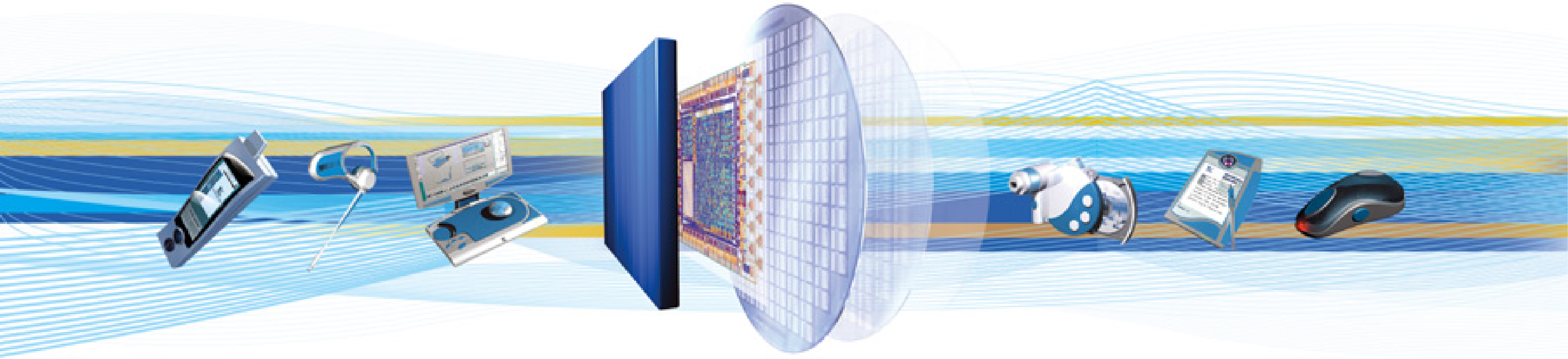
CSR Synergy Bluetooth 18.2.0

OPP Obex Push Profile

Demo Description

November 2011



Cambridge Silicon Radio Limited

Churchill House
Cambridge Business Park
Cowley Road
Cambridge CB4 0WZ
United Kingdom

Registered in England and Wales 3665875

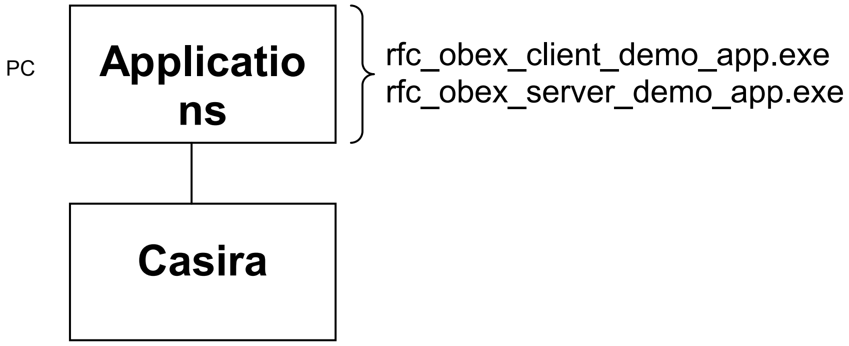
Tel: +44 (0)1223 692000
Fax: +44 (0)1223 692001
www.csr.com

Contents

1 OBEX PUSH (OPP) PROFILE

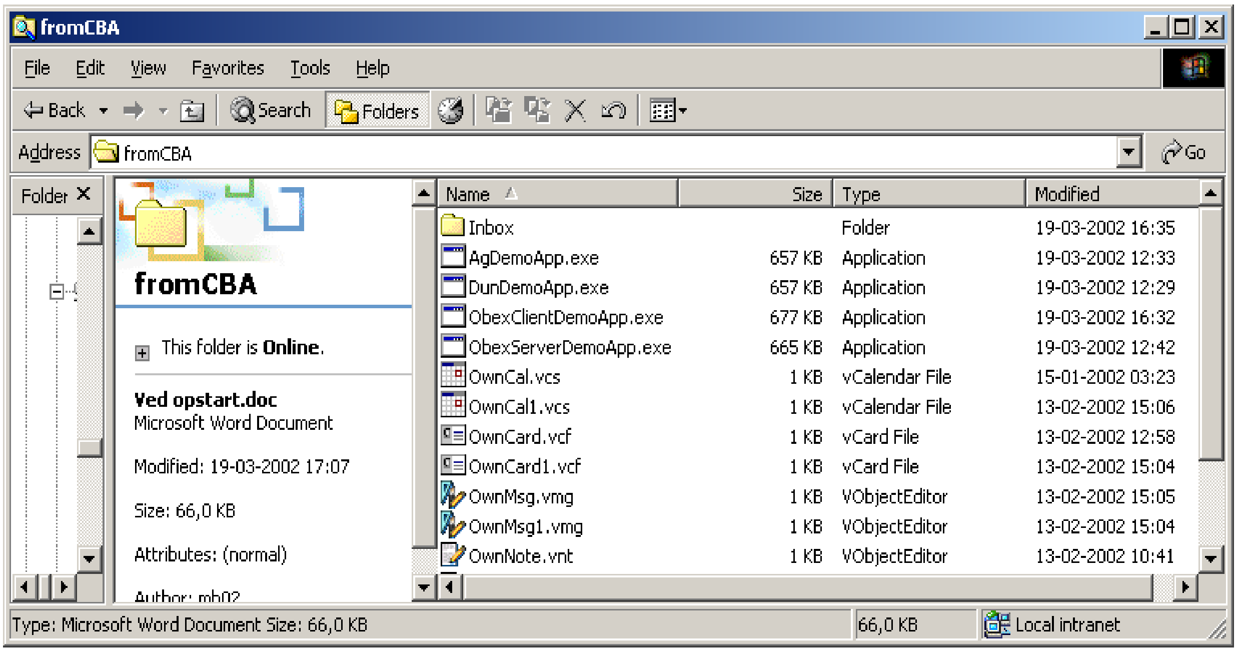
1.1 Generally

The OBEX programs can be used for object transfer. This demo is running with a CASIRA with RFCOMM-build firmware.



Before starting the programs it is necessary to create an Inbox folder in the same directory where the programs are for storing of input. The program directory has to contain two of each of the following object types, named as indicated on the below figure, which are used when another user pull objects from you, see below illustration:

- vCard
- vCal
- vMsg
- vNote



fromCBA

This folder is **Online**.

Ved opstart.doc
Microsoft Word Document

Modified: 19-03-2002 17:07

Size: 66,0 KB

Attributes: (normal)

Author: mb02

Name	Size	Type	Modified
Inbox		Folder	19-03-2002 16:35
AgDemoApp.exe	657 KB	Application	19-03-2002 12:33
DunDemoApp.exe	657 KB	Application	19-03-2002 12:29
ObexClientDemoApp.exe	677 KB	Application	19-03-2002 16:32
ObexServerDemoApp.exe	665 KB	Application	19-03-2002 12:42
OwnCal.vcs	1 KB	vCalendar File	15-01-2002 03:23
OwnCal1.vcs	1 KB	vCalendar File	13-02-2002 15:06
OwnCard.vcf	1 KB	vCard File	13-02-2002 12:58
OwnCard1.vcf	1 KB	vCard File	13-02-2002 15:04
OwnMsg.vmg	1 KB	VObjectEditor	13-02-2002 15:05
OwnMsg1.vmg	1 KB	VObjectEditor	13-02-2002 15:04
OwnNote.vnt	1 KB	VObjectEditor	13-02-2002 10:41

Type: Microsoft Word Document Size: 66,0 KB | 66,0 KB | Local intranet

In case 2 of each object types are not present the program will terminate

The scenarios covered by this profile are the following:

- Usage of a Bluetooth device e.g. a mobile phone to push an object to the inbox of another Bluetooth device. The object can e.g. be a business card or an appointment
- Usage of a Bluetooth device e.g. a mobile phone to pull a business card from another Bluetooth device
- Usage of a Bluetooth device e.g. a mobile phone to exchange business cards with another Bluetooth device. An exchange defines as a push of a business card followed by a pull of a business card

The application has been made to run on Windows and Linux and may be connected to the Casira using either a serial connection using BCSP (`rfc_obex_server_demo_app.exe`), a serial connection using H4DS (`rfc_obex_server_demo_app_h4ds.exe`) or an USB connection (`rfc_obex_server_demo_app_usb.exe`), and their HCI equivalent.

The description below is based on the Windows demo application but the description also holds for the Linux Demo Application.

1.2 OBEX Server

Use of program rfc_obex_server_demo_app.exe

Note 1: This description is for CSR Synergy Bluetooth RFCOMM. The functionality of the application for the HCI build is identical. The only difference is the naming: rfc_obex_server_demo_app.exe versus hci_obex_server_demo_app.exe.

Start the program rfc_obex_server_demo_app.,The program has the following options.

1. Choose the COM port on which the Bluetooth module is connected, by specifying the –C parameter to the program, e.g. rfc_obex_server_demo_app –C COM1. At start up COM1 is selected as default. (On Linux the default port is /dev/ttyS0)

2. Choose baudrate for the COM port on which the Bluetooth module is connected, by specifying the –B parameter to the program, e.g. rfc_obex_server_demo_app –B 115200. If no parameter is specified the default is 115200.

3. Choose a specific device for default connection by specifying the –a parameter to the program, e.g. rfc_obex_server_demo_app –a 0002:5b:01a494. If no address is specified it is necessary to perform a search for servers in order to establish a connection. This parameter is optional.

A server waits for the client to connect and either pushes or pulls objects, i.e. all activity takes place from the client side. However, to save an object or type in pin-code can be asked for.

Examples:

- The client opens a Bluetooth device and asks to e.g. ‘receive’
- The client can e.g. ask for a default vCard (Pull)
- The client can push different objects to the server, where they are stored in the Inbox (the following is supported by ‘Push’: business card, calendar, notes and messages)
- From a client program a vCard can be obtained or exchanged

1.3 OBEX Client

Use of program rfc_obex_client_demo_app.exe

Note 2: This description is for CSR Synergy Bluetooth RFCOMM. The functionality of the application for the HCI build is identical. The only difference is the naming: rfc_obex_client_demo_app.exe versus hci_obex_client_demo_app.exe.

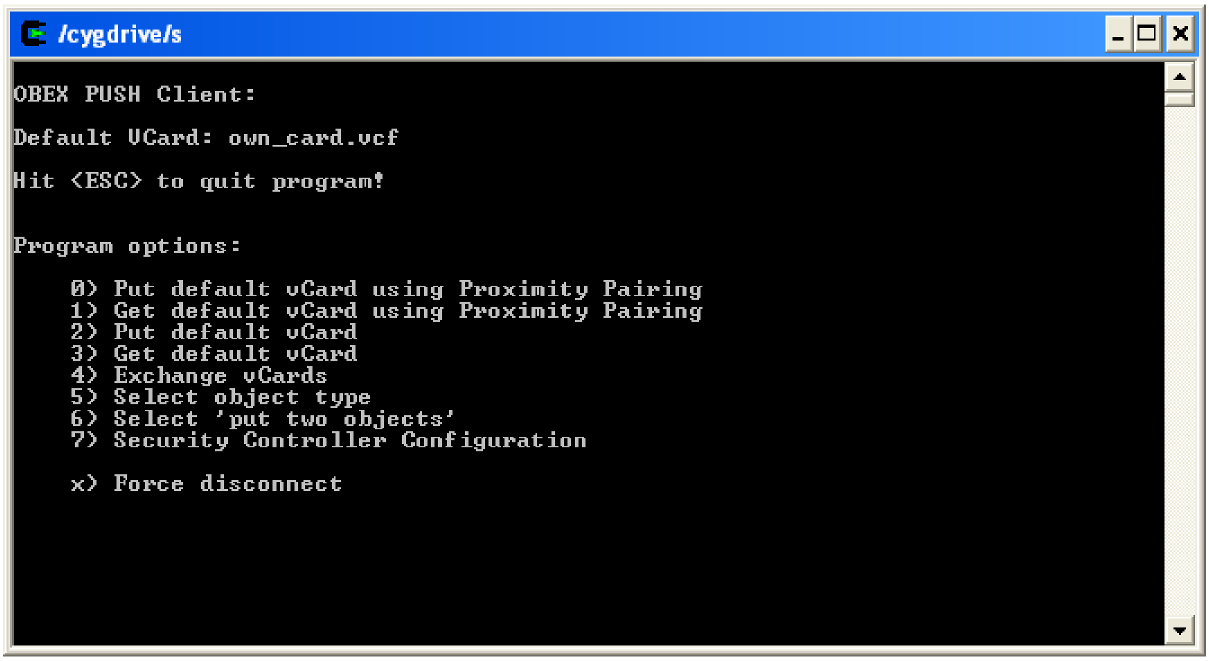
For use of this program a server side (or a similar e.g. Widcomm PC SW) is required.

Start the program rfc_obex_client_demo_app. The program has the following options.

1. Choose the COM port on which the Bluetooth module is connected, by specifying the –C parameter to the program, e.g. rfc_obex_client_demo_app –C COM1. At start up COM1 is selected as default.
2. Choose baudrate for the COM port on which the Bluetooth module is connected, by specifying the –B parameter to the program, e.g. rfc_obex_client_demo_app –B 115200. If no parameter is specified the default is 115200.

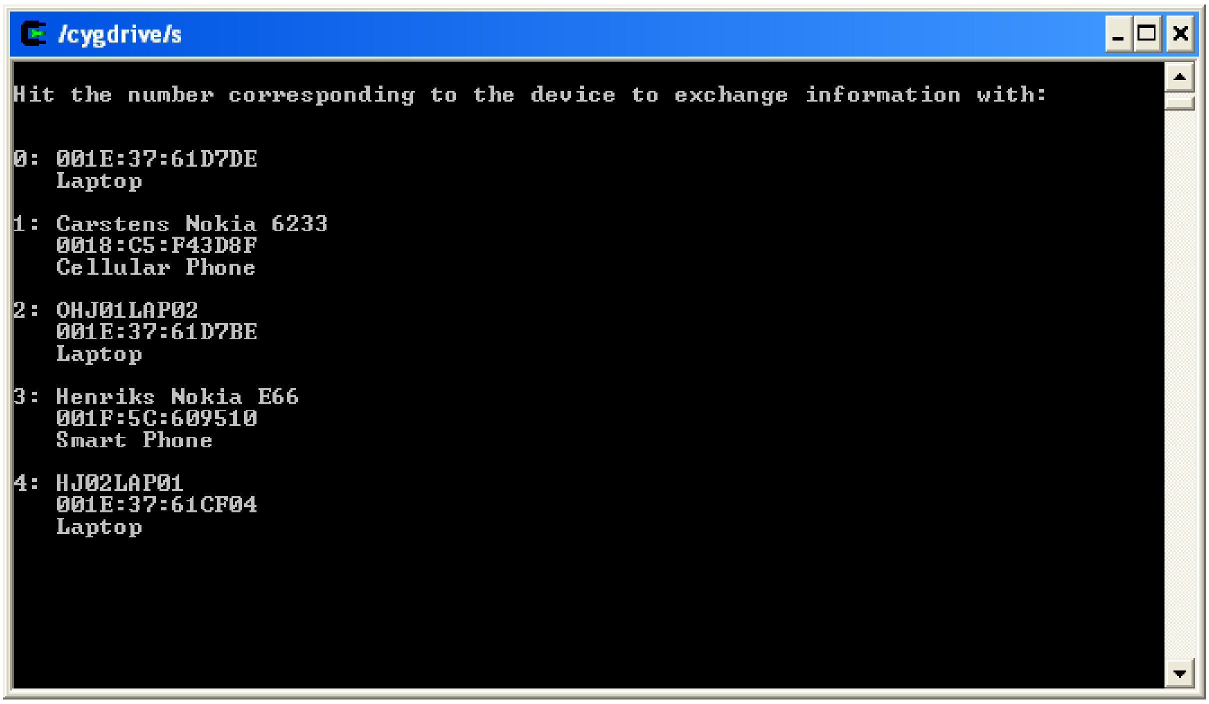
3. Choose a specific device for default connection by specifying the –a parameter to the program, e.g. rfc_obex_client_demo_app –a 0002:5b:01a494. If no address is specified it is necessary to perform a search for servers in order to establish a connection

At start up, the following possibilities are available, see below illustration:



```
OBEX PUSH Client:

Default VCard: own_card.vcf

Hit <ESC> to quit program!


Program options:

    0) Put default vCard using Proximity Pairing
    1) Get default vCard using Proximity Pairing
    2) Put default vCard
    3) Get default vCard
    4) Exchange vCards
    5) Select object type
    6) Select 'put two objects'
    7) Security Controller Configuration

    x) Force disconnect
```

When choosing e.g. ‘2’ a search for the devices to put to is activated, see below illustration. If the ‘-A’ command line parameter was specified during application start, the specified address will be added to the list of discovered devices with the name “Address from command line”



```
Hit the number corresponding to the device to exchange information with:


0: 001E:37:61D7DE
   Laptop

1: Carstens Nokia 6233
   0018:C5:F43D8F
   Cellular Phone

2: OHJ01LAP02
   001E:37:61D7BE
   Laptop

3: Henriks Nokia E66
   001F:5C:609510
   Smart Phone

4: HJ02LAP01
   001E:37:61CF04
   Laptop
```

Choose e.g. device ‘0’ and a vCard will be put to the laptop with address 001E:37:61D7DE.

In case ESC is chosen for closing the program, a list is updated/saved covering the names of the devices that were connected to, for use next time the program is opened.

If closing the program by using X, the list will not be updated.

2 Linux

This section describes how to build and run the OPC/OPS demo applications on Linux.

The OPC/OPS demo applications (pure user space), located in `./applications/obex_push`, may be compiled on Linux by means of:

> make clean all TARGET_ARCH=Linux-2.6-x86

This will output six files for the OPC demo:

1. `hci_push_client_demo_app`,
2. `hci_push_client_demo_app_h4ds`, and
3. `hci_push_client_demo_app_usb`, for serial and USB communication using a HCI split.
4. `rfc_push_client_demo_app`,
5. `rfc_push_client_demo_app_h4ds`, and
6. `rfc_push_client_demo_app_usb`, for serial and USB communication using a RFC split.

and six files for the OPS demo:

7. `hci_push_server_demo_app`,
8. `hci_push_server_demo_app_h4ds`, and
9. `hci_push_server_demo_app_usb`, for serial and USB communication using a HCI split.
10. `rfc_push_server_demo_app`,
11. `rfc_push_server_demo_app_h4ds` and
12. `rfc_push_server_demo_app_usb`, for serial and USB communication using a RFC split.

The demo applications are used like described above for Windows.

Terms and Definitions

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
CSR	Cambridge Silicon Radio
UniFi™	Group term for CSR's range of chips designed to meet IEEE 802.11 standards

Document History

Revision	Date	History
1	26 SEP 11	Ready for release 18.2.0

TradeMarks, Patents and Licences

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

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.