



Push every boundary.™

BlueCore®



# PSTool

## User Guide

### Issue 7

## Document History

Revision	Date	History
a	21 AUG 06	Original publication of this document
b	27 AUG 09	Updated to include a description of the filter field.
3	30 OCT 09	Updated to latest style guidelines. Document references updated.
4	03 AUG 10	Update for Bluetooth SDKs 2010 and remove BlueLab references.
5	09 MAR 12	Updated to latest Style guidelines.
6	09 MAY 12	Minor editorial correction
7	13 DEC 13	Updated to latest style for ADK 3.0

## Contacts

[General information](#)  
[Information on this product](#)  
[Customer support for this product](#)  
[More detail on compliance and standards](#)  
[Help with this document](#)

[www.csr.com](http://www.csr.com)  
[sales@csr.com](mailto:sales@csr.com)  
[www.csrsupport.com](http://www.csrsupport.com)  
[product.compliance@csr.com](mailto:product.compliance@csr.com)  
[comments@csr.com](mailto:comments@csr.com)

## Trademarks, Patents and Licences

Unless otherwise stated, words and logos marked with <sup>TM</sup> 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](http://www.csrsupport.com) for compliance and conformance to standards information.

## Contents

Document History .....	2
Contacts.....	2
Trademarks, Patents and Licences .....	3
Life Support Policy and Use in Safety-critical Compliance.....	3
Performance and Conformance .....	3
Contents .....	4
1. Introduction.....	5
1.1. PSTool.....	5
2. Opening the PSTool Application.....	6
2.1. Transport Options.....	6
3. Description of PSTool Main Window.....	11
3.1. General.....	11
4. Description of PSTool Functionality.....	12
4.1. File Menu.....	12
4.2. Entry Menu .....	14
4.3. Stores Menu .....	15
4.4. View Menu.....	16
4.5. Factory Menu.....	16
4.6. Help Menu .....	16
4.7. Filter Field.....	16
4.8. Button Functions.....	16
5. Troubleshooting.....	17
6. Technical Support.....	18
Document References .....	19
Terms and Definitions.....	20

## 1. Introduction

This document describes the PSTool utility provided with CSR Bluetooth SDKs and BlueSuite.

### 1.1. PSTool

PSTool is an editing tool that allows you to read and modify the Persistent Store of BlueCore ICs. The Persistent Store consists of key-value pairs 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 of the keys must be calibrated for each individual chip.

#### **Important Note:**

Do not modify Persistent Store values unless you have a clear understanding of what they do and the affect that changing the value will have. Careless changes to the Persistent Store can prevent the firmware from executing correctly or can stop BlueCore from communicating with the PC. In some cases, it may not be possible to restore the BlueCore device to the state in which it was supplied.

See CSR documentation on PS Key settings, supplied with individual Firmware releases, before editing values.

Perform a backup using **BlueFlash** or **E2Util** as appropriate before editing PS Keys.

## 2. Opening the PSTool Application

To open the PSTool utility:

1. Browse to the location of the PSTool executable:

**C:\<SDK-Name>\_version\tools\bin\PSTool.exe**

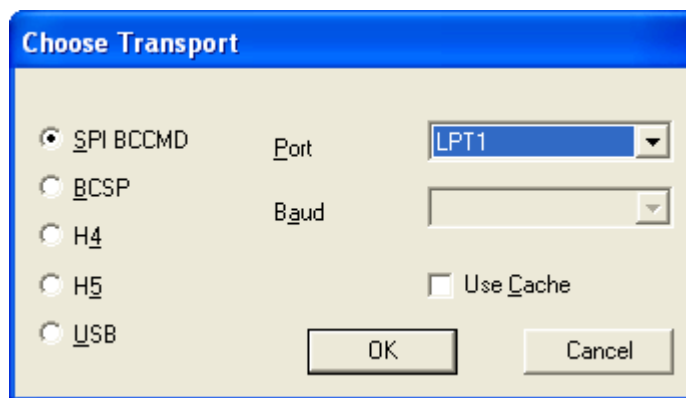
2. Double-click `PSTool.exe`.

Alternatively:

Select **PSTool** from the **Start** menu:

**Start\Programs\<SDK-Name>\Tools\PSTool**

3. The **Choose Transport** window opens:



The transport options are described in section 2.1.

4. Select the appropriate options for the required transport.
5. Click **OK**.

The PSTool main window appears.

### 2.1. Transport Options

PSTool initially opens a **Choose Transport** window that allows selection of the interface used to access the Persistent Store (PS).

The options available are:

- SPI BCCMD
- BCSP
- H4
- H5
- USB

These options are described in sections 2.1.1 and 2.1.2.

### 2.1.1. SPI BCCMD

SPI BCCMD uses the BlueCore Command (BCCMD) Protocol over the Serial Peripheral Interface layer. It is a relatively robust transport that also provides some error detection.

To make the connection using this protocol:

**Note:**

To connect using SPI, an SPI cable connection between the PC printer port and the SPI connection on the development board is required.

1. Select the SPI BCCMD Radio button.
2. Select a printer port (LPT1 to LPT20) from the Port drop-down list.
3. Check the **Halt Chip on SPI access** option if this is the required behaviour.

**Note:**

This option is only required when the device is running v15 firmware.

4. Click **OK**.

**Note:**

SPI BCCMD is the recommended transport protocol when editing the Persistent Store, since it does not depend on any particular Host Interface.

## 2.1.2. Connecting PSTool Using Other Transport Interfaces

Before selecting other transport interfaces in the **Choose Transport** window, certain PS Keys must be set to the appropriate values using an SPI connection.

The alternative transport interfaces available and their configuration is briefly described in the rest of this section.

### 2.1.2.1. BCSP

To connect PSTools using the BCSP protocol:

1. Using an SPI connection set the following PS Keys:
  - 1.1. PSKEY\_HOST\_INTERFACE: Select **UART link running BCSP** from drop-down list and click **Set**.
  - 1.2. PSKEY\_UART\_CONFIG\_BCSP: Check this key is set to 0x0806.

**Note:**

These two PS Keys can be set using the canned operation **@bcsp**.

- 1.3. PSKEY\_UART\_BAUDRATE: Select the preferred Baud rate (bits/s) from the drop-down list and click **Set** (this value is required later when connecting using the BSCP option). Unless your PC has a fast UART, speeds above 115Kbps may not work.
2. Click **Reset**.

**Note:**

Ensure there is cable connection from a COM port on the PC to the BlueCore development board.

3. Click **Reconnect**.
4. In the **Choose Transport** window:
  - 4.1. Select the BCSP Radio button.
  - 4.2. Select a COM Port (COM 1 to COM 50) from the drop-down list.
  - 4.3. Select the required Baud (bits/s) rate (this must match that set on the BlueCore IC in step 1.2).
  - 4.4. Click **OK**.

BCSP uses BlueCore Serial Port protocol over the UART interface.



#### 2.1.2.2. H4

To connect PSTools using H4 protocol:

1. Using an SPI connection set the following PS Keys:
  - 1.1. PSKEY\_HOST\_INTERFACE: Select **UART link running H4** from drop down list and click **Set**.
  - 1.2. PSKEY\_UART\_CONFIG\_H4: Check this key is set to 0x08a8.

**Note:**

These two PS Keys can be set using the canned operation **@h4**.

- 1.3. PSKEY\_UART\_BAUDRATE: Select the preferred Baud (bits/s) rate from the drop down list and click **Set** (this value is required later when connecting using the BSCP option). Unless your PC has a fast UART, speeds above 115Kbps may prove unreliable.
2. Click **Reset**.
3. Ensure there is cable connection from a COM port on the PC to the BlueCore development board.
4. Click **Reconnect**.
5. Reopen PSTool and in the Choose Transport window:
  - 5.1. Select the H4 Radio button.
  - 5.2. Select a COM Port (COM 1 to COM 50) from the drop down list.
  - 5.3. Select the required Baud rate (this must match that set on the BlueCore IC in step 1.3).
  - 5.4. Click **OK**.

H4 is a Host Control Interface (HCI) five-wire UART transport protocol. It is the simplest of the Bluetooth standard host transports and has the advantage of simplicity and speed but this is generally outweighed by the disadvantage of poor error detection.

#### 2.1.2.3. H5

To connect PSTools using H5 protocol:

1. Using an SPI connection set the following PS Keys:
  - 1.1. PSKEY\_HOST\_INTERFACE: Select **UART link running H5** from drop-down list and click **Set**.
  - 1.2. PSKEY\_UART\_CONFIG\_H5: Check this key is set to 0x1806.

**Note:**

These two PS Keys can be set using the canned operation **@h5**.

- 1.3. PSKEY\_UART\_BAUDRATE: Select the preferred Baud (bits/s) rate from the drop down list and click **Set** (this value is required later when connecting using the BSCP option).
2. Click **Reset**.
3. Ensure there is cable connection from a COM port on the PC to the BlueCore development board.
4. Click **Reconnect**.
5. Reopen PSTool and in the Choose Transport window:
  - 5.1. Select the H5 radio button.
  - 5.2. Select a COM Port (COM 1 to COM 50) from the drop-down list.
  - 5.3. Select the required Baud rate (this must match that set on the BlueCore IC in step 1.3).
  - 5.4. Click **OK**.

H5 is an HCI three-wire UART transport protocol which provides error detection and correction.

#### 2.1.2.4. USB

To connect PSTools using USB protocol:

1. Using an SPI connection set the following PS Keys:
  - 1.1. PSKEY\_HOST\_INTERFACE select **USB link** from drop-down list and click **Set**.
  - 1.2. PSKEY\_USB\_VM\_CONTROL set this key to False.

**Note:**

These two PS Keys can be set using the canned operation **@usb**.

2. Click **Reset**.
3. Ensure there is a USB cable connection between the PC and the BlueCore development board.
4. Click **Reconnect**.
5. Reopen PSTool and in the Choose Transport window:
  - 5.1. Select the USB Radio button.
  - 5.2. Select the device from the drop down list of USB devices.
  - 5.3. Click **OK**.

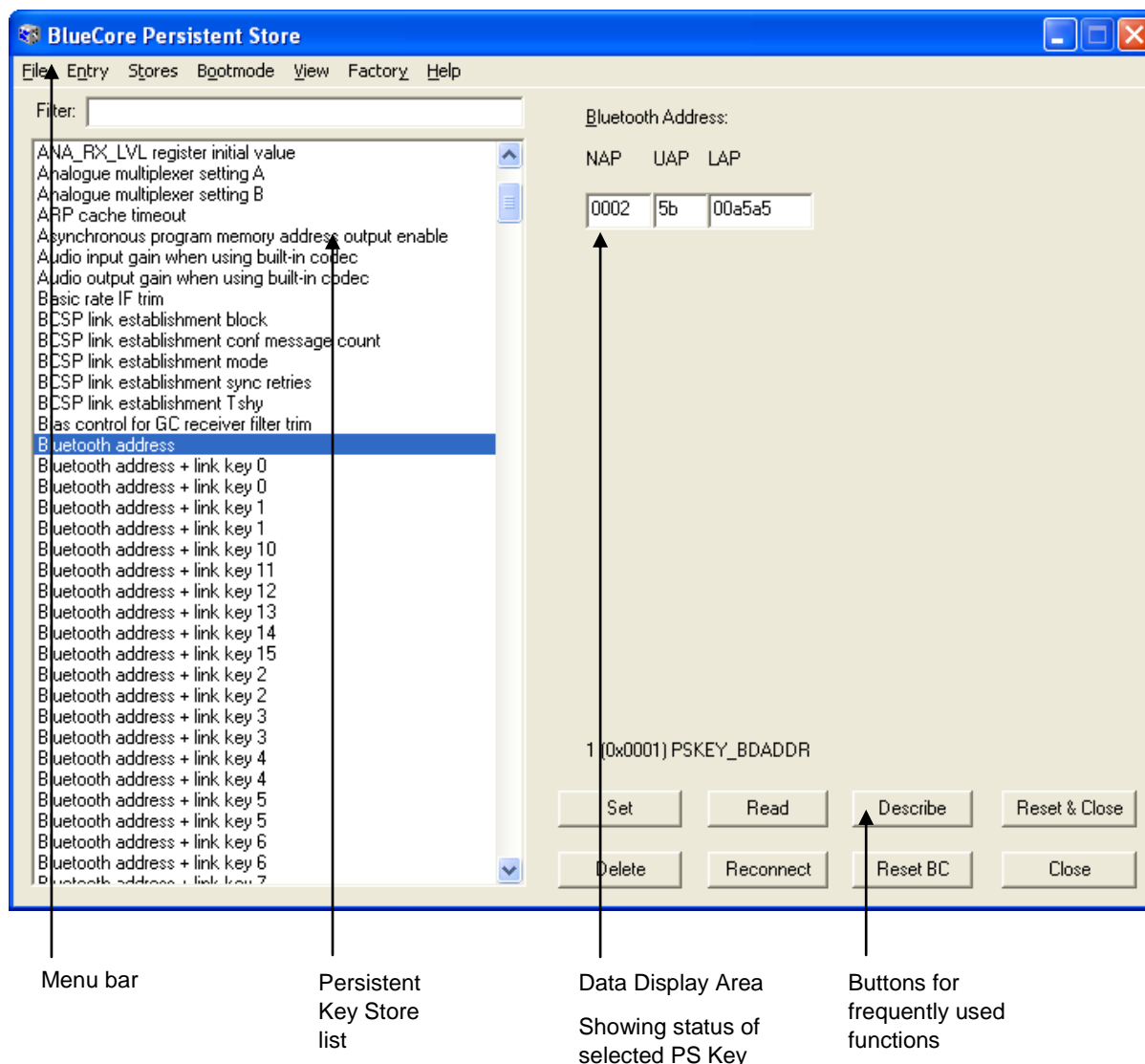
PSTool connects to the module using USB.

**Note:**

When a USB cable is connected to a Casira unit, the RS232 port does not work.

## 3. Description of PSTool Main Window

When the transport selection has been completed the PSTool main window appears:



### 3.1. General

The PSTool main window provides the main user interaction with the PS Key values.

The menus in the menu bar provide access to the full range of PSTool functionality, see section 4.

The Data Display Area displays the value of the currently selected PS Key in a user-friendly format. Here developers can read the existing value for a selected PS Key and amend it if required.

#### Note:

PS Key values displayed in this area are presented in a user-friendly format. Where appropriate drop-down lists of alternative settings for the PS Key are provided to simplify setting alternative values otherwise values can be changed using the keyboard.

To save changes made to the value of a PS Key click **Set** before selecting another PS Key.

The buttons displayed on the main PSTool window are provided for convenience. They perform the same function as frequently used functions that are otherwise accessed from menus in the menu bar.

## 4. Description of PSTool Functionality

### 4.1. File Menu

The File menu consists of four menu items:

- Merge
- Run Query
- Dump
- Exit

#### 4.1.1. Merge

This facility sets PS Keys to those contained in a previously saved `.psr` file (`.psr` files are created by the dump and query processes process described in the following sections).

To use a `.psr` file to set the values of PS Keys:

1. Select **Merge** from the **File** menu.  
  
An **Open Persistent Store File** window appears.
2. Browse to the location of the `.psr` you want to merge.
3. Select the required file and click **Open**.

The window closes and the PS Key values in the `.psr` file are downloaded to the BlueCore IC.

#### 4.1.2. Run Query

This option allows you to run a simple query to read the value of specific sets of PS Keys. To do this the user must prepare a `.psq` file to list the PS Keys to be read.

##### 4.1.2.1. `.psq` Files

`.psq` files can be written using a plain text editor such as notepad. The syntax for a `.psq` files is in the form:

```
?1234= // read and dump the value of PS key "1234".
?1234~ 5678 9adc // read and dump value of PS key "1234" if it differs from the data specified.
```

##### 4.1.2.2. Using the Run Query Facility

1. Select **Run Query** from the **File** menu.  
  
A **Run PS Query File** window appears.
2. Browse to the location of the prepared `.psq` file.
3. Select the required `.psq` file and click **Open**.  
  
A **Save Persistent Store File As** window appears.
4. Enter a name for the query results file and click **Save**.  
  
The file is saved as a `.psr` file in the selected location and the window closes.
5. The file can then be opened from its saved location. The file contains the values dumped as a result of the query.

### 4.1.3. Dump

This facility allows a copy of the PS Keys as set on the BlueCore module to be saved as a `.psr` file, this can be used to reset these values using the **Merge** facility described in section 4.1.1.

To save a dump of a modules PS Key values:

1. Select **Dump** from the **File** menu.  
  
A **Save Persistent Store File As** window:
2. Browse to the location in which you wish to save the `.psr` file.
3. Enter a name for the file.
4. Click **Save**.

The file is saved to the selected location.

### 4.1.4. Exit

This option closes the PSTool application.

## 4.2. Entry Menu

The **Entry** menu provides a range of options and facilities:

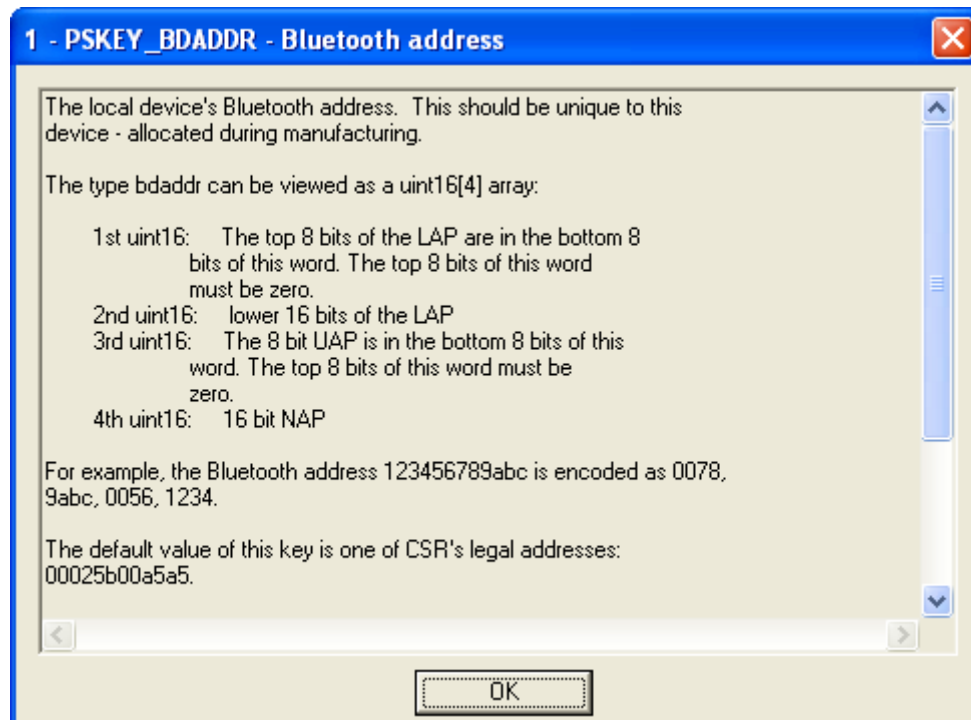
- **Read Entry:** reads the value of the PS key selected in the Persistent Key Store list and displays the value in the status area. This menu item performs the same function as the **Read** button.

**Note:**

A read is automatically performed when a key is selected from the Persistent Store list.

- **Set Entry:** this option writes the value displayed in the status area to the selected PS Key. This menu item performs the same function as the **Set** button.
- **Delete Entry:** this option removes the selected PS Key. If required, the stack creates the key using a default value the next time the stack reboots. Otherwise the key displays **Not Present** when selected in the list. This menu item performs the same function as the **Delete** button.
- **Describe:** this option displays a brief description of the currently selected PS Key in a separate window.

For example:



This menu item performs the same function as the **Describe** button.

- **Edit raw:** this allows the user to edit the raw words of the selected PS Key value rather than the user-friendly presentation in the data display area of the main PSTool window.
- **Canned Operations:** this option allows the user to select from a list of predefined operations. These write individual keys or combination of keys to the required values for the selected option.

For example selecting **@bcsp** writes values to two keys PSKEY\_HOST\_INTERFACE (to UART link running BCSP) and PSKEY\_UART\_CONFIG\_BCSP (the UART configuration key for BCSP) and sets them to the required values to allow BCSP transport communication with the host.

## 4.3. Stores Menu

This menu allows the user to select which of the various layers available to the Persistent Store that PSTool will interact with.

### Note:

During development the layer storing the PS Key value can be largely ignored. The distribution between stores only becomes significant during production (some PS Keys are set during module production, some later, it is at this point that the layer used to store the PS Key value becomes more significant).

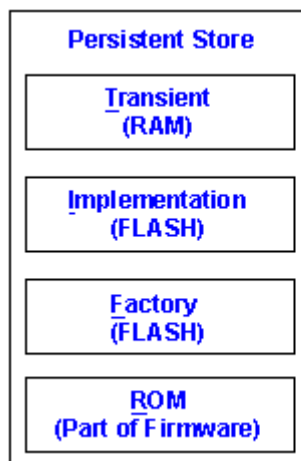
The options are:

- All (TIFR)
- Implementation Only (I)
- ROM Only (R)
- RAM Only (T)
- Factory Only (F)
- Not RAM (IFR)

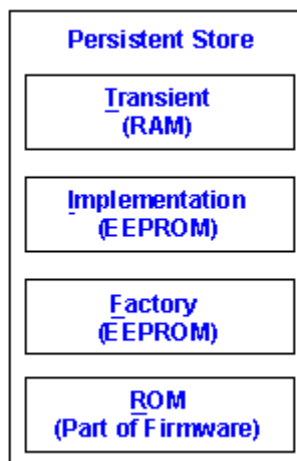
The architecture of the various stores can be represented as shown below:

Persistent Store Architecture for:

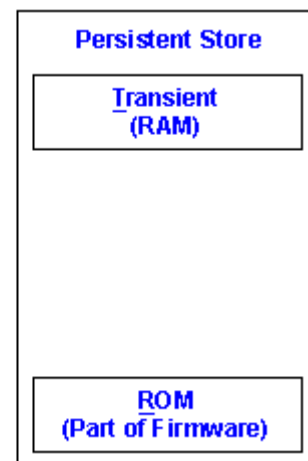
Flash IC



ROM IC with EEPROM



ROM IC without EEPROM



Default values of PS Keys defined by the firmware are stored in ROM.

Keys that are set during product manufacture are generally set in the Factory and Implementation area, while keys that are set during runtime as a result of user interaction with the product are generally stored in Transient.

At runtime the value stored at the highest level is used.

### Note:

If an attempt is made to write the same value for a PS Key in the layer immediately above the layer in which the value is currently stored, the write command is aborted.

If Flash is not identified on the IC and there is no E2 supported in the firmware, writes are made in the Transient layer by default.

ROM ICs that use an EEPROM device to store PS Keys require the EEPROM to be initialised before PSTools can read or write to it. The amount of space appropriate for the store is dependent on the BlueCore ROM variant. See Selection of I2C EEPROMS for use with BlueCore.

## 4.4. View Menu

The **View** menu provides options that control the way PS Keys are displayed in the Persistent Key Store list.

The list can be displayed using either their Friendly names or by Programmer IDs and can be ordered alphabetically (Sort by name) or by number (Sort by ordinal).

## 4.5. Factory Menu

The **Factory** menu contains:

- Factory Set: when a PS Key has a value currently set in the Implementation level selecting **Factory Set** applies the setting at the Factory level.
- Factory Restore: if the selected PS Key is set in the Factory level this option removes any values set at the higher levels.
- Factory Set All: selecting this option removes any values set in the higher levels for the full range of PS keys for which values have been set in the Factory level.

## 4.6. Help Menu

The **Help** menu contains:

- Firmware version: Reports the version of firmware currently running on the BlueCore IC.
- Describe key: See section 4.2.
- About: Opens a separate window displaying version information for PSTool.

## 4.7. Filter Field

Typing characters in this field with filter will filter out any PS Keys that do not contain the character or sequence of characters. This can be used to help locate specific PS Keys quickly.

## 4.8. Button Functions

The PSTool main window contains the following buttons:

- Set: Click to save the value displayed/entered in the data display area to the selected PS Key.
- Read: Click to get the Persistent Store value for the selected key which is displayed in the data display area.
- Describe: See section 4.2.
- Delete: Click to remove the selected PS Key from the Persistent Store.

### Note:

If required, the stack creates the PS Key using a default value the next time the stack reboots. Otherwise, the key continues to be reported in the data status area as **Not present**.

- Reconnect: Click to bring up the **Choose Transport** window allowing the user to reconnect after a device reset or if the transport was disrupted for some reason.
- Reset BC: Click to reset the BlueCore device.

### Note:

The device must be Reset and the stack rebooted before changes made to PS Keys using PSTool take effect.

- Close: Click to close the PSTool application.

### Note:

Changes to PS Keys are not written unless the **Set** button is clicked before closing.



## 5. Troubleshooting

The most common fault experienced when using PSTool is a communication failure with the BlueCore IC.

### Note:

A failure to communicate with the chip is indicated when PSTool reports ENTRY NOT PRESENT – Click to add for a PS Key that is known to store a valid entry e.g. Bluetooth Address.

If this occurs there are a few possible causes, which can be readily diagnosed and corrected if necessary:

1. Reset the chip and reconnect PSTool.
2. Check that **All (TIFR)** is selected in the **Stores** menu.
3. Check that the SPI cable is correctly connected and the device is powered.

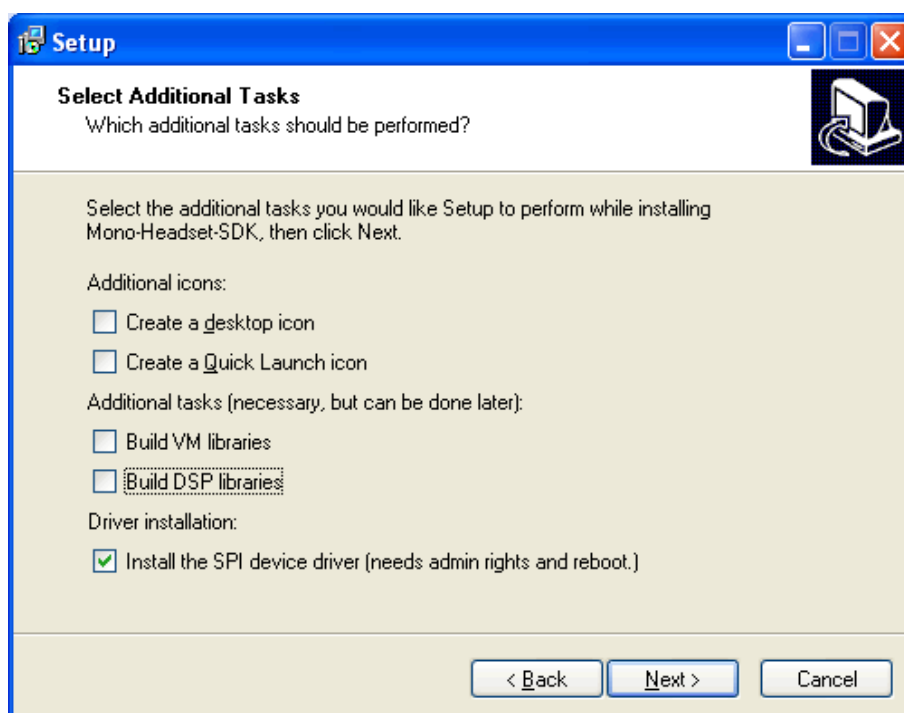
If PSTool still fails to communicate with the chip, two other possible causes are commonly found to be the source of the problem:

4. Cause: There is no firmware running on the device or the firmware is not suitable for the BlueCore variant being used.

Solution: Download compatible firmware using BlueFlash.

5. Cause: The SPI device driver was not included during installation of the SDK.

Solution: Install the SPI device driver by re-running SDK executable file ensuring the option to **Install SPI device driver** is selected in the **Select Additional Tasks** window:



### Note:

The **Install the SPI device driver** option is selected by default during installation.

If you only want to install the SPI device driver uncheck the other options before clicking **Next** to continue.

## 6. Technical Support

Further information on all CSR products can be found on the technical support website (<http://www.csrsupport.com>).

Developers are also recommended to view the public newsgroups hosted by CSR on the Internet news (NTTP) server [news.csr.com](http://news.csr.com). The newsgroups are a convenient forum for the Bluetooth community to exchange knowledge and are a valuable source of information.

For set up instructions and guidelines for the use of newsgroups follow the links on the CSR support website.

## Document References

Document	Reference
<i>Selection of I2C EEPROMS for use with BlueCore</i>	CS-101518-AN

## Terms and Definitions

BCCMD	BlueCore Command
BCSP	BlueCore Serial Protocol
BlueCore®	Group term for CSR's range of Bluetooth wireless technology ICs
Bluetooth SIG	Bluetooth Special Interest Group
Bluetooth®	Set of technologies providing audio and data transfer over short-range radio connections
COM	Communication
CSR	Cambridge Silicon Radio
e.g.	<i>exempli gratia</i> , for example
etc	<i>et cetera</i> , and the rest, and so forth
H4	Bluetooth defined UART transport
H5	Three-wire UART transport
HCI	Host Communication Protocol
IC	Integrated Circuit
i.e.	Id est, that is
LPT	Line Printer Terminal
PC	Personal Computer
PS	Persistent Store
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus