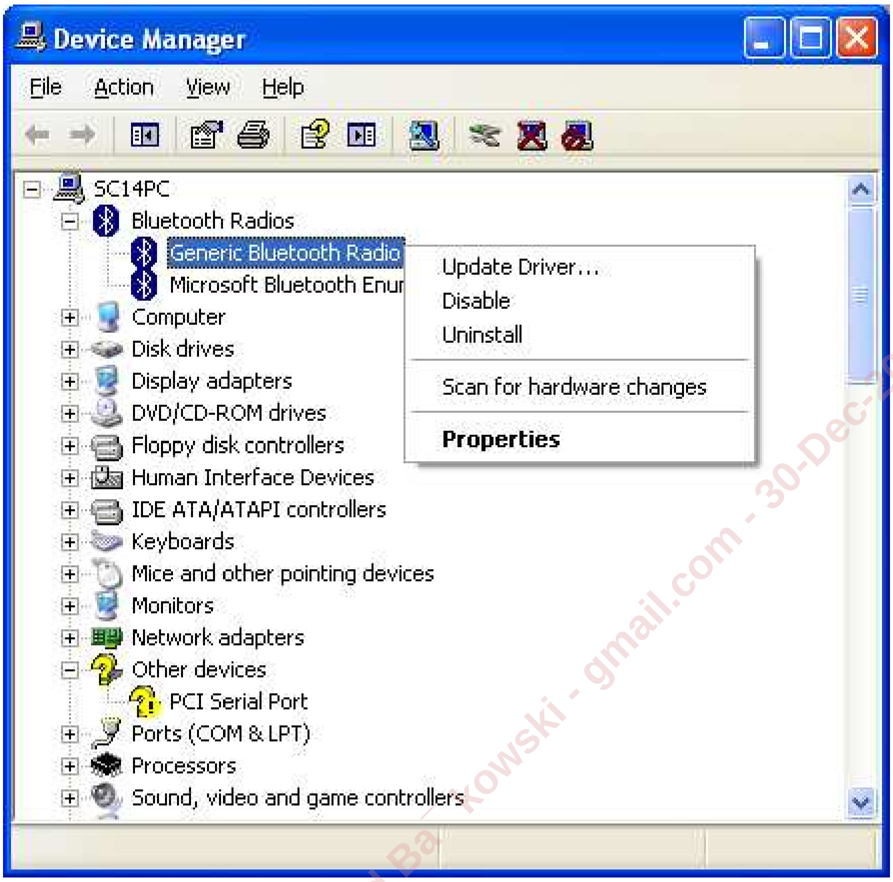


Figure 3.4: Update Driver Software for Generic Bluetooth Radio

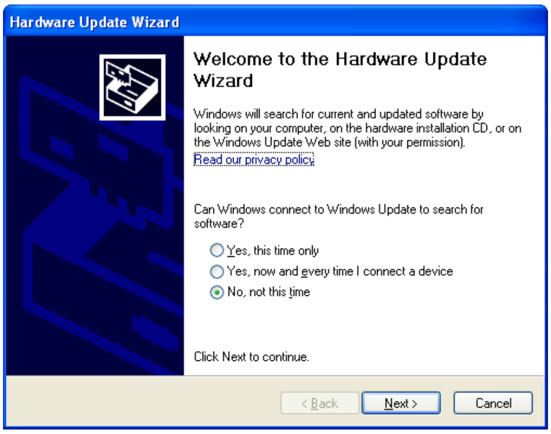
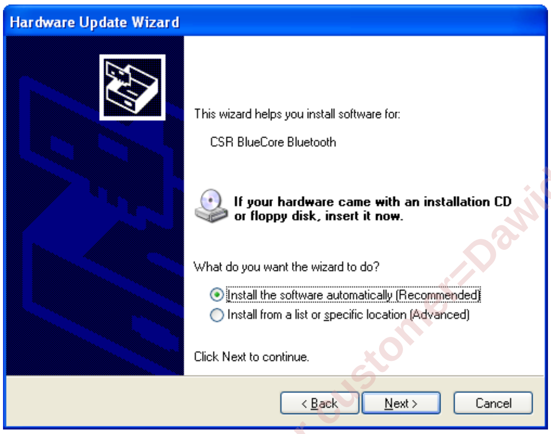
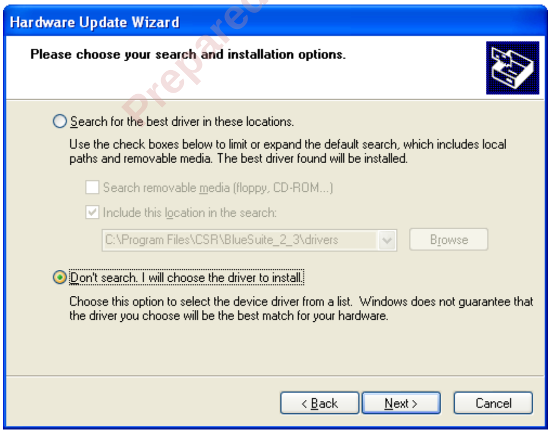
For Microsoft Windows XP, follow the procedure in section 3.1.

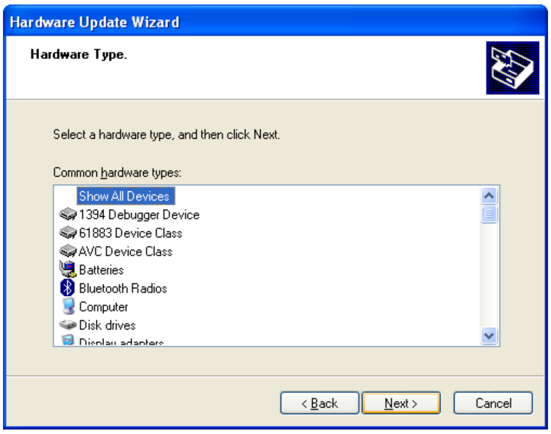
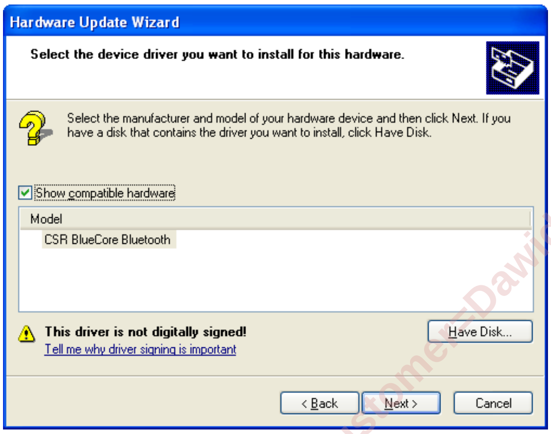
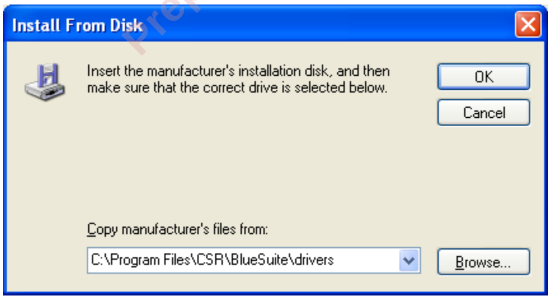
For Microsoft Vista and Windows 7, follow the procedure in section 3.2.

BlueCore USB Device Driver User Guide

3.1. Windows XP Driver Procedure

This section describes how to update the driver using the Hardware Update Wizard in Windows XP.

Prompt	Required Actions
Can Windows connect to Windows Update to search for software  Hardware Update Wizard Welcome to the Hardware Update Wizard Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission). Read our privacy policy Can Windows connect to Windows Update to search for software? ○ Yes, this time only ○ Yes, now and every time I connect a device ● No, not this time Click Next to continue. 	- Select **No, not this time** - Click **Next**
What do you want the wizard to do?  Hardware Update Wizard This wizard helps you install software for: CSR BlueCore Bluetooth **If your hardware came with an installation CD or floppy disk, insert it now.** What do you want the wizard to do? ● Install the software automatically (Recommended) ○ Install from a list or specific location (Advanced) Click Next to continue. 	- Select **Install from a list or specific location (Advanced)** - Click **Next**
Please choose your search and installation options  Hardware Update Wizard **Please choose your search and installation options.** ○ Search for the best driver in these locations. Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed. ☐ Search removable media (floppy, CD-ROM...) ☑ Include this location in the search: C:\Program Files\CSR\BlueSuite_2_3\drivers ● Don't search. I will choose the driver to install. Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware. 	Select **Don't search, I will choose the driver to install** Click **Next**

Prompt	Required Actions
Note: This window is not always displayed. **Select a hardware type and then click Next**  Hardware Update Wizard **Hardware Type.** Select a hardware type, and then click Next. Common hardware types: Show All Devices 1394 Debugger Device 61883 Device Class AVC Device Class Batteries Bluetooth Radios Computer Disk drives Display adapters 	▪ Click **Next**
 Hardware Update Wizard **Select the device driver you want to install for this hardware.** Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk. ☑ Show compatible hardware Model CSR BlueCore Bluetooth **This driver is not digitally signed!** Tell me why driver signing is important **Note:** A warning is only displayed if the driver has been changed or is unsigned.	▪ Click **Have Disk**
Insert the manufacturer's installation disk…  Install From Disk Insert the manufacturer's installation disk, and then make sure that the correct drive is selected below. Copy manufacturer's files from: C:\Program Files\CSR\BlueSuite\drivers 	▪ Click the **Browse…** button, and select the file `CSRBlueCoreUSB.inf`. By default, the file is located in: **Program Files\CSR\BlueSuite\drivers**

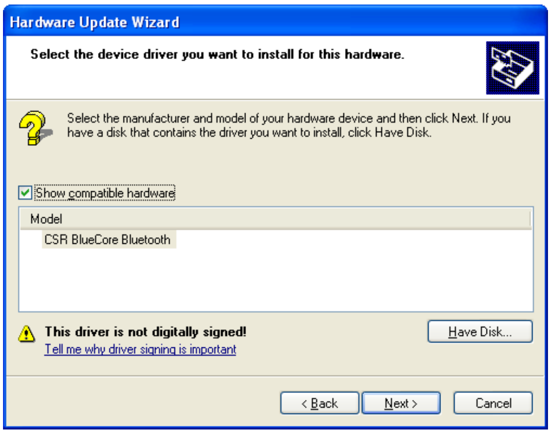
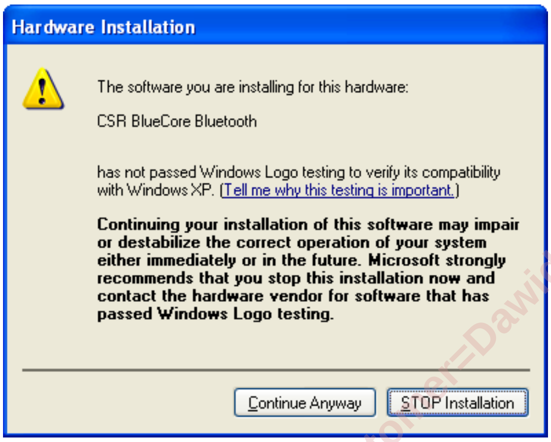
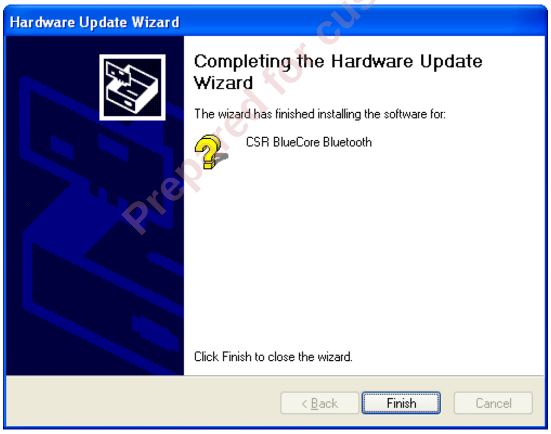
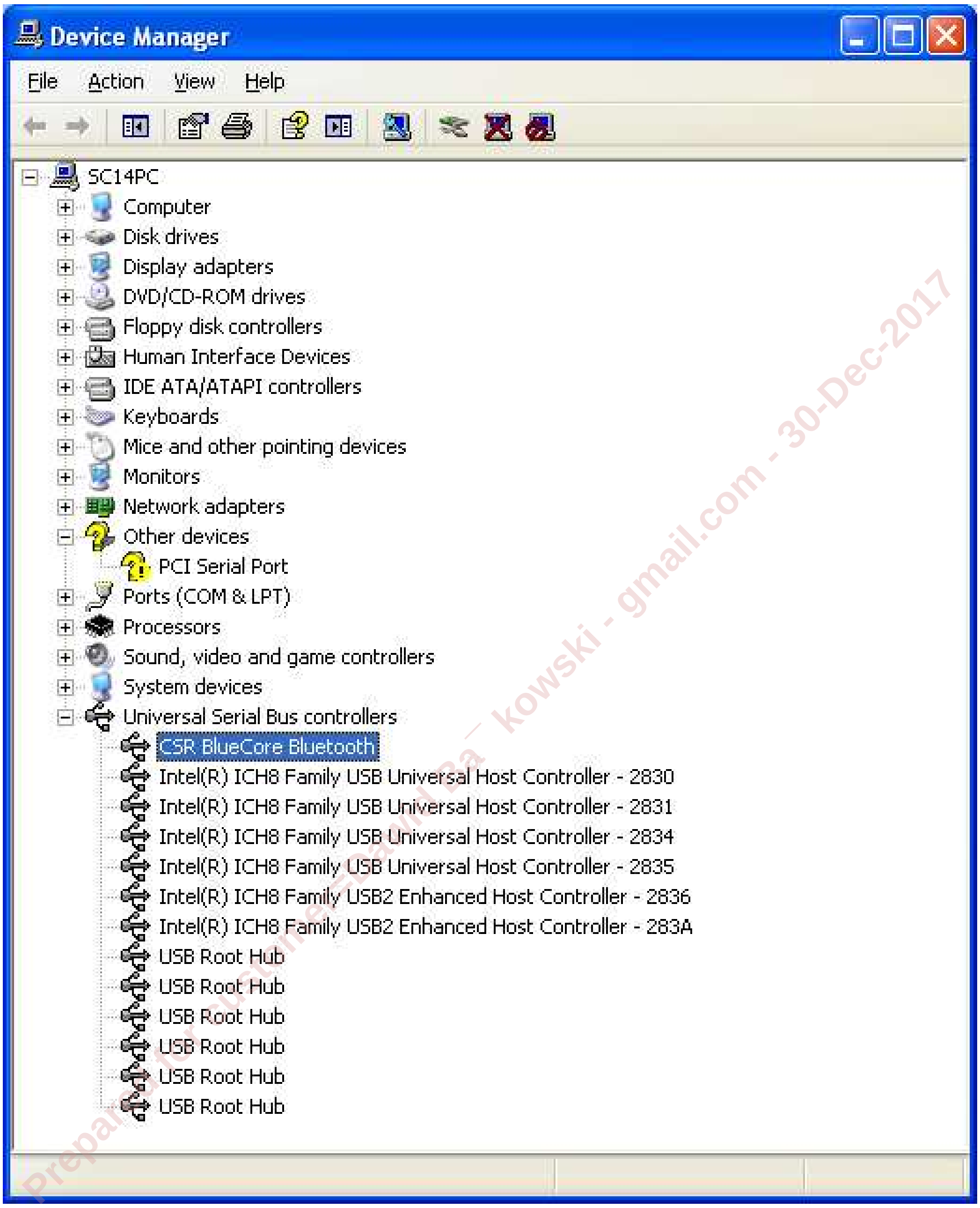
Prompt	Required Actions
 Hardware Update Wizard Select the device driver you want to install for this hardware. Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk. ☑ Show compatible hardware Model CSR BlueCore Bluetooth This driver is not digitally signed! Tell me why driver signing is important Have Disk... < Back Next > Cancel	- Click **Next**
Note: This warning is only displayed if the driver has been changed or is unsigned.  Hardware Installation The software you are installing for this hardware: CSR BlueCore Bluetooth has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.) **Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.** Continue Anyway STOP Installation	- Click **Continue Anyway**
 Hardware Update Wizard Completing the Hardware Update Wizard The wizard has finished installing the software for: CSR BlueCore Bluetooth Click Finish to close the wizard. < Back Finish Cancel	- Click **Finish**

Table 3.1: Windows XP Procedure

After loading the CSR device driver, the Bluetooth device is located in the **Universal Serial Bus controllers** section of the **Device Manager**, see Figure 3.5.



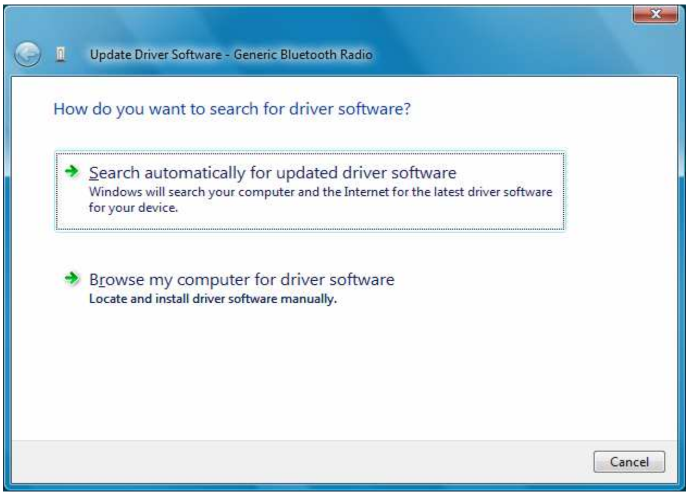
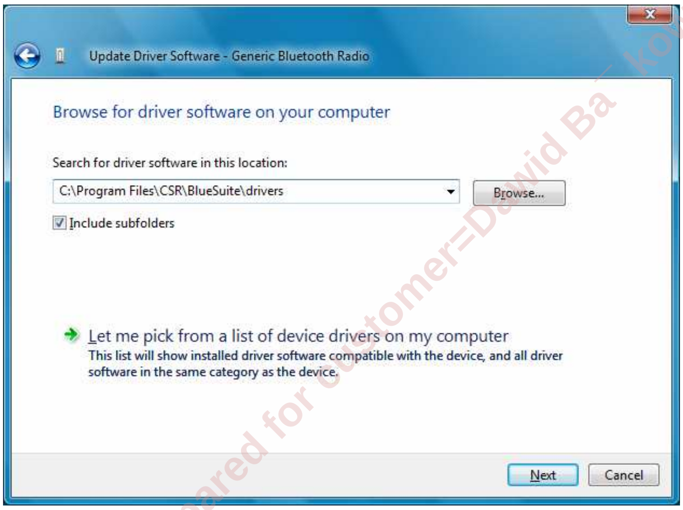
- SC14PC
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - Floppy disk controllers
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Other devices
    - PCI Serial Port
  - Ports (COM & LPT)
  - Processors
  - Sound, video and game controllers
  - System devices
  - Universal Serial Bus controllers
    - CSR BlueCore Bluetooth
    - Intel(R) ICH8 Family USB Universal Host Controller - 2830
    - Intel(R) ICH8 Family USB Universal Host Controller - 2831
    - Intel(R) ICH8 Family USB Universal Host Controller - 2834
    - Intel(R) ICH8 Family USB Universal Host Controller - 2835
    - Intel(R) ICH8 Family USB2 Enhanced Host Controller - 2836
    - Intel(R) ICH8 Family USB2 Enhanced Host Controller - 283A
    - USB Root Hub
    - USB Root Hub
    - USB Root Hub
    - USB Root Hub
    - USB Root Hub
    - USB Root Hub

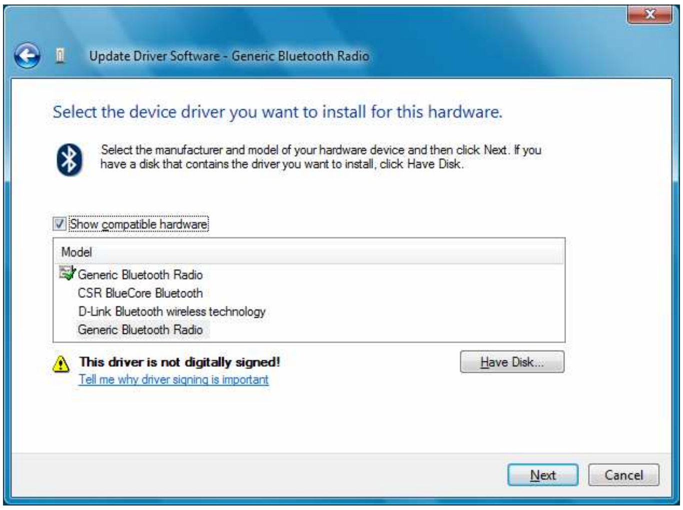
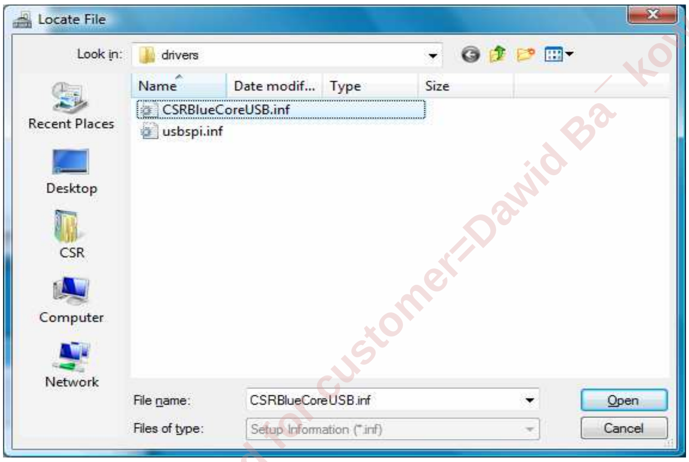
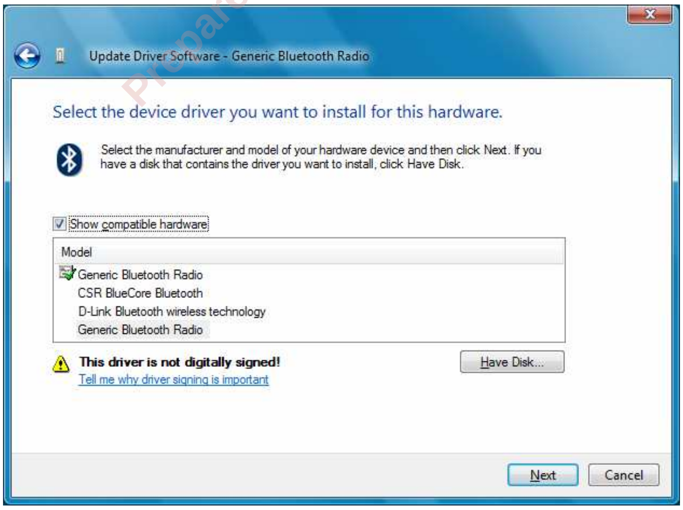
Figure 3.5: Device Manager Showing Generic Device CSR Driver

The BlueSuite tools and the DFU Wizard can now communicate with the device using USB.

3.2. Windows 7 and Vista Driver Procedure

This section describes how to update the driver in Windows 7 and Vista.

Prompt	Required Actions
How do you want to search for driver software?  Update Driver Software - Generic Bluetooth Radio How do you want to search for driver software? Search automatically for updated driver software Windows will search your computer and the Internet for the latest driver software for your device. Browse my computer for driver software Locate and install driver software manually. 	- Select **Browse my computer for driver software**
Browse for driver software on your computer  Update Driver Software - Generic Bluetooth Radio Browse for driver software on your computer Search for driver software in this location: C:\Program Files\CSR\BlueSuite\drivers ☑ Include subfolders Let me pick from a list of device drivers on my computer This list will show installed driver software compatible with the device, and all driver software in the same category as the device. 	- Select **Let me pick from a list of device drivers on my computer** - Click **Next**

Prompt	Required Actions
Select the device driver you want to install for this hardware **Note:** A warning is displayed if the driver has been changed or is unsigned.  Update Driver Software - Generic Bluetooth Radio Select the device driver you want to install for this hardware. Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk. ☑ Show compatible hardware Model Generic Bluetooth Radio CSR BlueCore Bluetooth D-Link Bluetooth wireless technology Generic Bluetooth Radio This driver is not digitally signed! Tell me why driver signing is important 	- Click **Have Disk**
 Locate File Look in: drivers Name | Date modif... | Type | Size CSRBlueCoreUSB.inf usbspi.inf File name: CSRBlueCoreUSB.inf Files of type: Setup Information (*.inf) Recent Places Desktop CSR Computer Network Open Cancel	- Click the **Browse…** button, and select the file `CSRBlueCoreUSB.inf`. By default, the file is located in: **Program Files\CSR\BlueSuite\drivers**
 Update Driver Software - Generic Bluetooth Radio Select the device driver you want to install for this hardware. Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk. ☑ Show compatible hardware Model Generic Bluetooth Radio CSR BlueCore Bluetooth D-Link Bluetooth wireless technology Generic Bluetooth Radio This driver is not digitally signed! Tell me why driver signing is important 	- Click **Next**

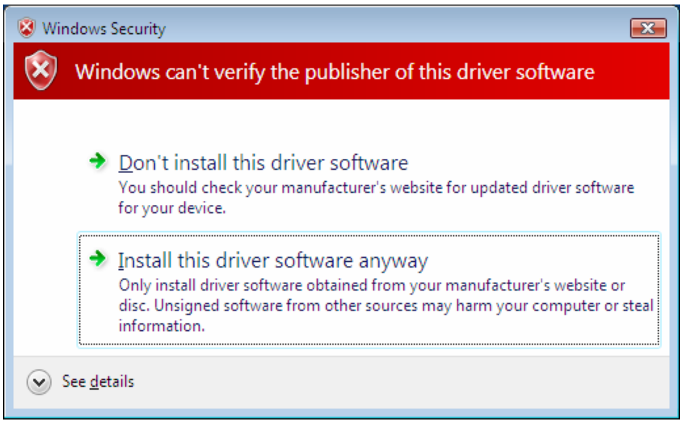
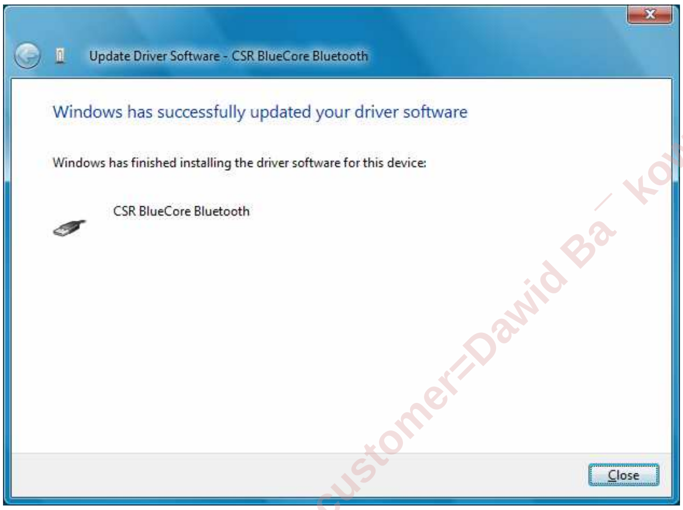
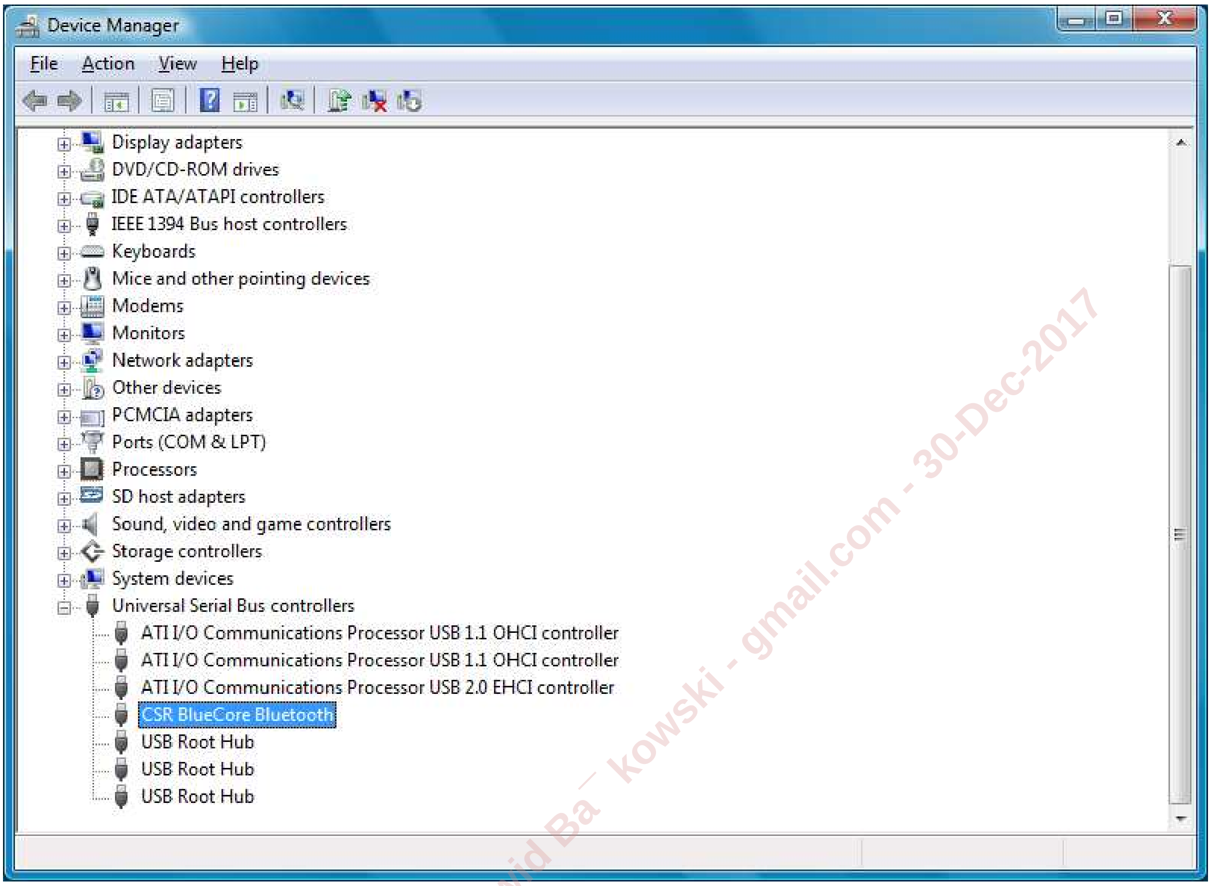
Prompt	Required Actions
Note: This warning is only displayed if the driver has been changed or is unsigned.  Windows Security Windows can't verify the publisher of this driver software Don't install this driver software You should check your manufacturer's website for updated driver software for your device. Install this driver software anyway Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information. See details	- Select **Install this driver software anyway**
 Update Driver Software - CSR BlueCore Bluetooth Windows has successfully updated your driver software Windows has finished installing the driver software for this device: CSR BlueCore Bluetooth Close	- Click **Close**

Table 3.2: Windows 7 and Vista Driver Procedure

After loading the CSR device driver, the Bluetooth device is located in the **Universal Serial Bus controllers** section of the **Device Manager**.



- Display adapters
- DVD/CD-ROM drives
- IDE ATA/ATAPI controllers
- IEEE 1394 Bus host controllers
- Keyboards
- Mice and other pointing devices
- Modems
- Monitors
- Network adapters
- Other devices
- PCMCIA adapters
- Ports (COM & LPT)
- Processors
- SD host adapters
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers
  - ATI I/O Communications Processor USB 1.1 OHCI controller
  - ATI I/O Communications Processor USB 1.1 OHCI controller
  - ATI I/O Communications Processor USB 2.0 EHCI controller
  - CSR BlueCore Bluetooth
  - USB Root Hub
  - USB Root Hub
  - USB Root Hub

Figure 3.6: Device Manager Showing CSR BlueCore Device

The BlueSuite tools and the DFU Wizard can now use the device using USB.

Rolling Back the Device Driver

After using BlueSuite or the DFU Wizard to upgrade firmware, you can restore the original device driver for the device. The easiest way to do this is by rolling back to the previous version of the driver.

Right-click **CSR BlueCore Bluetooth** device, and select **Properties**.

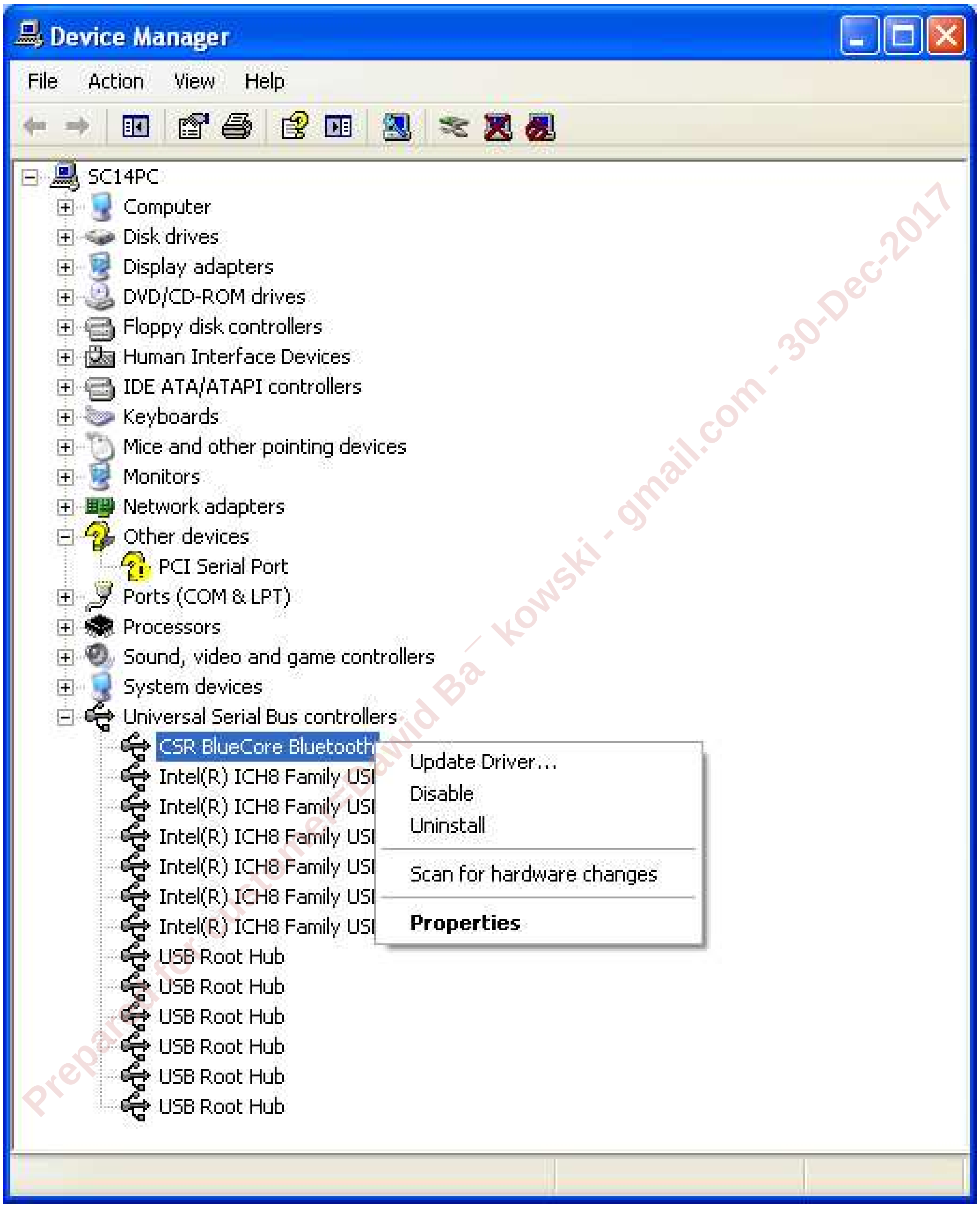


Figure 3.7: Device Manager Properties Popup

Note:

If there are no backed up drivers for this device (such as when the device is using drivers native to Microsoft Windows or when the drivers have never been updated), the system displays a dialogue box that informs you that no driver files have been backed up for this device.

Prompt	Required Actions
CSR BlueCore Bluetooth Properties	- In the **Driver** tab, select **Roll Back Driver** - Click **OK**
Are you sure you would like to roll back to the previously installed driver software	- Click **Yes**

Table 3.3: Rollback Procedure

Following the driver rollback, the Bluetooth device appears in its original section of the **Device Manager**. **Error! Reference source not found.** shows the device as a **Generic Bluetooth Radio** in the **Bluetooth Radios** section.

4. Certifying the USB Driver

This section gives basic information and references on how to certify a USB driver using the Windows logo program. When a change is made in any component of an already certified driver, e.g. changing or adding a VID/PID in the `.inf` file, the driver needs to be re-certified.

4.1. Windows Logo Program

The Windows logo signifies the compatibility and reliability of systems and devices with the Windows operating system. It gives customers confidence that a product is thoroughly tested with Microsoft-provided tools.

Microsoft has a Windows logo program. The details of the program, plus the steps to get drivers signed, are on the MSDN link:

http://msdn.microsoft.com/en-us/windows/hardware/gg487360

The Windows Hardware Quality Labs (WHQL) testing programs provided by Microsoft help ensure that your hardware and drivers qualify for the Windows Logo Program. The tools provided are described on:

http://msdn.microsoft.com/en-us/windows/hardware/gg487530

4.2. WHQL Testing and Submission

4.2.1. Certification Requirements

To certify the driver you require:

1. A Winqual account and a submission tool to submit the driver for certification. There are fees for some submission types.

   For more details, see the Sign Up page on https://winqual.microsoft.com
2. VeriSign Microsoft Authenticode Code Signing Digital Certificate available from https://www.verisign.com

   Occasionally, there are announcements in Winqual about VeriSign offering a “Microsoft certificate” at a substantially reduced price.
3. Windows Logo Kit downloadable from http://msdn.microsoft.com/en-us/windows/hardware/gg487530
4. The following PC and OS environments are required:
   - 64-bit machine with Windows server 2008 R2 OS, which is the DTM (Driver Test Manager)
   - At least one 32-bit and one 64-bit PC with Windows 7 OS to execute the WHQL tests
5. Device Under Test (DUT).
6. Driver files (sys file, `.cat` catalogue file, `.inf` file) signed with VeriSign Microsoft Authenticode Digital Certificate.

4.2.2. Driver Submission and Certification Process

To submit and certify a driver:

1. Set up the DTM machine and test machines as described in:

   http://msdn.microsoft.com/en-us/library/ff563424(v=vs.85).aspx
2. Run WHQL tests for the device category that your USB device belongs to. For a list of device categories see:

   http://msdn.microsoft.com/en-us/windows/hardware/gg462990.
3. Prepare the submission package files (CPK file).
4. Submit the result files (CPK files) to Winqual, https://winqual.microsoft.com, and get the certified drivers.

Note:

Contact Microsoft support teams with any queries: winqual@microsoft.com and logofb@microsoft.com.

5. Interface Definition

This section describes the interfaces provided by the BlueCore USB device driver. Some examples using Windows API functions are provided, but for full details refer to MSDN.

5.1. Opening the Device

To open a BlueCore USB device, call the `CreateFile` WIN32 API with device name `\\.\CSRx (x=0, 1, 2, 3...)`. This returns a handle to use in subsequent calls to interact with the device. If more than one CSR BlueCore device is on the USB bus, they are named CSR0, CSR1, CSR2, etc. This call fails if there are no devices attached to the USB bus.

Example:

```
handle = CreateFile
   (
         device_name,
         GENERIC_READ | GENERIC_WRITE,
         0,
         0,
         OPEN_EXISTING,
         FILE_FLAG_OVERLAPPED,
         0
   );
```

5.2. Closing the Device

To close the handle use `CloseHandle`.

Example:

```
CloseHandle (handle);
```

5.3. Writing ACL Data

To send ACL Data to the device, use `WriteFile`:

- `data` is a pointer to the data buffer that holds the ACL Data of size length.
- `written` parameter returns the number of bytes actually sent.

Example:

```
   status = WriteFile (handle, data, length, &written, 0);
```

5.4. Reading ACL Data

To read ACL data, use `ReadFile`:

- `buffer` is a pointer to where the data of size length should be copied
- `read` parameter returns the number of bytes actually received

Note:

This is a non-blocking call, so if no bytes are currently available, then the function returns immediately.

Example:

```
Status = ReadFile (handle, buffer, length, &read, 0);
```

5.5. Supported IOCTLs

Some of the driver's interfaces are exposed through Input/output Controls (IOCTLs). These can be called from the application (user mode) using the WIN32 API `DeviceIoControl`. The `DeviceIoControl` takes the IOCTL code as one of the parameters .The IOCTLs defines device-specific operation.

Signature for `DeviceIoControl` commands example:

```
status = DeviceIoControl
              (
                           handle,             // Handle obtained by CreateFile
                           IOCTL_CSRBC_XXX,    // any one of the IOCTL Codes in
the
                                               // table below
                           lpInBuffer,         // Input buffer pointer
                           nInBufferSize,      // Input buffer size
                           lpOutBuffer,        // Output buffer pointer
                           nOutBufferSize,     // Output buffer size
                           lpBytesReturned,    // Number of bytes returned
                           lpOverlapped        // Overlapped for Async IO
             );
```

The return value from `DeviceIoControl` is:

- If the operation completes successfully, the return value is nonzero
- If the operation fails or is pending, the return value is zero. To get extended error information, call `GetLastError()`

Important Note:

The `lpBytesReturned` value can be NULL only when the value of `lpOverlapped` is not NULL. If an operation returns no output data and `lpOutBuffer` is NULL, the value of `lpBytesReturned` can be ignored.

The supported IOCTLs are listed in Table 5.1.

Section	IOCTL	Description
5.5.1	`IOCTL_CSRBC_GET_DEVICE_DESCRIPTOR`	Gets USB device descriptor
5.5.2	`IOCTL_CSRBC_GET_CONFIG_DESCRIPTOR`	Gets USB configuration descriptor
5.5.3	`IOCTL_CSRBC_RESET_DEVICE`	Resets the device
5.5.4	`IOCTL_CSRBC_GET_DRIVER_NAME`	Gets driver name
5.5.5	`IOCTL_CSRBC_GET_VERSION`	Gets the USB speed
5.5.6	`IOCTL_CSRBC_SEND_HCI_COMMAND`	Sends HCI command
5.5.7	`IOCTL_CSRBC_GET_HCI_EVENT`	Gets HCI events
5.5.8	`IOCTL_CSRBC_SEND_HCI_DATA`	Sends HCI data
5.5.9	`IOCTL_CSRBC_GET_HCI_DATA`	Gets HCI data
5.5.10	`IOCTL_CSRBC_START_SCO_DATA`	Starts SCO interface
5.5.11	`IOCTL_CSRBC_SEND_SCO_DATA`	Sends SCO data
5.5.12	`IOCTL_CSRBC_RECV_SCO_DATA`	Receives SCO data
5.5.13	`IOCTL_CSRBC_STOP_SCO_DATA`	Stops the opened SCO interface
0	`IOCTL_CSRBC_SEND_CONTROL_TRANSFER`	Sends control information
5.6	`IOCTL_CSRBC_BLOCK_HCI_EVENT`	Blocks till HCI event is received
5.7	`IOCTL_CSRBC_BLOCK_HCI_DATA`	Waits till HCI data is received

Table 5.1: IOCTL Description

5.5.1. Get USB Device Descriptor

Use `IOCTL_CSRBC_GET_DEVICE_DESCRIPTOR`:

- `lpOutBuffer` is a pointer to a buffer
- `nOutBufferSize` is size of the buffer
- `lpBytesReturned` is actual number of bytes read

The `nOutBufferSize` must be greater than or equal to the size of structure, `_USB_DEVICE_DESCRIPTOR` (18 bytes). All the other parameters can be NULL. The USB device descriptor read is available in the `lpOutBuffer`.

5.5.2. Get USB Configuration Descriptor

Use `IOCTL_CSRBC_GET_CONFIG_DESCRIPTOR`:

- `lpOutBuffer` is a pointer to a buffer
- `nOutBufferSize` is size of the buffer
- `lpBytesReturned` is actual number of bytes read

The `nOutBufferSize` must be greater than or equal to the size of structure, `_USB_CONFIGURATION_DESCRIPTOR` (9 bytes).

Note:

For information on `_USB_DEVICE_DESCRIPTOR` and `_USB_CONFIGURATION_DESCRIPTOR`, refer to the USB 2.0 technical specification in www.usb.org.

5.5.3. Reset the Device

Use `IOCTL_CSRBC_RESET_DEVICE` to reset the device. All Parameters other than `handle`, `IOCTL` and `lpBytesReturned` are ignored.

Important Note:

A cold or warm reset sent to the IC causes a surprise removal of the BlueCore device. During this state, if a user-mode application retains its handle to the device driver instance, the driver is unable to create a device instance for the newly connected device and it fails, reporting error Code 31(Device is not working properly because Windows cannot load the drivers required for this device). This is a timing constraint of the Win32 driver model and its occurrence is rare.

5.5.4. Get Driver Name

Use `IOCTL_CSRBC_GET_DRIVER_NAME` to read the device driver name string. The name of the driver is **CSR BlueCore USB Kernel Device Driver**.

Note:

The output buffer (`lpOutBuffer`) passed should be big enough to hold this string. If the size (`nOutBufferSize`) is less, the invalid buffer size error code is returned.The actual number of bytes read is returned in the `lpBytesReturned` parameter.

5.5.5. Get the USB Speed

Use `IOCTL_CSRBC_GET_VERSION` to show if the device supports high speed or full speed. The size of the output buffer (`lpOutBuffer`) needs to be 4 bytes irrespective of the processor being used (either x86 or x64). The driver returns `2` in `lpOutBuffer` if high speed is supported and `1` if it is not supported. The actual number of bytes read is returned in the `lpBytesReturned` parameter.

5.5.6. Send HCI Command

Use `IOCTL_CSRBC_SEND_HCI_COMMAND`:

- `lpInBuffer` is a pointer to input data buffer
- `nInBufferSize` is size of the data to be sent
- `lpBytesReturned:` value returned can be ignored but a valid pointer has to be passed to the driver

5.5.7. Get HCI Event

`Use IOCTL_CSRBC_GET_HCI_EVENT` to know if an HCI Event has been received by the driver:

- `lpOutBuffer` is a pointer to a buffer
- `nOutBufferSize` is size of the buffer
- `lpBytesReturned` is actual number of bytes read

When the driver has received an event, the buffer is filled with the values corresponding to the event and the number of valid bytes read are returned in `lpBytesReturned`.

Note:

This is a non-blocking call, so if there is no event waiting to be sent up to the user application, then it returns immediately with a `value 0` in the `lpBytesReturned` parameter.

5.5.8. Send HCI Data

Use IOCTL_CSRBC_SEND_HCI_DATA:

- lpInBuffer is a pointer to input data buffer
- nInBufferSize is size of the data to be sent
- lpBytesReturned: value returned can be ignored but a valid pointer has to be passed to the driver

Note:

This data is sent through bulk endpoint of device.

5.5.9. Get HCI Data

Use IOCTL_CSRBC_GET_HCI_DATA:

- lpOutBuffer is a pointer to an empty buffer
- nOutBufferSize is size of the buffer
- lpBytesReturned is actual number of bytes read

5.5.10. Start SCO Interface

Use IOCTL_CSRBC_START_SCO_DATA to open Synchronous Connection-Oriented interface:

- lpInBuffer is a pointer to input data buffer. It takes 4 bytes irrespective of processor used (either x86 or x64) and should have the interface number that needs to be selected
- lpBytesReturned: value returned can be ignored but a valid pointer has to be passed to the driver

5.5.11. Send SCO Data

The data to be sent over SCO link with IOCTL_CSRBC_SEND_SCO_DATA requires SCO packet headers in order to be understood by the chip:

- lpInBuffer is a pointer to input data buffer
- nInBufferSize is size of the data to be sent
- lpBytesReturned: value returned can be ignored but a valid pointer has to be passed to the driver

5.5.12. Receive SCO Data

Use IOCTL_CSRBC_RECV_SCO_DATA:The data received (in lpOutBuffer) through

- lpOutBuffer is a pointer to a buffer containing the SCO header
- lpBytesReturned is actual number of bytes read

Usage is similar to IOCTL_CSRBC_SEND_SCO_DATA.

5.5.13. Stop SCO Interface

IOCTL_CSRBC_STOP_SCO_DATA closes the SCO channel opened by IOCTL_CSRBC_START_SCO_DATA. Subsequent SCO data IOTCLs, (IOCTL_CSRBC_SEND_SCO_DATA and IOCTL_CSRBC_RECV_SCO_DATA) fail.

5.5.14. Send Control Transfer

Use IOCTL_CSRBC_SEND_CONTROL_TRANSFER to perform a USB control transfer:

- lpInBuffer is a pointer to input data buffer
- nInBufferSize is size of the data to be sent
- lpOutBuffer is a pointer to output data buffer
- nOutBufferSize is size of the output buffer
- lpBytesReturned is actual number of bytes read

The buffer to send has a defined data format, which mirrors the USB request itself:

- Buffer[0] = bmRequestType
- Buffer[1] = bRequest
- Buffer[2,3] = value
- Buffer[4,5] = index

The actual data to be sent starts at Buffer[6]. The length field passed into DeviceIoControl should be the actual data + 6 bytes (to support the defined format).

If the bmRequestType of the control transfer happens to be a USB IN request, then the data received from the device is placed in lpOutBuffer and the actual number of bytes in lpOutBuffer is notified in lpBytesReturned.

5.6. High-speed Operation

To enable high-speed operation and minimal CPU usage, use the blocking IOCTL calls IOCTL_CSRBC_BLOCK_HCI_EVENT and IOCTL_CSRBC_BLOCK_HCI_DATA. These calls block until an event arrives. They can only be used when the driver is used in Overlapped mode.

5.6.1. Opening an Overlapped Device

To open a device in Overlapped mode, call CreateFile with the FILE_FLAG_OVERLAPPED flag.

5.6.2. Blocking Events

If an event is not expected to occur immediately, then call the `DeviceIoControl` `IOCTL_CSRBC_BLOCK_HCI_EVENT`.

In this example, an overlapped structure is created and the `overlapped.hEvent` is initialised with an event handle returned by a call to `CreateEvent` (WIN32 API). The `DeviceIoControl` exits immediately because it is an asynchronous call. The request is still pending, so the thread blocks on `WaitForSingleObject` on the `overlapped.hEvent`. When a HCI event arrives, the event is signalled and this unblocks the thread. Use `GetOverlappedResult` to find the status of the command.

Example:

```
OVERLAPPED overlapped = {0, 0, 0, 0};
overlapped.hEvent = wait_for_event;

status = DeviceIoControl
      (
               hci_handle,
               IOCTL_CSRBC_BLOCK_HCI_EVENT,
               0,
               0,
               0,
               0,
               &written,
               &overlapped
      );
WaitForSingleObject (wait_for_event, -1);
status = GetOverlappedResult (handle, &overlapped, &written, FALSE);
```

5.7. Blocking Data

`IOCTL_CSRBC_BLOCK_HCI_DATA` is similar to `IOCTL_CSRBC_BLOCK_HCI_EVENT`. This IOCTL blocks until HCI data arrives

6. Frequently Asked Questions

This section answers some frequently asked questions.

1. *How do I run WHQL tests and certify my driver?*

   See section 5 of this document.

2. *I have a driver that has WHQL installed and certified. I have modified the .inf file and the driver components. Should I re-certify my driver?*

   Yes. Even if a single component of a certified driver is modified, it should be re-certified. The change may be a simple one where the VID and PID in an `.inf` file are changed, or it may be a change in the driver source code and the driver is recompiled.

   The components of a certified driver are:

   - `.sys` files
   - `.inf` file
   - `.cat` file (catalogue file)

   The catalogue file is generated from the `.sys` files and `.inf` files. A change in any of the components makes the catalogue file invalid for use with the modified files. A new catalogue file must be generated from the modified components and then certified.

   The certification process is described in section 5 of this document.

3. *What is Error 52 "Windows cannot verify the digital signature for the drivers required for this device"?*

   If this error is reported while trying to install released drivers, it means that you have modified either the released `.inf` file or the released driver `.sys` file.

   Even if you have changed the `.inf` file and reverted the changes, you get an **Error 52**. The signed driver has "hash" of file, modified date, created date and other information inside the `.cat` file; which will not match if any of the details change.

   To find the source of the problem, supply CSR with the file `setupapi.dev.log` in the `c:\Windows\inf\` folder from the machine where the problem occurs.

4. *Is it mandatory to pass the USB-IF test? If it is, how do I set-up the test?*

   Yes. It is mandatory to pass the USB-IF test to get your driver certified:

   - It is made mandatory in Windows Logo Kit version 1.6. For more Information, see USB-IF testing for WLK at http://www.microsoft.com.

   The USB product has to go through the USB-IF compliance tests and get a test ID issued by USB-IF. For more Information, see USB-IF Compliance at http://www.usb.org/.
   The test ID is the input to USB-IF WHQL:

1. Open **Device Console** from **Explorers** tab.

The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.

Figure 6.1: Open Device Console

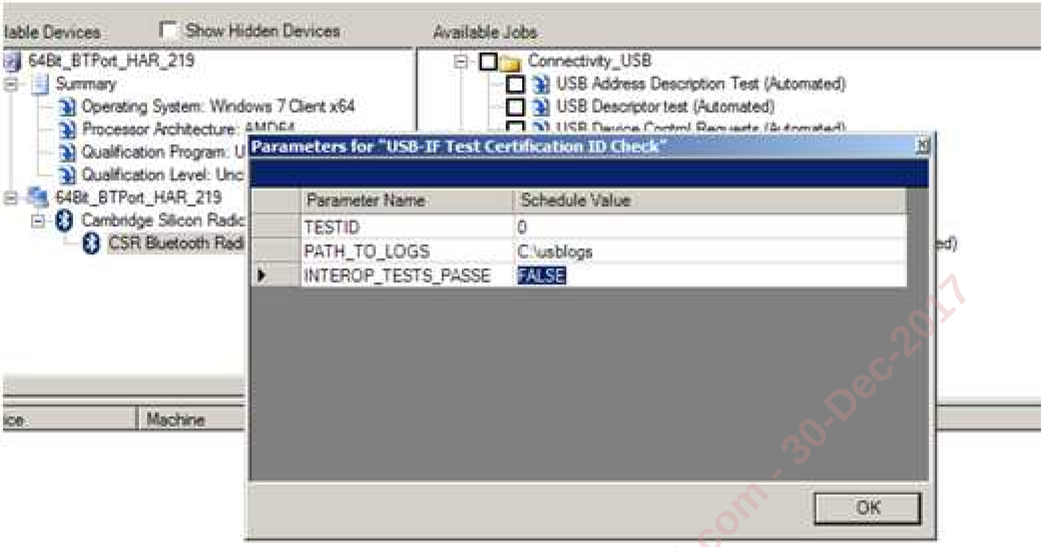
2. Select your submission in the left panel.

The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.

Figure 6.2: Edit Parameters for USB-IF Test Certification ID Check

3. Right-click **USB-IF Test Certification ID Check** in **Available Job** listed
4. Click **Edit Parameters**.
   A table with test parameters appears. See Figure 7.3 for an example from Version 1.6 WLK. The table looks different in other versions.

5. Enter the test id number in **TESTID** field and change the **INTEROP_TESTS_PASSED** field to **TRUE**.



Parameters for "USB-IF Test Certification ID Check"

	Parameter Name	Schedule Value
	TESTID	0
	PATH_TO_LOGS	C:\usblogs
▸	INTEROP_TESTS_PASSE	FALSE

Figure 6.3: Table with Test Parameters for USB-IF Test Certification ID Check

6. Why do some of the tests listed under Connectivity_USB category fail during WHQL?

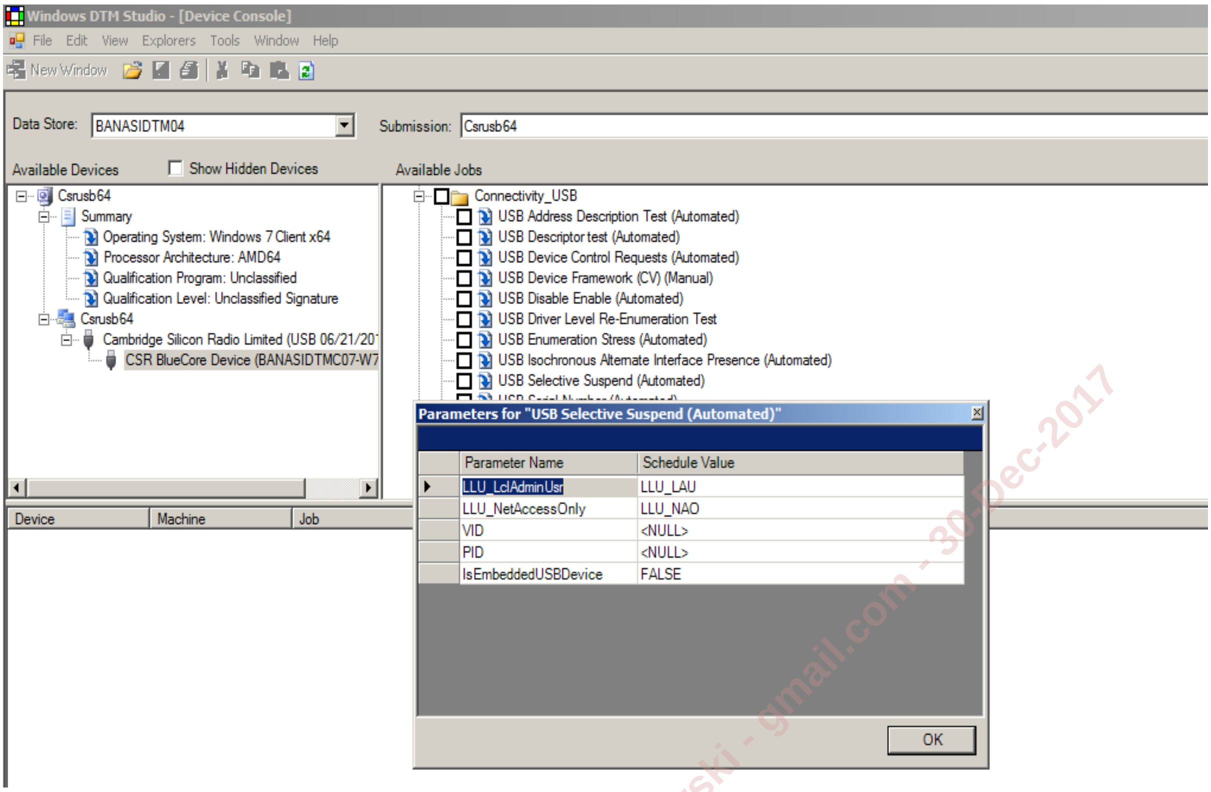
   The following tests listed under Connectivity_USB fail because the USB device is not set to be an Embedded USB device:

   - USB Device Framework(CV)
   - USB Driver Level Re-Enumeration Test
   - USB Selective suspend
   - USB Serial number

To set an Embedded USB device:

1. Right-click on the test and click Edit Parameters.

   A table with test parameters appears, see Figure 7.4.
2. Change the IsEmbeddedUsbDevice field to TRUE.



Windows DTM Studio - [Device Console]

Data Store: BANASIDTM04

Submission: Csrusb64

Available Devices ☐ Show Hidden Devices

- Csrusb64
  - Summary
    - Operating System: Windows 7 Client x64
    - Processor Architecture: AMD64
    - Qualification Program: Unclassified
    - Qualification Level: Unclassified Signature
  - Csrusb64
    - Cambridge Silicon Radio Limited (USB 06/21/20
      - CSR BlueCore Device (BANASIDTMC07-W7

Available Jobs

- ☐ Connectivity_USB
  - ☐ USB Address Description Test (Automated)
  - ☐ USB Descriptor test (Automated)
  - ☐ USB Device Control Requests (Automated)
  - ☐ USB Device Framework (CV) (Manual)
  - ☐ USB Disable Enable (Automated)
  - ☐ USB Driver Level Re-Enumeration Test
  - ☐ USB Enumeration Stress (Automated)
  - ☐ USB Isochronous Alternate Interface Presence (Automated)
  - ☐ USB Selective Suspend (Automated)

Parameters for "USB Selective Suspend (Automated)"

Parameter Name	Schedule Value
LLU_LclAdminUsr	LLU_LAU
LLU_NetAccessOnly	LLU_NAO
VID	<NULL>
PID	<NULL>
IsEmbeddedUSBDevice	FALSE

Figure 6.4: Set IsEmbeddedUsbDevice Field to TRUE

7. Complete Example

To see a complete example that demonstrates the use of the USB device driver interface in the source release, access the folder **CSRSource\devHost\UsbDeviceDriver\example**.

Terms and Definitions

ACL	Asynchronous Connection-Less
API	Application Programming Interface
BlueCore®	Group term for CSR’s range of Bluetooth wireless technology chips
Bluetooth®	Set of technologies providing audio and data transfer over short-range radio connections
CPU	Central Processing Unit
CSR	Cambridge Silicon Radio
DFU	Device Firmware Upgrade
DTM	Driver Test Manager
etc	et cetera, and the rest, and so forth
HCI	Host Controller Interface
ID	Identifier
INF	Information File
IOCTL	Input/output Controls
MSDN	Microsoft Developers Network
PID	Product ID
SCO	Synchronous Connection-Oriented
USB	Universal Serial Bus
USB-IF	Universal Serial Bus-Implementers forum
VID	Vendor ID
WHQL	Windows Hardware Quality Labs
WLK	Windows Logo Kit