

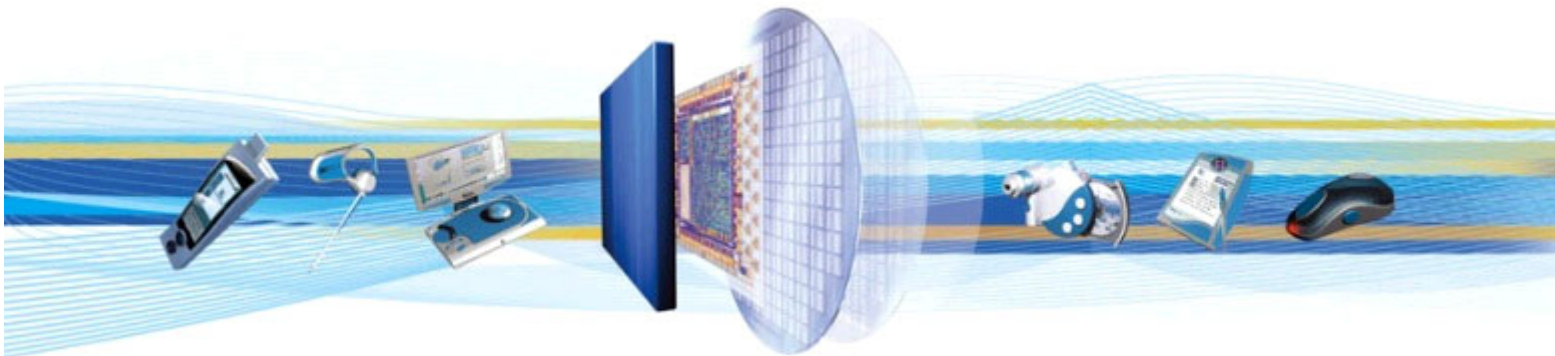


**BlueCore®**

**BlueSuite™**

**User Guide**

**Issue 3**



**CSR**  
Cambridge Science Park  
Milton Road  
Cambridge CB4 0WH  
United Kingdom  
Registered in England 3665875  
Tel.: +44 (0)1223 692000  
Fax.: +44 (0)1223 692001  
[www.csr.com](http://www.csr.com)

## Document History

Revision	Date	Change Reason
2	12 MAR 08	Original publication of this document.
3	13 NOV 08	Section on Command Line Tools transport options added. Minor text corrections.

## Trademarks, Patents and Licences

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

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.

CSR's products are not authorised for use in life-support or safety-critical applications.

Refer to [www.csrsupport.com](http://www.csrsupport.com) for compliance and conformance to standards information.

## Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>System Requirements</b>	<b>7</b>
2.1	Operating System	7
2.2	Software Required	7
<b>3</b>	<b>Install BlueSuite</b>	<b>8</b>
<b>4</b>	<b>Install a USB Driver</b>	<b>10</b>
4.1	Configuring the USB INF File For Your Device	10
4.1.1	Verifying VID and PID Operating System Support	10
4.1.2	Adding the VID and PID	10
4.2	Change the USB Driver	11
<b>5</b>	<b>Firmware Overview</b>	<b>13</b>
5.1	Firmware File Formats	13
5.2	Persistent Store	13
5.3	Firmware Build Types	14
5.4	Upgrading Firmware	14
<b>6</b>	<b>Getting Started</b>	<b>15</b>
6.1	Exploring the Bluetooth Stack	15
<b>7</b>	<b>BlueSuite Programs</b>	<b>16</b>
7.1	Transport Options for Command Line Tools	16
7.2	BlueFlash	17
7.2.1	User Interface	17
7.2.2	Troubleshooting BlueFlash	19
7.2.3	BlueFlashCmd	19
7.3	PSTool	20
7.3.1	PsCli	20
7.3.2	Documentation	20
7.4	BlueTest	20
7.4.1	BlueTest3	20
7.4.2	Documentation	20
7.5	BTCli	21
7.5.1	Example Usage	21
7.6	RFCLi	21
7.7	BCFMRadio	22
7.8	BCFMcli	22
7.9	E2Util	22
7.10	E2BlueCmd	22
7.10.1	EEPROM File Formats	22
7.11	DFU Wizard	23
7.12	DFU Babel	23
7.12.1	Upgrading The USB to SPI Converter Firmware	23
7.13	DFU ToolTips	24
7.14	DFU Tools	24
<b>8</b>	<b>TrueTest Toolkit</b>	<b>25</b>
<b>9</b>	<b>Document References</b>	<b>26</b>
	<b>Terms and Definitions</b>	<b>27</b>

## List of Figures

Figure 3.1	Full Installation of BlueSuite .....	8
Figure 3.2	Additional Tasks Dialogue .....	9
Figure 4.1	Changing the Driver .....	12
Figure 7.1	BlueFlash User Interface in Windows Vista .....	17
Figure 7.2	A DFU ToolTip .....	24

## List of Tables

Table 5.1	Firmware File Formats .....	13
Table 7.1	Command Line Transport Options .....	16
Table 7.2	Explanation of the BlueFlash User Interface .....	18
Table 7.3	Example Connections and How to Make Them with BTCLI .....	21

# 1 Introduction

BlueSuite™ is a BlueCore® development suite designed to help with your Bluetooth® wireless technology development program.

BlueSuite contains:

- Graphical and command-line applications to configure, test and program the BlueCore device in your design.
- The TrueTest™ Toolkit, a set of libraries for developing production test applications for your BlueCore enabled devices.
- Device drivers to communicate with BlueCore.
- Documentation to help development.

This user guide explains:

- The minimum system requirements in order to be able to run Bluesuite. See Section 2.
- How to install BlueSuite Section 3 and Section 4.
- The features and functions Bluesuite contains. See Section 7.

## 2 System Requirements

Minimum system requirements for BlueSuite are:

- Pentium processor based personal computer
- 64MB of RAM
- 20MB of available hard disk space

### 2.1 Operating System

BlueSuite can be used on the following operating systems:

- Microsoft Windows XP Service Pack 2 and later (not x64)
- Microsoft Windows Vista (not x64)
- Microsoft Windows 2000 Service Pack 3 and later (only for versions of BlueSuite prior to 2.0)

See the release notes for testing information and detailed operating system requirements.

### 2.2 Software Required

Your copy of BlueSuite comes with:

- Microsoft .NET Framework Version 2.0 Redistributable Package
- Visual C++ 2005 Redistributable Package

The BlueSuite installer automatically installs these packages on your computer if they are missing.

### 3 Install BlueSuite

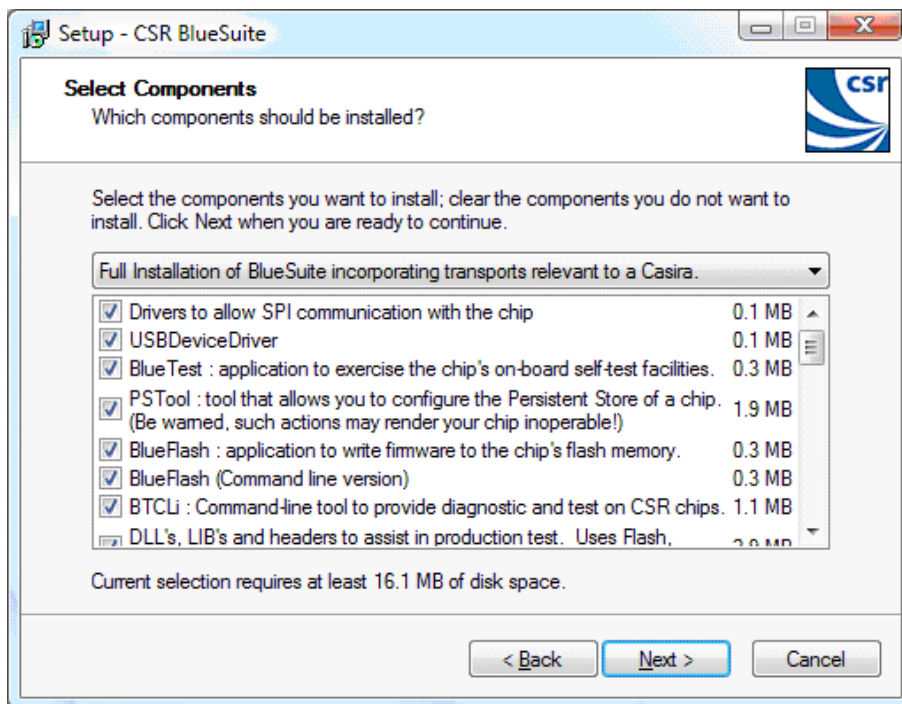
**Note:**

A bug in BlueSuite version 1.23 means that even if a later version of BlueSuite is installed, the version information for the USB driver may still indicate 1.23. If version 1.23 has previously been installed on the PC, unplug all CSR hardware, make sure you have the latest version of BlueSuite installed and run `usbpurger` from the BlueSuite installation directory. When a CSR BlueCore is plugged in, the Hardware Wizard prompts you for drivers.

Run the BlueSuite installer and follow the instructions displayed by the setup wizard.

By default the installer does a full installation, as shown in Figure 3.1. If you do not want to install all of the possible components, be sure that you install:

- **Drivers to allow SPI communication with the chip.** You will need these to back-up your BlueCore device.
- **USBDeviceDriver** if you want to connect your device to your computer's USB port.
- **DLL's, LIB's and headers to assist in production test** if you want to do production testing.



**Figure 3.1: Full Installation of BlueSuite**

When you get to the Additional Tasks dialogue in Figure 3.2, you have to select the default debug transport. This configures the default behaviour of the BlueSuite programs. You can change this after installation by changing environment variable `SPI_TRANS` to `LPT` or `USB`. See Section 7.1



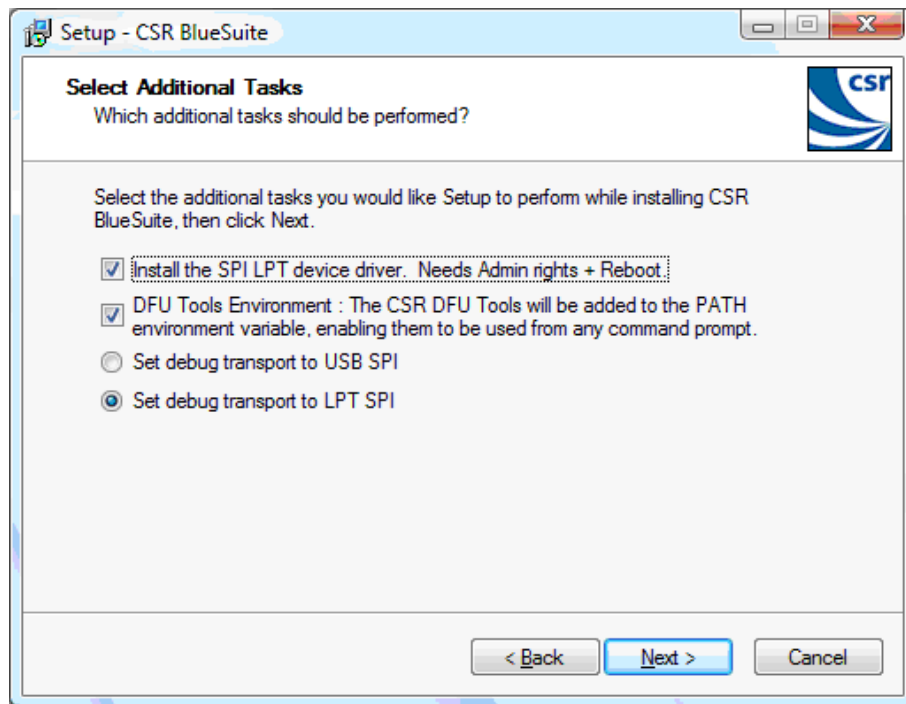


Figure 3.2: Additional Tasks Dialogue

## 4 Install a USB Driver

BlueCore devices can be used with a USB host interface. The USB host interface is not a debugging interface and can be used by end users or in production modules. If you want to use the USB host interface to BlueCore, you must first configure your BlueCore for the USB host interface, and then install a USB device driver.

### 4.1 Configuring the USB INF File For Your Device

If you are developing a USB device containing a BlueCore chip, during your development you can use the CSR USB driver.

When a USB device is plugged into a computer, it sends identifying information to the computer so the computer can decide which driver to use with it. Two of the pieces of information sent are the Vendor ID (VID) and the Product ID (PID).

If your device's VID and PID are not currently supported by the CSR USB Driver, you must add them to the driver INF (information) file so Windows recognises that the CSR USB driver will work for your device.

This chapter describes how to check the file to determine if your device is supported, and if not, how to add the device.

#### 4.1.1 Verifying VID and PID Operating System Support

The CSR USB INF file is `CSRBlueCoreUSB.inf`, by default located in `ProgramFiles/CSR/BlueSuite/drivers`. Open the file with a text editor, and search for your device VID and PID. E.g. for a standard example Bluetooth device:

```
%Example.DeviceDesc%=CSRBC.Inst.NTx86.5.1,  
USB\VID_1234&PID_5678 ; Example USB Device VID&PID
```

If your device is present in `CSRBlueCoreUSB.inf` skip Section 4.1.2 as the CSR USB driver supports your device.

#### 4.1.2 Adding the VID and PID

The example below adds a device with a VID of `0x1234` and a PID of `0x5678` with a name of `Generic Device`. All PIDs and VIDs in an INF file are in hexadecimal.

This procedure adds the device to all supported operating systems of the INF file. While just the operating system of interest could be modified, CSR recommend you add the information to all three possible operating systems, for completeness.

CSR recommend you add the device information after the final line in each section.

1. Create a backup copy of `CSRBlueCoreUSB.inf`.
2. Uncheck the **Read-only** attribute of the file before you edit it.
3. 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
```

4. Add the device to the 32-bit Windows XP and Vista 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
```

5. You can add the device to the 64-bit Windows XP and Vista section of the file, but CSR does not support 64-bit platforms:

```
;
; 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
```

6. 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"
```

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

The CSR USB driver now supports your device.

## 4.2 Change the USB Driver

By default Windows Vista and Windows XP use the pre-installed generic Microsoft Bluetooth USB drivers when a CSR USB device is connected. The Microsoft driver will not enable you to connect to your BlueCore device using BlueSuite. In order to use BlueSuite over USB the CSR USB device drivers must be installed.

To install the CSR drivers:

1. If a previous version of BlueSuite is installed, remove the old drivers:
  - 1.1 Disconnect any CSR USB devices
  - 1.2 Run `usbpurger.exe` from the BlueSuite install area
2. Plug the CSR device's USB cable into the PC
3. Open the **Control Panel->System**, select the **Hardware** tab and click **Device Manager**.
4. Expand the **Bluetooth Radio** section. Right-click on **Generic Bluetooth radio** and select **Update Driver...**, as shown in Figure 4.1.

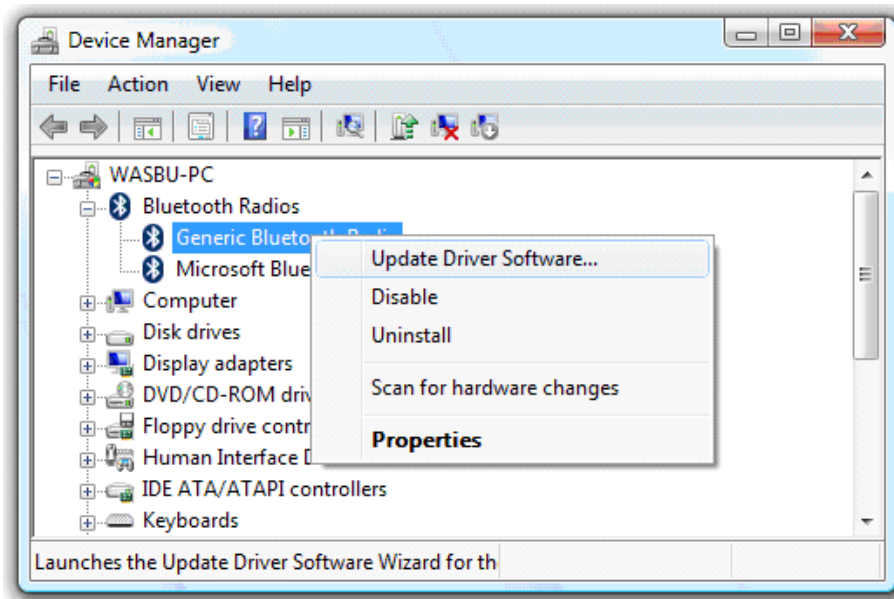


Figure 4.1: Changing the Driver

5. Complete the **Hardware Update Wizard** :

**For Vista:**

- 5.1 **Browse my computer**
- 5.2 **Let me pick from a list of devices to install**
- 5.3 **Have disk**
- 5.4 **Browse** and navigate to the **Drivers** folder installed as part of BlueSuite and select **CSRBlueCoreUSB**. Click **Next**

**For XP:**

- 5.1 **No, not this time**
- 5.2 **Install from a list or specified location (Advanced)**
- 5.3 Select **Search for the best driver in these locations**, tick **Include this location in the search**, then **Browse** and navigate to the **drivers** folder installed as sub-directory of BlueSuite. Click **Next**

6. **CSR BlueCore Controller** should now appear in the list of **Universal serial Bus Controllers**

CSR recommends using PSTool via USB to check that communications can be successfully established.

## 5 Firmware Overview

CSR's BlueCore ICs contain control software (called firmware). Firmware in BlueCores with flash can be updated using tools in BlueSuite:

- **BlueFlash** (see Section 7.2) is used during development to update your BlueCore's firmware using the Serial Peripheral Interface (SPI) debugging interface. **TestFlash** (part of **TrueTest**, see Section 8) can be used for the same purpose on the production line.
- **DFUWizard** (see Section 7.11) can sometimes be used to upgrade the firmware both in development and in the field.

DFU (Device Firmware Upgrade) is a USB standard that allows end users to upgrade the BlueCore's firmware over the USB or UART (Universal Asynchronous Receiver Transmitter) interface.

DFUWizard requires some firmware (the loader) to already be present in the device. The loader can only be updated by BlueFlash.

The firmware version number is an important piece of information. You can identify your current firmware version using BlueFlash, see Section 7.2.

### 5.1 Firmware File Formats

Table 5.1 describes the different firmware file formats you may encounter.

File Extension(s)	Description
.xpv / .xdv	The standard BlueCore firmware file format holds the binary data for firmware releases in two separate files, each of which hold part of the data.
.xuv	This firmware file format is one file containing the contents of both the .xpv file and the .xdv file.
.xhv	Chips with integrated flash have firmware files containing extra production information about the flash.
.dfu	File format for use with <b>DFUWizard</b> .  <b>Note:</b>  <b>DFUTools</b> , see Section 7.14, can convert .xpv and .xdv files into a .dfu file.

**Table 5.1: Firmware File Formats**

The .dfu file format is a flexible container format that can hold multiple different firmware images (for BlueCore2 and later chips) and multiple versions of the Persistent Store. This means you can have a universal .dfu file that works for multiple hardware devices.

See *BlueCore Device Firmware Upgrade Overview* for a description of what DFU is and how it is used.

### 5.2 Persistent Store

As well as containing the firmware image (i.e. control software), firmware files can also contain configuration information known as Persistent Store (PS). DFU files can contain information for a partial or full update of the PS. Any areas of the PS that are not in the DFU file will remain as they were before the update. Other firmware files cannot contain partial information; they either preserve or replace the PS.

Firmware upgrades from the CSR support website do not contain any PS settings, thus the current settings are preserved. Firmware dumps that you make contain PS settings, and so will overwrite any existing ones if you upgrade using a previously dumped file.

**Note:**

If firmware with no PS is downloaded onto a blank flash, a new PS is created on the flash memory using default values suitable for a Casira module. However, some keys require calibration per module for optimal performance.

See `pskeys.html` in the firmware release for details on using individual PS Keys. Some of the same information is available by clicking the **Describe** button in **PSTool**, see the *PSTool User Guide*.

### 5.3 Firmware Build Types

The Bluecore's Bluetooth stack firmware comes in two forms, either containing the Bluetooth stack layers up to the Host Controller Interface (HCI) layer, or containing all the stack layers up to the Radio Frequency COMMunication (RFCOMM) layer.

Since Firmware version 18, firmware builds are called Unified. They contain all the Bluetooth stack layers up to RFCOMM, but can be configured to present either an HCI interface, or an RFCOMM interface. The behaviour is controlled by `PSKEY_ONCHIP_HCI_CLIENT`, see the firmware release files for more information.

All types of firmware builds can be held in any BlueCore firmware file format.

### 5.4 Upgrading Firmware

Go to the Bluetooth Firmware section on the CSR support website ([www.csrsupport.com/BluetoothFirmware](http://www.csrsupport.com/BluetoothFirmware)) to get the latest BlueCore firmware files. Use **BlueFlash** or **DFUWizard** to download the firmware to your BlueCore device.

**Note:**

Casira users should see the *Casira User Guide* for instructions on upgrading their firmware.

## 6 Getting Started

When you have connected your BlueCore device(s) to your computer, you can use BlueSuite tools to communicate with them.

To test you have connected your BlueCore device(s) successfully, you can use:

- The command-line tool **btcli** (Section 7.5).
- One of the graphical programs PStool (See the *PStool User Guide*) or BlueTest3 (See the *BlueTest User Guide*).

### 6.1 Exploring the Bluetooth Stack

Use **btcli** if you want to explore the Bluetooth stack at all before starting development, as it provides a lower-level interface than the graphical programs in BlueSuite.

The Bluetooth stack is described in the *Specification of the Bluetooth System* from [www.bluetooth.com](http://www.bluetooth.com).

## 7 BlueSuite Programs

This chapter describes the programs included in BlueSuite. The features of each program are briefly described and any relevant documentation is listed. Some programs without separate user guides are described in full detail.

### 7.1 Transport Options for Command Line Tools

Some command line tools in BlueSuite take a `-trans` option, which controls the transport options. Table 7.1 indicates which variables can be used with each tool:

Variable	e2bluecmd/ blueflashcmd	pscli	Default	Description	Example Values
CSRTRANS	-	x	SPI	The transport to use	SPI, USB, BCSP, H4 and H5
HCIPORT	-	x	autodetect if USB, otherwise COM1	The port to use for the selected HCI transport if not CSRTRANS=SPI	com1, com2, csr0
HCIBAUD	-	x	115200	The baud rate to use for uart based HCI transports	115200, 38400, etc.
SPITRANS	x	x	LPT	The SPI transport to use if CSRTRANS=SPI	LPT, USB
SPIPORT	x	x	autodetect	The port to use for the selected SPI transport	<ul style="list-style-type: none"> <li>▪ A LPT port number: 1, 2 etc.</li> <li>▪ A USB-to-SPI converter's ordinal position: 1, 2 etc.</li> <li>▪ A USB-to-SPI converter's serial number (as printed on the converter e.g. 185920)</li> </ul>
SPIMUL	x	x	0	The device to use if the SPI port is multiplexed	0, 1, etc.

**Table 7.1: Command Line Transport Options**

**Note:**

Where appropriate, BlueSuite tools attempts to prepend `\\. \` to a string in HCIPORT. For example the name of a CSR USB device on WinXP is `\\. \csr0`; BlueSuite tools re-interpret `csr0` as `\\. \csr0`.

**Examples**

To use USB-SPI (where 185920 is the serial number printed on the USB-SPI):

```
-trans "CSRTRANS=SPI SPITRANS=USB SPIPORT=185920"
```

(or just "SPITRANS=USB SPIPORT=185920")

To use BCSP on COM1 at 38.4 kbaud:

```
-trans "CSRTRANS=BCSP HCIPORT=com1 HCIBAUD=38400"
```

It is possible to set these as Windows environment variables, which are then used as the default setting for a variable e.g. "set SPITRANS=USB" would change the default SPI transport from LPT-SPI to USB-SPI.



## 7.2 BlueFlash

BlueFlash is a utility that allows you to download and upload firmware to and from the flash memory on the Casira modules or on your own BlueCore designs.

As described in Section 5, there are several different firmware file formats, and firmware files may contain Persistent Store settings. Read Section 5.2 to make sure you do not accidentally change Persistent Store settings and to see which parts of BlueCore's firmware is updated by BlueFlash.

### Important Note:

Do not erase the Persistent Store with **Flash Erase** (either by selection or via **Full Erase**) unless you are completely sure.

Fully erasing the flash removes all Persistent Store settings. Use **Full Erase** only in the event that there may be a problem with the Persistent Store and you want to reload to a previous known working version.

### 7.2.1 User Interface

BlueFlash opens with a user interface window, see Figure 7.1.

Table 7.2 explains the function of each button or area.

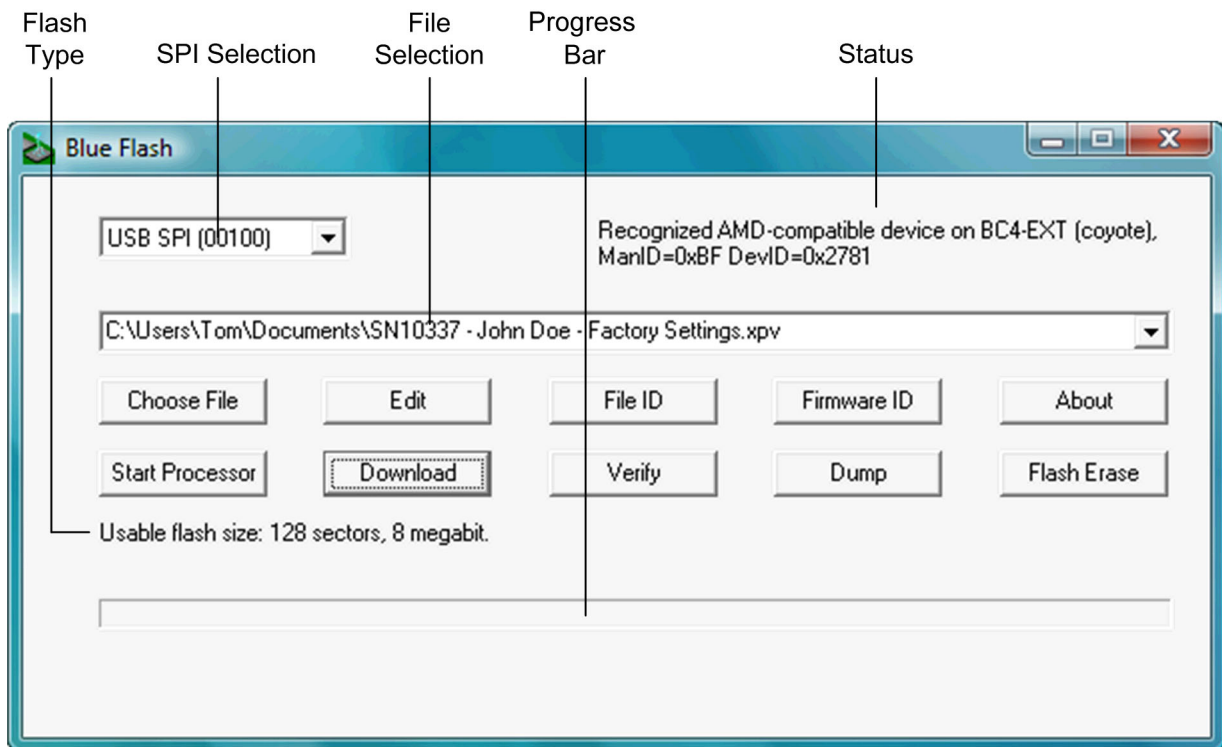


Figure 7.1: BlueFlash User Interface in Windows Vista

Button or Area	Explanation
Choose File	<p>Opens a dialogue box for selection of the firmware file to download to the flash on the BlueCore Module. You may open a .xpv, .xuv or .xlv firmware file</p> <p><b>Note:</b></p> <p>If you open a .xpv file, the .xlv file must be present in the same folder.</p>
Edit	<p>Opens a hexadecimal editing window. CSR recommends that you do not edit the .xpv file in any way. If the code is edited, the Bluetooth protocol stack may behave erratically and it will no longer be Bluetooth qualified.</p>
Download	<p>Copies the selected file to the flash on the BlueCore module. This process updates each flash block as required and verifies them in turn.</p>
Verify	<p>Compares the selected file with the code programmed into the flash on the BlueCore module. <b>Verify</b> only compares components contained in the file. The status area shows any reported differences.</p>
Status	<p>Provides a text report of the status of the current download, upload or verify.</p>
Progress Bar	<p>Displays a visual indication of progress for the current download, upload or verification.</p>
Start Processor	<p>Changes depending on the status of the processor. If it says <b>Stop Processor</b>, the processor is running. It must be stopped before attempting to download code. When the new code has been loaded, the processor can be restarted by clicking <b>Start Processor</b>.</p>
Firmware ID	<p>Identifies which version of firmware is currently loaded on the module. BlueFlash does not identify pre-Beta 10 firmware releases. If BlueFlash is unable to identify the firmware version, you can still upgrade the firmware using this utility.</p>
Dump	<p>Allows the user to dump the contents of the flash on the BlueCore module to a file. The data can be stored either as a pair of .xpv/.xlv files, as a combined .xuv file or as a raw binary file. The dumped file also contains all of the Persistent Store settings, allowing you to restore to a known firmware build and Persistent Store settings.</p>
Flash Erase	<p>Provides options for erasing some or all of the flash memory including the Persistent Store area.</p>
File ID	<p>Displays the name and version of firmware of the file selected for download on the module.</p>
File Selection	<p>Displays the name and location of the file selected for download.</p>
Flash Type	<p>Displays the type of flash memory that has been identified on the BlueCore module. Flash type is identified after you stop the processor. Different types of flash memory may have different memory block structures that require adjustments to BlueCore's memory map. If the flash type indicates <b>Unrecognised Flash assuming SST</b>, either the flash on the BlueCore module is not supported, or there is a problem with the SPI connection. See Section 7.2.2.</p>
SPI Selection	<p>Allows the user to select the LPT or USB port to which the SPI cable is connected.</p>
About	<p>Indicates the version of the BlueFlash application in use. CSR advises using the most up to date version. Refer to the CSR support website for available updates.</p>

**Table 7.2: Explanation of the BlueFlash User Interface**

## 7.2.2 Troubleshooting BlueFlash

Potential problems:

- If BlueFlash cannot identify the current build of firmware on BlueCore, check your firmware version. BlueFlash does not identify pre-Beta 10 firmware releases.
- If BlueFlash cannot identify flash type, BlueFlash fails to stop processor or BlueFlash times out trying to make a connection to BlueCore, ensure that the following actions have been completed:
  - Run `InstParSpi.bat` (`C:\Program Files\CSR\BlueSuite\drivers`) to copy and register the `parspi.sys` device driver. You need administrative rights to install the device driver on a Windows XP, Windows 2000 system or Windows Vista. Reboot your PC.
  - Driver has been installed in the registry by looking for `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Parspi`
  - SPI cable is correctly connected
  - Check for conflicts with other drivers, especially printer drivers (this includes the PDF converters). Conflicting printer drivers should be removed. To amend or remove a driver, go to **Start / Settings / Printers**. Right click for **Server Properties**. Click on the **Ports** tab. Move the port to another location or delete the device.
  - Ensure that another SPI application is not trying to access the SPI port. SPI uses mutexes around port access, therefore another application may lock the port. Close all SPI based applications (use Task Manager if necessary) and retry. Reboot if necessary.

## 7.2.3 BlueFlashCmd

BlueFlashCmd is a command-line tool to download and upload firmware to flash memory on your BlueCore device. It can do everything BlueFlash does.

Type `BlueFlashCmd.exe -help` for usage instructions.

**Note:**

Running BlueFlashCmd will reset your BlueCore chip

## 7.3 PSTool

PSTool is an editing suite that allows you to read and modify the Persistent Store. The Persistent Store consists of configuration keys that modify the operation of the firmware. BlueCore is a very flexible device with many setup options. PS Keys must be set correctly for BlueCore to work in a particular design. Some PS Keys must be calibrated for each module.

### 7.3.1 PsCli

PsCli is a command-line interface for reading and modifying the Persistent Store. Type `pscli.exe -help` to see usage instructions.

### 7.3.2 Documentation

PSTool and pscli are described in the *PSTool User Guide*.

See `pskeys.html` in the firmware release for details on using individual PS Keys.

## 7.4 BlueTest

BlueTest is a program that enables you to execute BlueCore's Built In Self Test (BIST) functions for RF testing. The BIST functions consist mainly of low level radio tests (transmitting a continuous wave at a prescribed frequency or receiving pseudo random data and calculating bit error rates). Included are tests for the PCM port and other internal blocks.

### 7.4.1 BlueTest3

BlueTest3 is a newer product which will replace BlueTest. Both programs support all of the currently relevant tests, but only BlueTest3 can make debug interface connections over the *serial peripheral interface* (SPI) transport.

BlueTest is still distributed as it supports some older tests in firmware versions 22 and earlier that BlueTest3 does not support.

#### Important Note:

CSR recommends using BlueTest3 unless you need features available only in BlueTest.

### 7.4.2 Documentation

See the *BlueTest User Guide*.

## 7.5 BTCli

Bluetooth Command Line Interface (BTCli) is a command-line program that lets you send Host Controller Interface (HCI) commands to your BlueCore device. Host computers use the HCI interface (part of the Bluetooth specification) to communicate with Bluetooth controllers (e.g. your BlueCore chip).

A basic user guide for BTCli is provided as a text file (`btcliug.txt`) in the BlueSuite directory.

To run BTCli, you can run it from the Start Menu and use the graphical dialogue to select a host interface, or you can run it from the command-line.

To run it from the command-line, open a Command Prompt window and navigate to the BlueSuite directory. From there, run BTCli with the appropriate arguments for the current host interface:

```
btcli <transport> <port> <baud rate>
```

### 7.5.1 Example Usage

Table 7.3 shows some examples:

Desired Connection	Command Line
BCSP on COM port 1 at 115.2 kbaud	<code>btcli bcspl com1 115200</code>
H4 on COM port 3 at 38.4 kbaud	<code>btcli h4 com3 38400</code>
USB (No data rate is required)	<code>btcli usb csr0</code> or <code>btcli usb BTWUSB_0</code>

**Table 7.3: Example Connections and How to Make Them with BTCli**

The most common use for BTCli is to activate Inquiry/Page Scan and allow the module to accept over air test commands from an automated tester. With BTCli this is very simple, involving only two commands: `slave` (which activates the scan mode and enables auto accept of connections) and `edutm` (*enable device under test mode*). A printout from BTCli carrying out these commands is shown below. Text entered by the user is shown emphasised.

```
C:\Program Files\CSR\BlueSuite>btcli.exe usb csr0
Version 2.0
Copyright (C) 2000-2008 Cambridge Silicon Radio Ltd.; all rights reserved
usb on csr0 at 12000000 bps
slave
sef connection_setup all auto_accept
wse inquiry_page
rba
command_complete nhcp:0x01 set_event_filter success
command_complete nhcp:0x01 write_scan_enable success
command_complete nhcp:0x01 read_bd_addr success ba:0x00025b02164c
edutm
command_complete nhcp:0x01 enable_device_under_test_mode success
```

## 7.6 RFCli

RFClI is a command line utility for communicating with RFCOMM firmware builds. It enables you to send and receive RFCOMM primitives to the chip (as well as other protocols). It also has a Tool Command Language (TCL) interpreter built in so you can write scripts.

See `rfcli.html` for a quick introduction, and `rfcli.txt` for full documentation.

## 7.7 BCFMRadio

### Important Note:

Read the document referenced below before running this program, as you must configure your BlueCore device before this program will connect to it.

BCFMRadio is a graphical tool for exercising the FM receiver on Bluecore5-FM. It is able to communicate with the Bluecore using I<sup>2</sup>C or BCCMDs using BCSP, H4, H4 Deep Sleep, H5, USB and SPI. BCFMRadio provides a user-friendly interface allowing configuration of the BlueCore5-FM radio.

See *Operating the BlueCore5-FM Radio Application Note*.

## 7.8 BCFMcli

### Important Note:

Read the document referenced below before running this program, as you must configure your BlueCore device before this program will connect to it.

BCFMcli is a command line application for controlling the FM receiver on Bluecore5-FM. It is able to communicate with the Bluecore using I<sup>2</sup>C or BCCMD with SPI over an LPT port.

Type `help` from within BCFMcli for usage instructions.

See *Operating the BlueCore5-FM Radio Application Note*.

## 7.9 E2Util

### Note:

E2BlueCmd (Section 7.10) will replace E2Util. E2Util is deprecated and may not be included in future releases.

E2Util is a command-line tool to do basic operations on an I<sup>2</sup>C EEPROM like dumping the EEPROM's raw contents, clearing it, and programming it.

E2Util controls your BlueCore chip over SPI, and does not require running firmware.

Type `e2util -help` for usage instructions.

## 7.10 E2BlueCmd

E2BlueCmd is a new command-line tool which replaces E2Util. E2BlueCmd performs EEPROM device operations using a boot program running on BlueCore. This approach makes it faster than E2Util, and provides support for gang programming operations using the CSR gang programmer hardware. As well as dumping and writing files, it can write the BlueCore EEPROM header, read and write blocks, and verify EEPROM contents against a file.

E2BlueCmd works over the SPI interface.

Type `e2bluecmd -help` for usage instructions.

### 7.10.1 EEPROM File Formats

E2BlueCmd saves EEPROM files in a different format to those saved by E2Util, and should only be loaded back to EEPROM with E2BlueCmd.

Files saved by E2Util can be loaded back to EEPROM with E2Util or E2BlueCmd.

## 7.11 DFU Wizard

The DFU Wizard is a program for end users to upgrade the firmware on their BlueCore devices. The upgrade process is documented in the *BlueCore Device Firmware Upgrade (DFU) Overview*.

## 7.12 DFU Babel

The USB to SPI Converter (or Babel) contains a BlueCore3-Multimedia chip with special firmware that transforms the USB packets from the host computer to SPI signals sent out over Category 5 cable.

DFU Babel is a program that lets you upgrade your USB to SPI Converter with new firmware. If CSR release a firmware upgrade for the USB to SPI Converter, you will be able to find it on [www.csrsupport.com](http://www.csrsupport.com).

See the *USB-SPI Converter User Guide*.

### 7.12.1 Upgrading The USB to SPI Converter Firmware

#### Important Note:

Do not attempt to upgrade a USB to SPI Converter with firmware designed for different hardware (e.g. Casira firmware). The USB to SPI Converter needs special firmware and any other firmware may make it permanently unusable.

To upgrade the firmware on your USB to SPI Converter:

1. Connect the USB to SPI Converter to the host computer. You do not need to connect the converter to a BlueCore module.
2. Run DFU Babel from the Start Menu. Alternatively run `dfubabel.exe` from the command-line. For command-line usage type `dfubabel.exe -help`.
3. Your USB to SPI Converter is now in DFU mode. You may have to manually load the DFU mode device driver for it. This driver is installed by `CSRBlueCoreUSB.inf` as before. While your device is in DFU mode, it appears to be a normal BlueCore3-Multimedia device attached over USB.
4. Follow the instructions in the DFU Wizard to upgrade the firmware on your USB to SPI Converter.
5. Reset your USB to SPI Converter to exit DFU mode and start the new firmware. Unplug it and plug it back in.

### 7.13 DFU ToolTips

BlueSuite comes with DFU ToolTips, a shell extension that shows tool tips for DFU files in Windows. To see the tool tip, point the mouse over a DFU file, and wait for the tool tip to appear. Figure 7.2 shows a DFU tool tip.

If the tool tip does not appear, you may have disabled tool tips in your version of Windows. To enable tool tips for files:

1. Go to the **Control Panel**
2. Open **Folder Options**
3. Select the **View** tab
4. Scroll down the Advanced settings to **Show pop-up description for folder and desktop items**. Tick this box.
5. Click **OK**

DFU ToolTips also lets you right-click on a DFU file, and **Download to BlueCore**. This runs the DFU Wizard to download the DFU file to your BlueCore device.

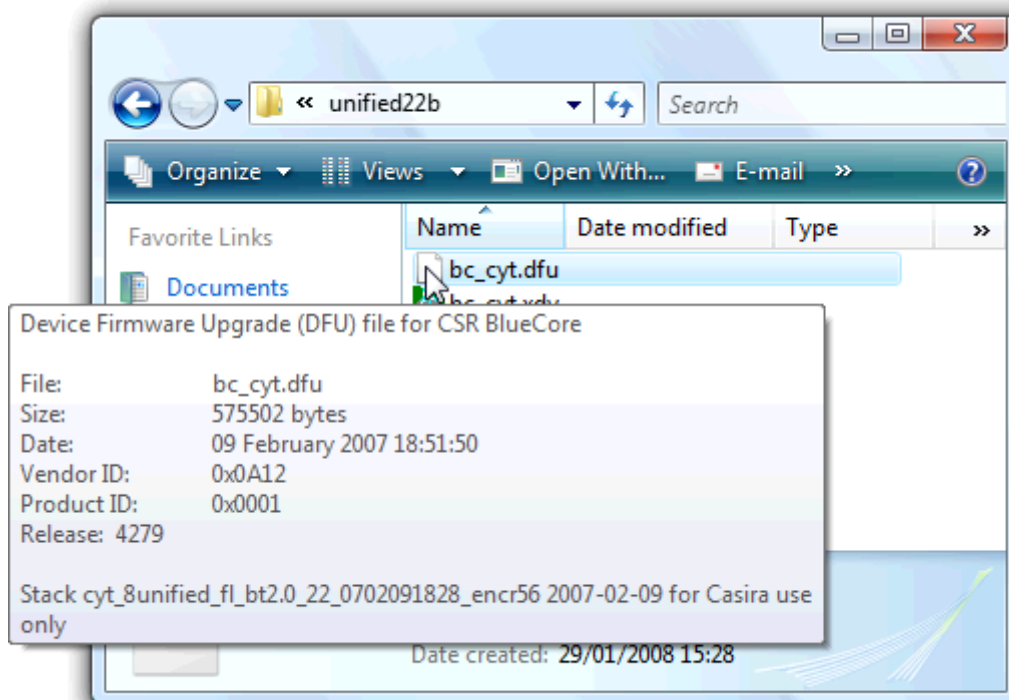


Figure 7.2: A DFU ToolTip

### 7.14 DFU Tools

DFU Tools are a suite of command-line tools to let your sign and combine firmware and persistent store files to form DFU files.

The DFU Tools documentation can be accessed from the Start Menu. Open the **CSR BlueSuite** item, and click **CSR DFU Tools Documentation**.



## 8 TrueTest Toolkit

The TrueTest toolkit consists of the necessary libraries and documentation to let you develop applications in a variety of languages for production programming and testing of BlueCore enabled devices.

The toolkit is designed for use in production test systems. Instead of writing test scripts to call other BlueSuite programs, you can write a test program that accesses the test APIs (*Application Programming Interface*) directly.

See the *TrueTest Toolkit Introduction*.

# Regulatory Notes & Information

This device complies with Part 15 of the FCC Rules.

Operation is subject to the two following conditions :-

- (1) This device may not cause harmful interference, and
- (2) This device must accept any interference received, including interference that may cause undesired operation.

Changes or modifications to the transmitter not expressly approved by CSR could void the authority ability to operate the equipment.

## **User Label Warning !**

The transmitter is in compliance with the Part 15 FCC Rules. A label which demonstrates compliance with the FCC is located on the bottom of the laptop.  
The label contains the following information :-

Contains FCC ID:- PIWBSMAN

## **Transmitter Warning !**

Installers of the Bluetooth transmitter should note that the Bluetooth transmitter should not be co-located with any other transmitter, for example WLAN interfaces in the lap top. The distance between the Bluetooth transmitter and antennas of other radio equipment must be at least 20cm in distance away from each other. Any closer installations require additional certification filings to examine co-location issues.

## 9 Document References

Document	Reference
<i>Specification of the Bluetooth System</i>	Version 2.1 + EDR, Core Package, 26 July 2007 <a href="http://www.bluetooth.com">www.bluetooth.com</a>
BlueCore Class 2 Example Designs	<a href="http://www.csrsupport.com">www.csrsupport.com</a>
BlueCore Production Information Data Books	<a href="http://www.csrsupport.com">www.csrsupport.com</a>
<i>BlueCore Device Firmware Upgrade (DFU) Overview</i>	CS-112861-AN
<i>USB Device Firmware Upgrade Specification</i>	Part of the USB Specification <a href="http://www.usb.org/developers/devclass_docs/DFU_1.1.pdf">http://www.usb.org/developers/devclass_docs/DFU_1.1.pdf</a>
<i>Device Firmware Upgrade Protocol Specification</i>	CS-102656-SP
<i>Device Firmware Upgrade File Format Specification</i>	CS-113166-AN
<i>PSTool User Guide</i>	CS-101505-UG
<i>BlueTest User Guide</i>	CS-102736-AN
<i>Operating the BlueCore5-FM Radio</i>	CS-113378-AN
<i>USB-SPI Converter User Guide</i>	CS-114776-UG
<i>TrueTest Toolkit Introduction</i>	CS-101531-AN
<i>Casira User Guide</i>	CS-102077-UG

## Terms and Definitions

Term	Definition
API	Application Programming Interface
BlueCore®	Group term for CSR's range of Bluetooth wireless technology ICs
Bluetooth®	Set of technologies providing audio and data transfer over short-range radio connections
BTCLI	Bluetooth Command Line Interface
CD ROM	Compact Disk Read Only Memory
CSR	Cambridge Silicon Radio
DFU	Device Firmware Upgrade
DLL	Dynamically Linked Library
HCI	Host Controller Interface
IC	Integrated Circuit
i.e.	<i>Id est</i> , that is
INF	Setup Information File, often used for Windows device drivers
Persistent Store	Store of configuration values in non-volatile memory
PID	Product ID used to identify USB devices
PsCli	Persistent Store Command Line Interface
RAM	Random Access Memory
RfCli	RFCOMM Command Line Interface
RFCOMM	Protocol layer providing serial port emulation over L2CAP; element of Bluetooth
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VID	Vendor ID used to identify USB devices