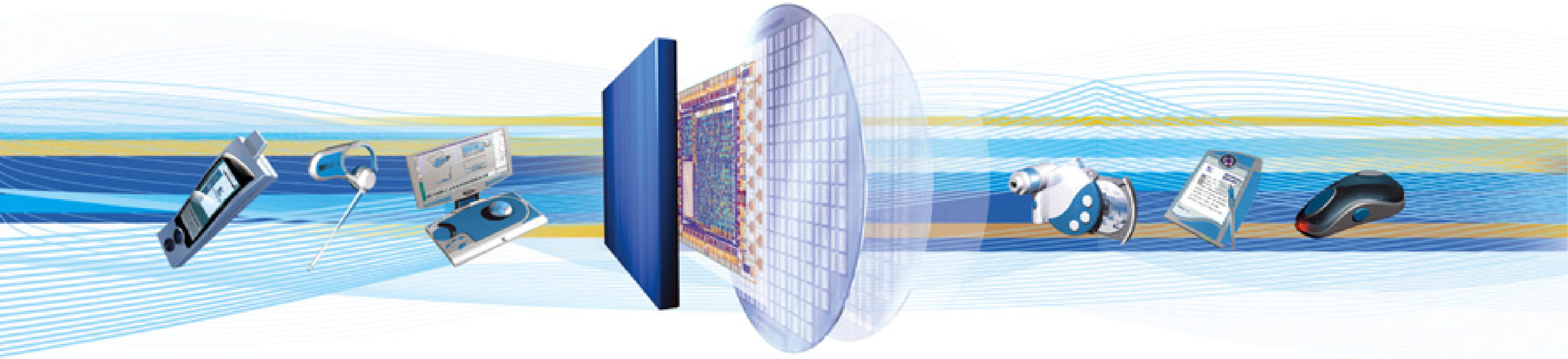
CSR Synergy Bluetooth 18.2.0

PAS – Obex Phone Book Access Server Profile

API 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 Introduction....4

1.1 Introduction and Scope 4

1.2 Assumptions....4

2 Description....5

2.1 Introduction....5

2.2 Reference Model 5

2.3 Sequence Overview....6

3 Interface Description....7

3.1 Activation....7

3.2 Connect....7

3.3 Pulling a Phone Book Object....8

3.4 Browsing Folders....9

3.5 Deactivation....10

3.6 Payload Encapsulated Data 10

3.6.1 Using Offsets....10

3.6.2 Payload Memory....11

4 OBEX Phone Book Access Server Primitives 12

4.1 List of All Primitives....12

4.2 CSR_BT_PAS_ACTIVATE....13

4.3 CSR_BT_PAS_DEACTIVATE....14

4.4 CSR_BT_PAS_CONNECT....15

4.5 CSR_BT_PAS_AUTHENTICATE 16

4.6 CSR_BT_PAS_PULL_PB / CSR_BT_PAS_NEXT 18

4.7 CSR_BT_PAS_PULL_PB_SIZE....20

4.8 CSR_BT_PAS_PULL_MCH 21

4.9 CSR_BT_PAS_PULL_MCH_SIZE....23

4.10 CSR_BT_PAS_SET_FOLDER....24

4.11 CSR_BT_PAS_SET_BACK_FOLDER....25

4.12 CSR_BT_PAS_SET_ROOT_FOLDER 26

4.13 CSR_BT_PAS_PULL_VCARD_LIST....27

4.14 CSR_BT_PAS_PULL_VCARD_LIST_SIZE 29

4.15 CSR_BT_PAS_PULL_VCARD_MCH_LIST 30

4.16 CSR_BT_PAS_PULL_VCARD_MCH_LIST_SIZE....32

4.17 CSR_BT_PAS_PULL_VCARD_ENTRY....33

4.18 CSR_BT_PAS_ABORT....35

4.19 CSR_BT_PAS_DISCONNECT....36

4.20 CSR_BT_PAS_SECURITY_IN....37

5 Document References....38

CSR Synergy Bluetooth 18.2.0 PAS API

List of Figures

Figure 1: Reference model 5
Figure 2: PAS state diagram 6
Figure 3: PAS activation 7
Figure 4: Connection handling 8
Figure 5: Outgoing message handling 8
Figure 6: Folder browsing handling 9
Figure 7: PAS deactivation 10

List of Tables

Table 1: List of all primitives 12
Table 2: CSR_BT_PAS_ACTIVATE Primitives 13
Table 3: CSR_BT_PAS_DEACTIVATE Primitives 14
Table 4: CSR_BT_PAS_CONNECT Primitives 15
Table 5: CSR_BT_PAS_AUTHENTICATE Primitives 16
Table 6: CSR_BT_PAS_PULL_PB, CSR_BT_PAS_NEXT Primitives 18
Table 7: CSR_BT_PAS_PULL_PB_SIZE Primitives 20
Table 8: CSR_BT_PAS_PULL_PB_MCH, CSR_BT_PAS_NEXT Primitives 21
Table 9: CSR_BT_PAS_PULL_MCH_SIZE Primitives 23
Table 10: CSR_BT_PAS_SET_FOLDER Primitives 24
Table 11: CSR_BT_PAS_SET_BACK_FOLDER Primitives 25
Table 12: CSR_BT_PAS_SET_ROOT_FOLDER Primitives 26
Table 13: CSR_BT_PAS_PULL_VCARD_LIST Primitives 27
Table 14: CSR_BT_PAS_PULL_VCARD_LIST_SIZE Primitives 29
Table 15: CSR_BT_PAS_PULL_VCARD_MCH_LIST Primitives 30
Table 16: CSR_BT_PAS_PULL_VCARD_MCH_LIST_SIZE Primitives 32
Table 17: CSR_BT_PAS_PULL_VCARD_ENTRY Primitives 33
Table 18: CSR_BT_PAS_ABORT Primitives 35
Table 19: CSR_BT_PAS_DISCONNECT Primitives 36
Table 20: CSR_BT_PAS_SECURITY_IN Primitives 37

1 Introduction

1.1 Introduction and Scope

This document describes the message interface provided by the OBEX Phone Book Access Profile Server (PAS). The PAS conforms to the server side of the Phone Book Access Profile, ref. [PBAP].

1.2 Assumptions

The following assumptions and preconditions are made in the following:

- There is a secure and reliable transport between the profile part, i.e. PAS and the application
- The PAS shall only handle one request at the time
- Bonding (pairing) is NOT handled by the PAS

2 Description

2.1 Introduction

The scenarios covered by this profile are the following:

- Usage of a Bluetooth® device e.g. a car-kit installed in a car retrieves phone book objects from a mobile phone
- A second usage is a car-kit browsing the phone book virtual folder structure and copying single entries from the phone book

The OBEX Phone Book Access Server (PAS) must be activated by the application. When it is activated it is able to provide the application with ability to pull and browse phone book entries and call lists.

The PAS provides Service Discovery handling.

The PAS is handling the interpretation of the OBEX packet.

The application is responsible for handling the indications from the PAS and sending the correct responses. The response codes used are described in the IrOBEX Specification [OBEX]. The PAS does not check and verify the data in the responses. Thus, it is the responsibility of the application to make sure that data follows the appropriate standards and formats. For further details on this subject please consult ref. [PBAP] and [OBEX].

2.2 Reference Model

The PAS interfaces to the Connection Manager (CM) and to the Service Discovery Server (SDS).

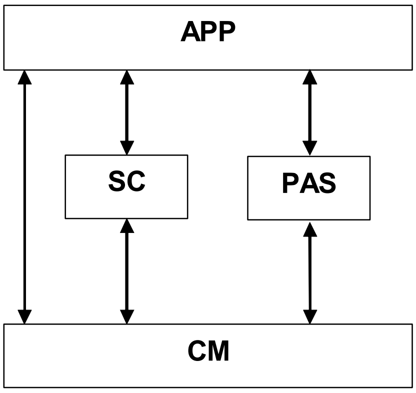


Figure 1: Reference model

2.3 Sequence Overview

The PAS starts up being in IDLE state. When the application activates PAS, the server enters ACTIVATE state and is ready to handle incoming requests. The server remains in this state until deactivated by the application. When deactivated it re-enters IDLE state.

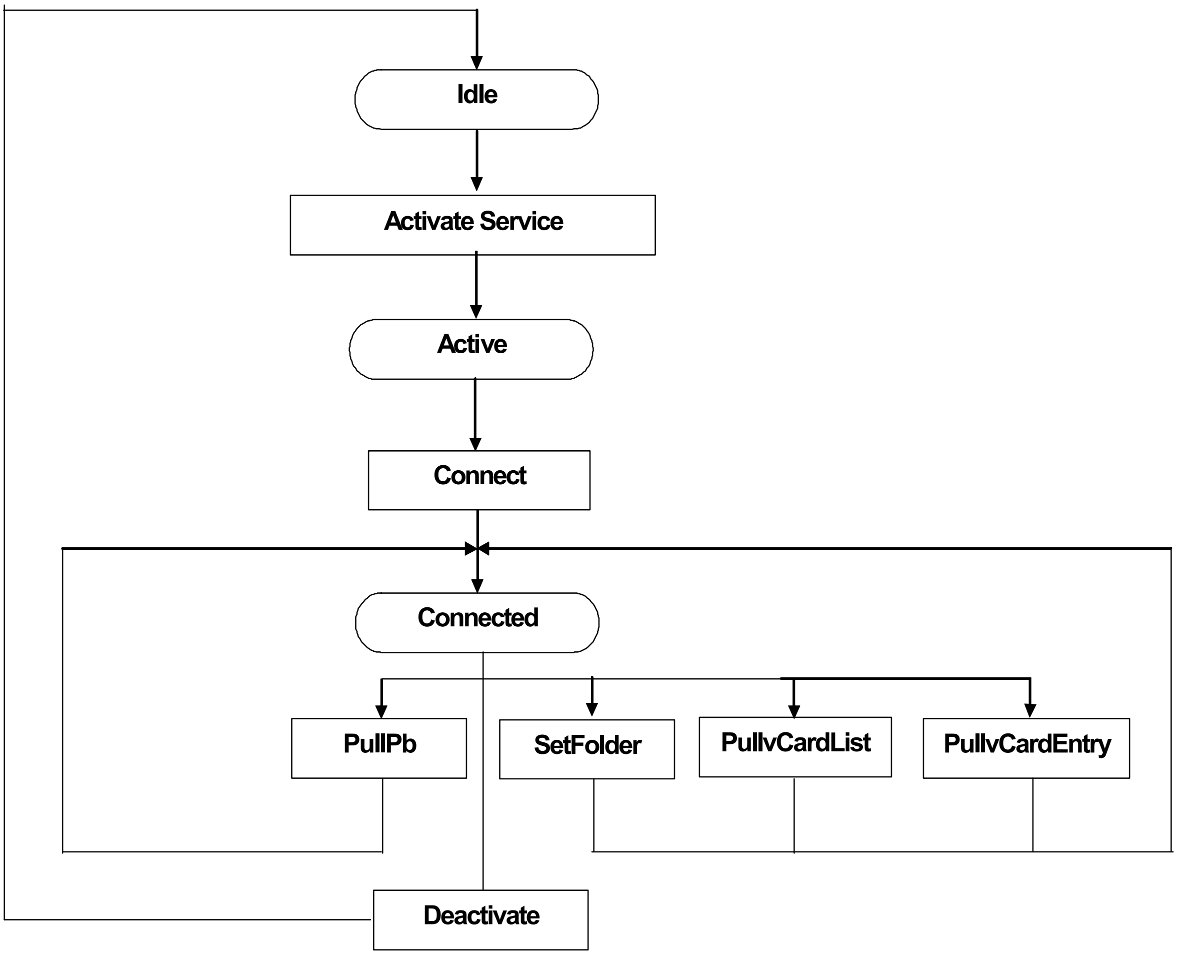


Figure 2: PAS state diagram

3 Interface Description

3.1 Activation

Sending a CSR_BT_PAS_ACTIVATE_REQ to the PAS activates the PAS. The PAS then registers a Service Record, in the Service Discovery Server, and makes it connectable. The PAS is now ready to handle incoming requests.

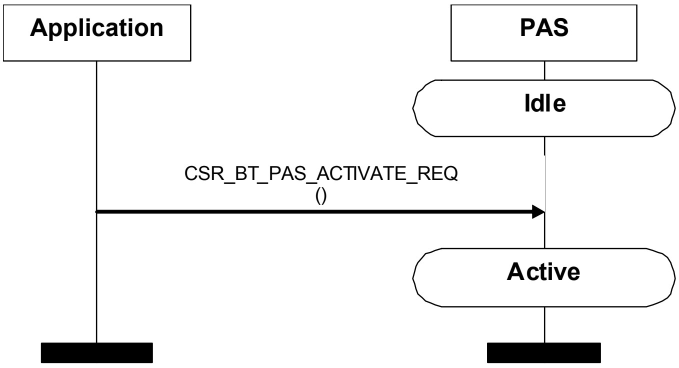


Figure 3: PAS activation

Please note that whether or not the Bluetooth device will be discoverable, i.e. can be found by other Bluetooth devices, it must be controlled by the application. For more information, please refer to [CM]. After initialization of CSR Synergy Bluetooth the Bluetooth® device is set up to be discoverable.

3.2 Connect

When the client is making a connect against the server the first message the application receives is CSR_BT_PAS_CONNECT_IND, this message has a parameter ObexPeerMaxPacketSize indicating the maximum OBEX packet size which the application can send down to the PAS in the body in one message response.

The application response is a CSR_BT_PAS_CONNECT_RES message with the appropriate result code. This message has the parameter ObexMaxPacketSize being the maximum packet (body) size that the application wants to receive from the client. There is a defined CSR_BT_MAX_OBEX_SIGNAL_LENGHT and the application must use this in the response. This value is calculated from the defined CSR_BT_MAX_OBEX_SIGNAL_LENGTH and both defines are placed in the file csr_bt_obex.h. The value can be between 255 bytes – 64K bytes – 1, see definition in ref. [OBEX]. If the packet size is large it is optimizing for quick file transfer, but the disadvantage will be use of big memory block. The memory use will increase with the packet size.

In case of a connection with the server with OBEX authentication the application must respond with a CSR_BT_PAS_AUTHENTICATE_REQ, it will then get a CSR_BT_PAS_AUTHENTICATE_CFM if client passes the authentication and a new CSR_BT_PAS_CONNECT_IND.

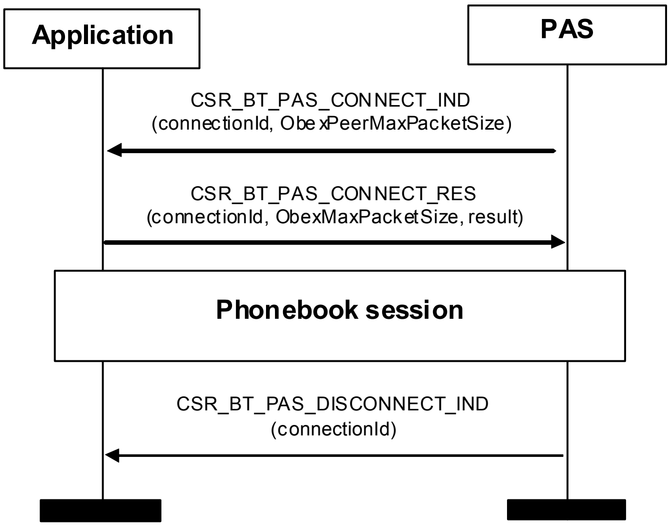

Application
PAS
CSR_BT_PAS_CONNECT_IND
(connectionId, ObexPeerMaxPacketSize)
CSR_BT_PAS_CONNECT_RES
(connectionId, ObexMaxPacketSize, result)
Phonebook session
CSR_BT_PAS_DISCONNECT_IND
(connectionId)

Figure 4: Connection handling

3.3 Pulling a Phone Book Object

When the PAS receives a request to send a phone book object to the client side, it sends a CSR_BT_PAS_PULL_PB_IND message to the application with the name parameter set to the requested name of the object. The application responds with a CSR_BT_PAS_PULL_PB_RES with the appropriate result code. CSR_BT_PAS_PULL_PB_RES only contains the data about the phone book object; the actual phone book object is transferred using CSR_BT_PAS_NEXT_IND/RES signals. In case the phone book object is too big to be contained in one CSR_BT_PAS_NEXT_RES signal it can be spilt and transferred using multiply CSR_BT_PAS_NEXT_IND/RES signals. The finalFlag is used for telling PAS that this is the last signal in the operation.

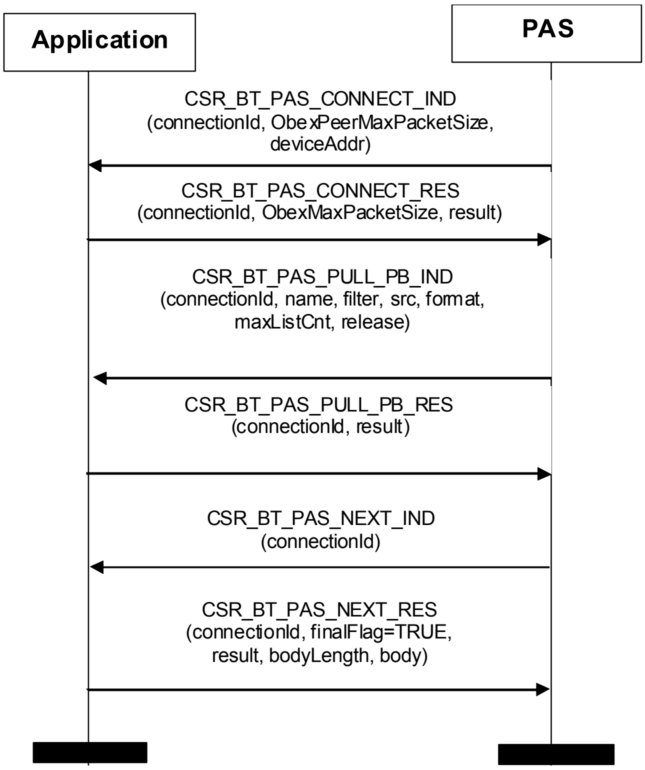


Figure 5: Outgoing message handling

3.4 Browsing Folders

Browsing the application's virtual folder structure (see, ref. [PBAP]) involves pulling the folder list and setting the current folder. The CSR_BT_PAS_SET_FOLDER_IND is used for changing the current folder. The CSR_BT_PAS_PULL_VCARD_LIST_IND is used for getting a vCard listing of the current folder structure. It must respond with a CSR_BT_PAS_PULL_VCARD_LIST_RES. The actual vCard listing object will be transferred using the CSR_BT_PAS_NEXT_IND/RES signals. The fragmentation and use of finalFlag is described under section 3.3.

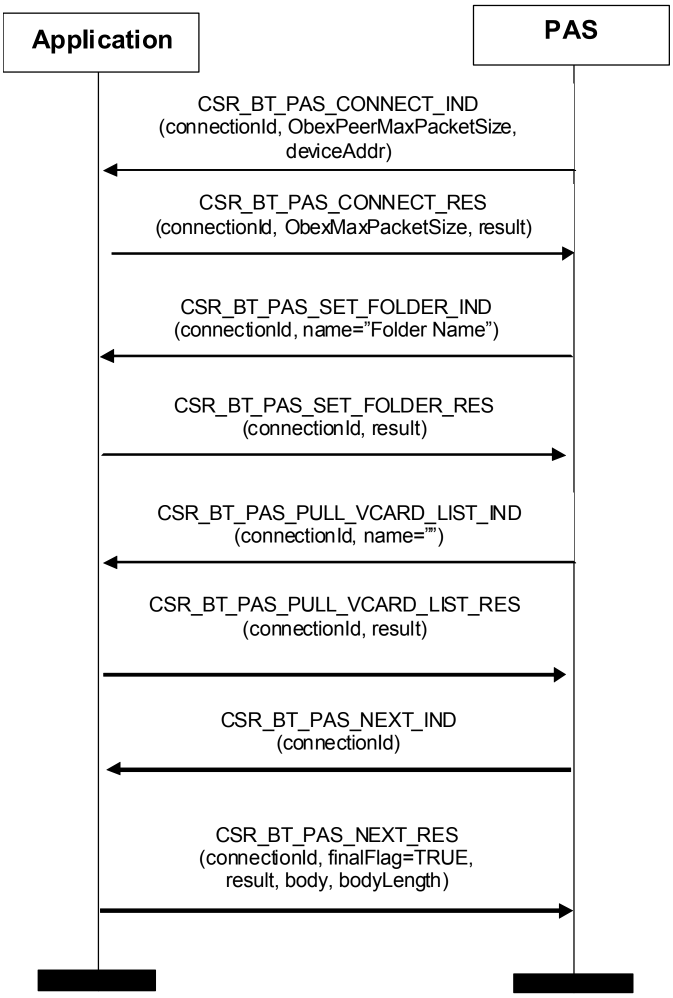

Application
PAS
CSR_BT_PAS_CONNECT_IND (connectionId, ObexPeerMaxPacketSize, deviceAddr)
CSR_BT_PAS_CONNECT_RES (connectionId, ObexMaxPacketSize, result)
CSR_BT_PAS_SET_FOLDER_IND (connectionId, name="Folder Name")
CSR_BT_PAS_SET_FOLDER_RES (connectionId, result)
CSR_BT_PAS_PULL_VCARD_LIST_IND (connectionId, name="")
CSR_BT_PAS_PULL_VCARD_LIST_RES (connectionId, result)
CSR_BT_PAS_NEXT_IND (connectionId)
CSR_BT_PAS_NEXT_RES (connectionId, finalFlag=TRUE, result, body, bodyLength)

Figure 6: Folder browsing handling

CSR Synergy Bluetooth 18.2.0 PAS API

3.5 Deactivation

Sending a CSR_BT_PAS_DEACTIVATION_REQ to the PAS can deactivate the PAS. This procedure can take some time depending on the current PAS activity. When deactivated, the PAS confirms the deactivation with a CSR_BT_PAS_DEACTIVATE_CFM message.

Any transaction in progress will be terminated immediately when this message is received by the PAS.

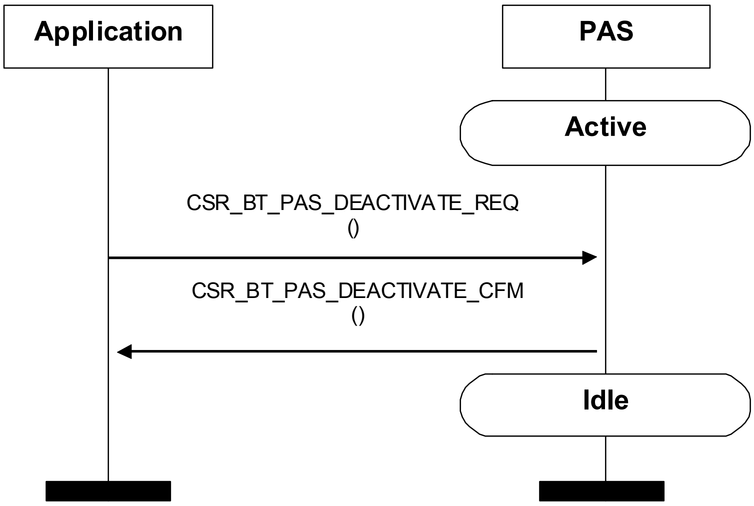


Figure 7: PAS deactivation

3.6 Payload Encapsulated Data

3.6.1 Using Offsets

As many OBEX messages contain multiple parameters with variable length, some of the parameters are based on *offsets* instead of standard pointers to the data. Signals with offset-based data can easily be recognized as they have both a *payload* and a *payloadLength* parameter. The *payload* contains the actual data, on which the offset is based. For example, a typical signal may contain the following:

```
CsrBtCommonPrim    type;
CsrUint8               result;
CsrUint16          ucs2nameOffset;
CsrUint16          bodyOffset;
CsrUint16          bodyLength;
CsrUint16          payloadLength;
CsrUint8               *payload;
```

In this example, two offset parameters can be found, namely *ucs2nameOffset* and *bodyOffset*. To obtain the actual data, the offset value is added to the *payload* pointer, which yields a pointer to the data, i.e.:

```
  CsrUint8  *ucs2name;
  ucs2name = (CsrUint8*)(primitive->payload + primitive->ucs2nameOffset);
```

As can be seen, the offset contains the number of bytes within the *payload* where the information begins. Similarly, the body data can be retrieved using the following:

```
CsrUint8 *body;
body = (CsrUint8*)(primitive->payload + primitive->bodyOffset);
```

And to illustrate the usage of the *length* parameter, which is also a common parameter, to copy the body one would typically use:

```
CsrMemCpy( copyOfBody, body, primitive->bodyLength );
```

Offset parameters will always have an “Offset” suffix on the name, and offsets are *always* relative to the “payload” parameter. Thus, an offset field containing the value 0 (zero) means that the information it relates to does not actually exist in the “payload”. For example, if the “ucs2nameOffset” field in the example above is 0 (zero) then it means that the parameter “name” does not exist in the “payload”.

If the `bodyOffset` or the `bodyLength` is 0 (zero) this means that the signal does not contain any body. The same holds when the `payloadLength` is 0 (zero), which means that there is not payload.

3.6.2 Payload Memory

When the application receives a signal which has a *payload* parameter, the application must always free the payload pointer to avoid memory leaks, for example

```
    CsrPfree(primitive->payload);
    CsrPfree(primitive);
```

will free both the payload data and the message itself. Note that when the payload has been freed, offsets can not be used anymore, as the actual data is contained within the payload.

Signals that do not use the *payload* parameter must still have each of their pointer-based parameters freed.

4 OBEX Phone Book Access Server Primitives

This section gives an overview of the primitives and parameters in the interface. Detailed information can be found in the corresponding csr_bt_pas_prim.h file.

4.1 List of All Primitives

Primitives:	Reference:
CSR_BT_PAS_ACTIVATE_REQ	See section 4.2
CSR_BT_PAS_DEACTIVATE_REQ	See section 4.3
CSR_BT_PAS_DEACTIVATE_CFM	See section 4.3
CSR_BT_PAS_CONNECT_IND	See section 4.4
CSR_BT_PAS_CONNECT_RES	See section 4.4
CSR_BT_PAS_AUTHENTICATE_REQ	See section 4.5
CSR_BT_PAS_AUTHENTICATE_CFM	See section 4.5
CSR_BT_PAS_AUTHENTICATE_IND	See section 4.5
CSR_BT_PAS_AUTHENTICATE_RES	See section 4.5
CSR_BT_PAS_PULL_PB_IND	See section 4.6
CSR_BT_PAS_PULL_PB_RES	See section 4.6
CSR_BT_PAS_PULL_PB_SIZE_IND	See section 4.7
CSR_BT_PAS_PULL_PB_SIZE_RES	See section 4.7
CSR_BT_PAS_PULL_MCH_IND	See section 4.8
CSR_BT_PAS_PULL_MCH_RES	See section 4.8
CSR_BT_PAS_PULL_MCH_SIZE_IND	See section 4.9
CSR_BT_PAS_PULL_MCH_SIZE_RES	See section 4.9
CSR_BT_PAS_SET_FOLDER_IND	See section 4.10
CSR_BT_PAS_SET_FOLDER_RES	See section 4.10, 4.11 and 4.12
CSR_BT_PAS_SET_BACK_FOLDER_IND	See section 4.13
CSR_BT_PAS_SET_ROOT_FOLDER_IND	See section 4.12
CSR_BT_PAS_PULL_VCARD_LIST_IND	See section 4.13
CSR_BT_PAS_PULL_VCARD_LIST _RES	See section 4.13
CSR_BT_PAS_PULL_VCARD_LIST_SIZE_IND	See section 4.14
CSR_BT_PAS_PULL_VCARD_LIST_SIZE_RES	See section 4.14
CSR_BT_PAS_PULL_VCARD_MCH_LIST_IND	See section 4.15
CSR_BT_PAS_PULL_VCARD_MCH_LIST_RES	See section 4.15
CSR_BT_PAS_PULL_VCARD_MCH_LIST_SIZE_IND	See section 4.16
CSR_BT_PAS_PULL_VCARD_MCH_LIST_SIZE_RES	See section 4.16
CSR_BT_PAS_PULL_VCARD_ENTRY_IND	See section 4.17
CSR_BT_PAS_PULL_VCARD_ENTRY_RES	See section 4.17
CSR_BT_PAS_NEXT_IND	See section 4.6, 4.13 and 4.17
CSR_BT_PAS_NEXT_RES	See section 4.6, 4.13 and 4.17
CSR_BT_PAS_ABORT_IND	See section 4.18
CSR_BT_PAS_DISCONNECT_IND	See section 4.19
CSR_BT_PAS_SECURITY_IN_REQ	See section 4.20
CSR_BT_PAS_SECURITY_IN_CFM	See section 4.20

Table 1: List of all primitives

4.2 CSR_BT_PAS_ACTIVATE

Primitives	type	appHandle	feature	repository	obexMaxPacketSize	windowSize	srmEnable
CSR_BT_PAS_ACTIVATE_REQ	✓	✓	✓	✓	✓	✓	✓

Table 2: CSR_BT_PAS_ACTIVATE Primitives

Description

This signal is used for activating the PAS and make it accessible from a remote device. The process includes:

1. Register the OBEX PAS Server service in the service discovery database.
2. Enabling page scan.

The PAS will remain activated until a CSR_BT_PAS_DEACTIVATE_REQ is received.

Parameters

type — Signal identity, CSR_BT_PAS_ACTIVATE_REQ.

appHandle — The identity of the calling process. It is possible to initiate the procedure by any higher layer process as the response is returned to appHandle.

feature — 8 bit flag indicating the feature the PBAP server will support. Bit 0 = downloaded supported, bit 1 = browsing supported, bit 2-7 reserved for feature use

repository — 8 bit flag indicating the repositories the PBAP server supports. Bit 0 = internal phone book supported, bit 1 SIM card phone book supported, bit 2- 7 reserved for future use

obexMaxPacketSize — To control the maximum allowed obex packet size the application can receive. There is a define CSR_BT_MAX_OBEX_SIGNAL_LENGHT (in csr_bt_obex.h) to be used for this value, the max allowed value is 64K bytes – 1.

windowSize — Controls how many packets the OBEX profile (and lower protocol layers) are allowed to cache on the data receive side. A value of zero (0) will cause the system to auto-detect this value.

srmEnable — Enable local support for Single Response Mode.

4.3 CSR_BT_PAS_DEACTIVATE

Parameters / Primitives	type
CSR_BT_PAS_DEACTIVATE_REQ	✓
CSR_BT_PAS_DEACTIVATE_CFM	✓

Table 3: CSR_BT_PAS_DEACTIVATE Primitives

Description

This signal deactivates the PAS. The service cannot be re-activated until after the application has received a CSR_BT_PAS_DEACTIVATE_CFM.

The service will no longer be visible to inquire devices and the inquiry and page scan may be stopped (depending on the fact if other services are available or not). The OBEX PAS Server service is removed from the service discovery database.

The signal will stop any ongoing transaction.

Parameters

type	Signal identity, CSR_BT_PAS_DEACTIVATE_REQ/CFM.

4.4 CSR_BT_PAS_CONNECT

Primitives \ Parameters	type	connectionId	obexPeerMaxPacketSize	deviceAddr	responseCode	length	count	btConnId
CSR_BT_PAS_CONNECT_IND	✓	✓	✓	✓		✓	✓	✓
CSR_BT_PAS_CONNECT_RES	✓	✓			✓			

Table 4: CSR_BT_PAS_CONNECT Primitives

Description

This signal is indicating that a PAS client is starting a phone book session. The application can then accept or deny the result and has to return the connectionId received in the indication.

Parameters

type Signal identity, CSR_BT_PAS_CONNECT_IND/RES.

connectionId Is the connection Id for this session, the PAS client must use this Id in the request.

obexPeerMaxPacketSize The maximum OBEX packet size being allowed to send to the client application.

deviceAddr The Bluetooth address which is connected to the device

responseCode The valid response codes are defined (in csr_bt_obex.h). For accepting a connection the code is CSR_BT_OBEX_SUCCESS_RESPONSE_CODE, any other response code indicates a failure to make a connection.

length The length parameter contains the length in bytes of the bodies of all the objects that the sender plans to send. Note this length cannot be guarantee correct, so while the value may be useful for status indicators and resource reservations, the application should not die if the length is not correct.

If 0 this parameter were not included in the received OBEX Connect Request packet.

count Count is use to indicate the number of objects that will be sent by the sender during this connection.

If 0 this parameter were not included in the received OBEX Connect Request packet.

btConnId Identifier used when moving the connection to another AMP controller, i.e. when calling the `CsrBtAmpmMoveReqSend`-function.

4.5 CSR_BT_PAS_AUTHENTICATE

Primitives \ Parameters	type	options	realmLength	* realm	deviceAddr	*password	passwordLength	*userId
CSR_BT_PAS_AUTHENTICATE_REQ	✓		✓	✓		✓	✓	✓
CSR_BT_PAS_AUTHENTICATE_CFM	✓							
CSR_BT_PAS_AUTHENTICATE_IND	✓	✓	✓	✓	✓			
CSR_BT_PAS_AUTHENTICATE_RES	✓					✓	✓	✓

Table 5: CSR_BT_PAS_AUTHENTICATE Primitives

Description

The request signal is used when the PAS wants to OBEX authenticate the client. The application has to send a password or pin number in the password to authenticate the client with. The authentication of the client is only a success if the application receives a CSR_BT_PAS_AUTHENTICATE_CFM.

The indication and response signal is used when the Phone Book Access Profile client wants to OBEX authenticate the PAS. The application has to response with a password or pin number in the password and userId for client to identify the proper password.

Parameters

type — Signal identity, CSR_BT_PAS_AUTHENTICATE_REQ/CFM/IND/RES.

options — Challenge information of type CsrUint8.

Bit 0 controls the responding of a valid user Id.
If bit 0 is set it means that the application must response with a user Id in a CSR_BT_PAS_AUTHENTICATE_RES message. If bit 0 is not set the application can just set the userId to NULL.

Bit 1 indicates the access mode being offered by the sender
If bit 1 is set the access mode is read only. If bit 1 is not set the sender gives full access, e.g. both read and write.

Bit 2 - 7 is reserved.

realmLength — Number of bytes in realm of type CsrUint16

Note in this release version the ‘realmLength’ parameter is always set to 0x0000 in the CSR_BT_PAS_AUTHENTICATE_IND and in the CSR_BT_PAS_AUTHENTICATE_REQ is ignored right now.

* realm — A displayable string indicating for the user which userid and/or password to use. The first byte of the string is the character set of the string. The table below shows

the different values for character set.

Note that this pointer must be CsrPfree by the application, and that this pointer can be NULL because the realm field is optional to set by the peer device.

Note in this release version the ‘realm’ pointer is always set to NULL in the CSR_BT_PAS_AUTHENTICATE_IND and in the CSR_BT_PAS_AUTHENTICATE_REQ the ‘realm’ is ignored right now.

Char set Code	Meaning
0	ASCII
1	ISO-8859-1
2	ISO-8859-2
3	ISO-8859-3
4	ISO-8859-4
5	ISO-8859-5
6	ISO-8859-6
7	ISO-8859-7
8	ISO-8859-8
9	ISO-8859-9
0xFF = 255	UNICODE

deviceAddr	The Bluetooth address of the device that has initiated the OBEX authentication procedure
*password	Containing the challenge password of the OBEX authentication. This is a pointer which shall be allocated by the application.
passwordLength	The length of the challenge password.
*userId	Zero terminated string (ASCII) containing the userId for the authentication. This is a pointer which shall be allocated by the application. Note in CSR_BT_PAS_AUTHENTICATE_REQ the userId is ignored right now.

4.6 CSR_BT_PAS_PULL_PB / CSR_BT_PAS_NEXT

Parameters / Primitives	type	connectionId	ucs2nameOffset	filter[8]	src	format	maxListCnt	listStartOffset	responseCode	bodyLength	*body	payloadLength	*payload	finalFlag
CSR_BT_PAS_PULL_PB_IND	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓	
CSR_BT_PAS_PULL_PB_RES	✓	✓							✓					
CSR_BT_PAS_NEXT_IND	✓	✓												
CSR_BT_PAS_NEXT_RES	✓	✓							✓	✓	✓			✓

Table 6: CSR_BT_PAS_PULL_PB, CSR_BT_PAS_NEXT Primitives

Description

Used for retrieving one or more phone book objects from the PAS. The parameter name, in the CSR_BT_PAS_PULL_PB_IND signal specifies the phone book of interest, the server responses with a CSR_BT_PAS_PULL_PB_RES. If the response has a body (which is true in most cases) then multiple requests (CSR_BT_PAS_NEXT_IND) are used for retrieving the object. The last response (CSR_BT_PAS_NEXT_RES) must set the finalFlag to true to indicate it's the last packet in the session. In case the result is different from success, the other parameters are invalid and not used.

Parameters

type — Signal identity, CSR_BT_PAS_PULL_PB _IND/RES, CSR_BT_PAS_NEXT_IND/RES.

connectionId — The connection Id for this session, the PAS client will use this Id in the request.

ucs2nameOffset — Payload relative offset of a null terminated 16 bit Unicode text string (UCS2) containing the (file) name of the object.

The function "CsrUcs2ByteString2Utf8" can be used for converting a null terminated UCS2 text string into a null terminated UTF8 text string

filter[8] — Indicate the attributes requested by the client; see the detailed description in [PBAP].

The filter is a 64 bit long flag store in an array of 8 CsrUint8. The lower part of the flag; filter[0] contains bit 0 – 7, filter[1] contains bit 8 – 15, filter[2] contains bit 16 – 23, filter[3] contains bit 24 – 31, filter[4] contains bit 32 – 39, filter[5] contains bit 40 – 47, filter[6] contains bit 48 – 55 and filter[7] contains bit 56 – 63.

src — Indicate whether phone book source should be the internal memory (CSR_BT_PAS_SRC_PHONE) or the SIM (CSR_BT_PAS_SRC_SIM).

format — Used for indicating the requested format (vCard 2.1 or 3.0) to be returned in the operation.

maxListCnt — The maximum number of entries allowed in returned phone book object.

listStartOffset — Used for indicating the offset of the first entry in phone book object

responseCode — The valid response codes are defined (in csr_bt_obex.h). For success in the request the code is CSR_BT_OBEX_SUCCESS_RESPONSE_CODE, any other response

	code indicates a failure.
bodyLength	The length of the body (object).
*body	The object itself. "body" is a CsrUint8 pointer to the object.
payloadLength	Number of bytes in the payload structure.
*payload	OBEX payload data. Offsets are relative to this pointer.
finalFlag	Indicates that the body (Object) fits the whole objects or that it is the last part

4.7 CSR_BT_PAS_PULL_PB_SIZE

Parameters / Primitives	type	connectionId	ucs2nameOffset	src	pbSize	responseCode	payloadLength	*payload	smpOn
CSR_BT_PAS_PULL_PB_SIZE_IND	✓	✓	✓	✓			✓	✓	
CSR_BT_PAS_PULL_PB_SIZE_RES	✓	✓			✓	✓			✓

Table 7: CSR_BT_PAS_PULL_PB_SIZE Primitives

Description

Retrieve the size of a phone book. The parameter name specifies the phone book of interest, the server responses with CSR_BT_PAS_PULL_PB_SIZE_RES. In case the result is different from success, the other parameters are invalid and not used.

Parameters

type
Signal identity, CSR_BT_PAS_PULL_PB _SIZE_IND/RES.

connectionId
The connection Id for this session, the PAS client will use this Id in the request.

ucs2nameOffset
Payload relative offset of a null terminated 16 bit Unicode text string (UCS2) containing the (file) name of the object.

The function “CsrUcs2ByteString2Utf8” can be used for converting a null terminated UCS2 text string into a null terminated UTF8 text string

src
Indicate whether phone book source should be the internal memory (CSR_BT_PAS_SRC_PHONE) or the SIM (CSR_BT_PAS_SRC_SIM)

responseCode
The valid response codes are defined (in csr_bt_obex.h). For success in the request the code is CSR_BT_OBEX_SUCCESS_RESPONSE_CODE, any other response code indicates a failure.

pbSize
The number of entries in the phone book.

payloadLength
Number of bytes in the payload structure.

*payload
OBEX payload data. Offsets are relative to this pointer.

smpOn
Reserved for future use. Set to FALSE.

4.8 CSR_BT_PAS_PULL_MCH

Parameters / Primitives	type	connectionId	filter[8]	src	format	maxListCnt	listStartOffset	newMissedCall	responseCode	*body	bodyLength	finalFlag	smpOn
CSR_BT_PAS_PULL_MCH_IND	✓	✓	✓	✓	✓	✓	✓						
CSR_BT_PAS_PULL_MCH_RES	✓	✓						✓	✓				✓
CSR_BT_PAS_NEXT_IND	✓	✓											
CSR_BT_PAS_NEXT_RES	✓	✓							✓	✓	✓	✓	✓

Table 8: CSR_BT_PAS_PULL_PB_MCH, CSR_BT_PAS_NEXT Primitives

Description

To retrieve a Missed call history object from the PAS. The server responses the CSR_BT_PAS_PULL_MCH_IND signal with a CSR_BT_PAS_PULL_PB_RES. If the response has a body (which is true in most cases) then multiple requests (CSR_BT_PAS_NEXT_IND) are used for retrieving the object. The last response (CSR_BT_PAS_NEXT_RES) must set the finalFlag to true to indicate it is the last packet in the session. In case the result is different from success, the other parameters are invalid and not used.

Parameters

type — Signal identity, CSR_BT_PAS_PULL_MCH _IND/RES, CSR_BT_PAS_NEXT_IND/RES.

connectionId — The connection Id for this session, the PAS client will use this Id in the request.

filter[8] — Indicate the attributes requested by the client; see the detailed description in [PBAP].

The filter is a 64 bit long flag store in an array of 8 CsrUint8. The lower part of the flag; filter[0] contains bit 0 – 7, filter[1] contains bit 8 – 15, filter[2] contains bit 16 – 23, filter[3] contains bit 24 – 31, filter[4] contains bit 32 – 39, filter[5] contains bit 40 – 47, filter[6] contains bit 48 – 55 and filter[7] contains bit 56 – 63.

src — Indicate whether phone book source should be the internal memory (CSR_BT_PAS_SRC_PHONE) or the SIM (CSR_BT_PAS_SRC_SIM)

format — Used for indicating the requested format (vCard 2.1 or 3.0) to be returned in the operation.

maxListCnt — The maximum number of entries allowed in returned phone book object.

listStartOffset — Used for indicating the offset of the first entry in phone book object

responseCode — The valid response codes are defined (in csr_bt_obex.h). For success in the request the code is CSR_BT_OBEX_SUCCESS_RESPONSE_CODE, any other response code indicates a failure.

newMissedCall — The number of missed calls that has not been checked yet since the PBAP session started.

bodyLength — The length of the body (object).

*body The object itself. "body" is a CsrUint8 pointer to the object.

finalFlag Indicates that the body (Object) fits the whole objects or that it is the last part.

smpOn Reserved for future use. Set to FALSE.

4.9 CSR_BT_PAS_PULL_MCH_SIZE

Parameters / Primitives	type	connectionId	src	pbSize	newMissedCall	responseCode	smpOn
CSR_BT_PAS_PULL_MCH_SIZE_IND	✓	✓	✓				
CSR_BT_PAS_PULL_MCH_SIZE_RES	✓	✓		✓	✓	✓	✓

Table 9: CSR_BT_PAS_PULL_MCH_SIZE Primitives

Description

To retrieve the size of the Missed Call history object from the PAS, the server responses with CSR_BT_PAS_PULL_MCH_SIZE_RES. In case the result is different from success, the other parameters are invalid and not used.

Parameters

type	Signal identity, CSR_BT_PAS_PULL_MCH_SIZE_IND/RES.
connectionId	The connection Id for this session, the PAS client will use this Id in the request.
src	Indicate whether phone book source should be the internal memory (CSR_BT_PAS_SRC_PHONE) or the SIM (CSR_BT_PAS_SRC_SIM)
responseCode	The valid response codes are defined (in csr_bt_obex.h). For success in the request the code is CSR_BT_OBEX_SUCCESS_RESPONSE_CODE, any other response code indicates a failure.
pbSize	The number of entries in the phone book.
newMissedCall	The number of missed calls that has not been checked yet since the PBAP session started.
smpOn	Reserved for future use. Set to FALSE.

4.10 CSR_BT_PAS_SET_FOLDER

Parameters / Primitives	type	connectionId	ucs2nameOffset	responseCode	payloadLength	*payload
CSR_BT_PAS_SET_FOLDER_IND	✓	✓	✓		✓	✓
CSR_BT_PAS_SET_FOLDER_RES	✓	✓		✓		

Table 10: CSR_BT_PAS_SET_FOLDER Primitives

Description

This signal is used for changing the current folder on the server, to a folder specified with the name parameter. This is used for navigating down in the directory hierarchy. The result of the change folder operation is given in the response signal. The result contains an error code corresponding to the reason for failure, the folder does not exist or if the application does not permit this operation from the client.

Parameters

type Signal identity, CSR_BT_PAS_SET_FOLDER_IND/RES.

connectionId The connection Id for this session, the PBAP client will use this Id in the request.

ucs2nameOffset Payload relative offset of a null terminated 16 bit Unicode text string (UCS2) containing the (file) name of the object.

The function "CsrUcs2ByteString2Utf8" can be used for converting a null terminated UCS2 text string into a null terminated UTF8 text string

responseCode The valid response codes are defined (in csr_bt_obex.h). For success in the request the code is CSR_BT_OBEX_SUCCESS_RESPONSE_CODE. Any other response code indicates a failure.

payloadLength Number of bytes in the payload structure.

*payload OBEX payload data. Offsets are relative to this pointer.

4.11 CSR_BT_PAS_SET_BACK_FOLDER

Parameters Primitives	type	connectionId	responseCode
CSR_BT_PAS_SET_BACK_FOLDER_IND	✓	✓	
CSR_BT_PAS_SET_FOLDER_RES	✓	✓	✓

Table 11: CSR_BT_PAS_SET_BACK_FOLDER Primitives

Description

This signal is used for setting the current folder back to the parent folder. The result of the operation is given in the response signal. If the current folder is the root folder the response is the CSR_BT_OBEX_NOT_FOUND_RESPONSE_CODE result code.

Parameters

type	Signal identity, CSR_BT_PAS_SET_BACK_FOLDER_IND.
connectionId	The connection Id for this session. The PBAP client will use this Id in the request.
responseCode	The valid response codes are defined (in csr_bt_obex.h). For success in the request the code is CSR_BT_OBEX_SUCCESS_RESPONSE_CODE. Any other response code indicates a failure.

4.12 CSR_BT_PAS_SET_ROOT_FOLDER

Parameters / Primitives	type	connectionId	responseCode
CSR_BT_PAS_SET_ROOT_FOLDER_IND	✓	✓	
CSR_BT_PAS_SET_FOLDER_RES	✓	✓	✓

Table 12: CSR_BT_PAS_SET_ROOT_FOLDER Primitives

Description

This signal is used for setting the current folder back to the root folder. The result will always be CSR_BT_OBEX_SUCCESS_RESPONSE_CODE for this operation.

Parameters

type — Signal identity, CSR_BT_PAS_SET_ROOT_FOLDER_IND, CSR_BT_PAS_SET_FOLDER_RES.

connectionId — The connection Id for this session, the PBAP client will use this Id in the request.

responseCode — The valid response codes are defined (in csr_bt_obex.h). For success in the request the code is CSR_BT_OBEX_SUCCESS_RESPONSE_CODE. Any other response code indicates a failure.

4.13 CSR_BT_PAS_PULL_VCARD_LIST

Primitives \ Parameters	type	connectionId	ucs2nameOffset	order	uft8SearchValOffset	searchValLength	searchAtt	maxListCnt	listStartOffset	responseCode	*body	bodyLength	payloadLength	*payload	finalFlag	smpOn
CSR_BT_PAS_PULL_VCARD _LIST_IND	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓		
CSR_BT_PAS_PULL_VCARD _LIST_RES	✓	✓								✓						✓
CSR_BT_PAS_NEXT_IND	✓	✓														
CSR_BT_PAS_NEXT_RES	✓	✓								✓	✓	✓			✓	✓

Table 13: CSR_BT_PAS_PULL_VCARD_LIST Primitives

Description

This signal is used for pulling the content of the current folder or a folder specified by the name parameter. The application must support the PBAP virtual folders structure as specified in [PBAP]. The content of the folder must be sent in the vCard-Listing format specified in, ref. [PBAP].

The signals CSR_BT_PAS_NEXT_IND/CSR_BT_PAS_NEST_RES is used for transferring the folder object (vCard-listing). The last CSR_BT_PAS_NEXT_RES signal must set finalFlag to TRUE, indicate that the operation have finish.

Parameters

type — Signal identity, CSR_BT_PAS_PULL_VCARD_LIST_IND /RES.

connectionId — The connection Id for this session, the PAS client will use this Id in the request.

ucs2nameOffset — Payload relative offset of a null terminated 16 bit Unicode text string (UCS2) containing the (file) name of the object.

The function “CsrUcs2ByteString2Utf8” can be used for converting a null terminated UCS2 text string into a null terminated UTF8 text string

order — This parameter is used for indicating to the PAS, which sorting order shall be used for the folder listing. The ordering must always be ascendant. The order can be Alphabetical (CSR_BT_PAS_ORDER_ALPHABETICAL), Indexed (CSR_BT_PAS_ORDER_INDEXED) or Phonetic (CSR_BT_PAS_ORDER_PHONETICAL), see detail in ref. [PBAP].

uft8SearchValOffset — Payload relative offset to a UTF8 string indicating to the server, which vCards shall be contained in the vCard-listing, see the description ref. [PBAP].

searchValLength — The length of the search value (utf8SearchValOffset)

searchAtt — Search attribute is used for indicating to the server, which attribute the search shall be performed.

maxListCnt — The maximum number of entries allowed in returned phone book object.

listStartOffset — Used for indicating the offset of the first entry in phone book object responseCode The valid response codes are defined (in csr_bt_obex.h). For success in the request the code is CSR_BT_OBEX_SUCCESS_RESPONSE_CODE, any other response code indicates a failure in the request.

bodyLength The length of the body (object).

*body The object itself. "body" is a CsrUint8 pointer to the object.

payloadLength Number of bytes in the payload structure.

*payload OBEX payload data. Offsets are relative to this pointer.

finalFlag Indicates that the body (Object) fits the whole objects or that it is the last part.

smpOn Reserved for future use. Set to FALSE.

4.14 CSR_BT_PAS_PULL_VCARD_LIST_SIZE

Parameters Primitives	type	connectionId	ucs2nameOffset	pbSize	responseCode	*payload	payloadLength	smpOn
CSR_BT_PAS_PULL_VCARD_LIST_SIZE_IND	✓	✓	✓			✓	✓	
CSR_BT_PAS_PULL_VCARD_LIST_SIZE_RES	✓	✓		✓	✓			✓

Table 14: CSR_BT_PAS_PULL_VCARD_LIST_SIZE Primitives

Description

This signal is used for getting the size of the current folder or a folder specified by the name parameter. The application must support the PBAP virtual folders structure as specified in [PBAP]. The content of the folder must be sent in the vCard-Listing format specified in, ref. [PBAP].

Parameters

type Signal identity, CSR_BT_PAS_PULL_VCARD_LIST_SIZE_IND /RES.

connectionId The connection Id for this session, the PAS client will use this Id in the request.

ucs2nameOffset Payload relative offset of a null terminated 16 bit Unicode text string (UCS2) containing the (file) name of the object.

The function “CsrUcs2ByteString2Utf8” can be used for converting a null terminated UCS2 text string into a null terminated UTF8 text string

responseCode The valid response codes are defined (in csr_bt _obex.h). For success in the request the code is CSR_BT_OBEX_SUCCESS_RESPONSE_CODE, any other response code indicates a failure.

pbSize The number of entries in the phone book.

payloadLength Number of bytes in the payload structure.

*payload OBEX payload data. Offsets are relative to this pointer.

smpOn Reserved for future use. Set to FALSE.

4.15 CSR_BT_PAS_PULL_VCARD_MCH_LIST

Primitives \ Parameters	type	connectionId	order	uft8SearchValOffset	searchValLength	searchAtt	maxListCnt	listStartOffset	newMissedCall	responseCode	*body	bodyLength	payloadLength	*payload	finalFlag	smpOn
CSR_BT_PAS_PULL_VCARD_MCH_LIST_IND	✓	✓	✓	✓	✓	✓	✓	✓					✓	✓		
CSR_BT_PAS_PULL_VCARD_MCH_LIST_RES	✓	✓							✓	✓						✓
CSR_BT_PAS_NEXT_IND	✓	✓														
CSR_BT_PAS_NEXT_RES	✓	✓								✓	✓	✓			✓	✓

Table 15: CSR_BT_PAS_PULL_VCARD_MCH_LIST Primitives

Description

This signal is used for pulling the content of the MCH folder. The application must support the PBAP virtual folders structure as specified in [PBAP]. The content of the folder must be sent in the vCard-Listing format specified in, ref. [PBAP].

The signals CSR_BT_PAS_NEXT_IND/CSR_BT_PAS_NEST_RES is used for transferring the folder object (vCard-listing). The last CSR_BT_PAS_NEXT_RES signal must set finalFlag to TRUE, indicate that the operation have finish.

Parameters

type	Signal identity, CSR_BT_PAS_PULL_VCARD_MCH_LIST_IND /RES.
connectionId	The connection Id for this session, the PAS client will use this Id in the request.
order	This parameter is used for indicating to the PAS, which sorting order shall be used for the folder listing. The ordering must always be ascendant. The order can be Alphabetical (CSR_BT_PAS_ORDER_ALPHABETICAL), Indexed (CSR_BT_PAS_ORDER_INDEXED) or Phonetic (CSR_BT_PAS_ORDER_PHONETICAL), see detail in ref. [PBAP].
uft8SearchValOffset	Payload relative offset to a UTF8 string indicating to the server, which vCards shall be contained in the vCard-listing, see the description ref. [PBAP].
searchValLength	The length of the search value (utf8SearchValOffset)
searchAtt	Search attribute is used for indicating to the server, which attribute the search shall be performed.
maxListCnt	The maximum number of entries allowed in returned phone book object.
listStartOffset	Used for indicating the offset of the first entry in phone book object
responseCode	The valid response codes are defined (in csr_bt_obex.h). For success in the request the code is CSR_BT_OBEX_SUCCESS_RESPONSE_CODE, any other response code indicates a failure.
newMissedCall	The number of missed calls that has not been checked yet since the PBAP session

	started.
bodyLength	The length of the body (object).
*body	The object itself. "body" is a CsrUint8 pointer to the object.
payloadLength	Number of bytes in the payload structure.
*payload	OBEX payload data. Offsets are relative to this pointer.
finalFlag	Indicates that the body (Object) fits the whole objects or that it is the last part.
smpOn	Reserved for future use. Set to FALSE.

4.16 CSR_BT_PAS_PULL_VCARD_MCH_LIST_SIZE

Primitives \ Parameters	type	connectionId	pbSize	newMissedCall	responseCode	smpOn
CSR_BT_PAS_PULL_VCARD_MCH_LIST_SIZE_IND	✓	✓				
CSR_BT_PAS_PULL_VCARD_MCH_LIST_SIZE_RES	✓	✓	✓	✓	✓	✓

Table 16: CSR_BT_PAS_PULL_VCARD_MCH_LIST_SIZE Primitives

Description

This signal is used for getting the size of the Missed Call history object. The application must support the PBAP virtual folders structure as specified in [PBAP]. The content of the folder must be sent in the vCard-Listing format specified in, ref. [PBAP].

Parameters

type	Signal identity, CSR_BT_PAS_PULL_VCARD_MCH_LIST_SIZE_IND/RES.
connectionId	The connection Id for this session, the PAS client will use this Id in the request.
responseCode	The valid response codes are defined (in csr_bt_obex.h). For success in the request the code is CSR_BT_OBEX_SUCCESS_RESPONSE_CODE, any other response code indicates a failure.
pbSize	The number of entries in the phone book.
newMissedCall	The number of missed calls that has not been checked yet since the PBAP session started.
smpOn	Reserved for future use. Set to FALSE.

4.17 CSR_BT_PAS_PULL_VCARD_ENTRY

Parameters Primitives	type	connectionId	ucs2nameOffset	filter[8]	format	responseCode	*body	bodyLength	payloadLength	*payload	finalFlag	smpOn
CSR_BT_PAS_PULL_VCARD_ENTRY_IND	✓	✓	✓	✓	✓				✓	✓		
CSR_BT_PAS_PULL_VCARD_ENTRY_RES	✓	✓				✓					✓	✓
CSR_BT_PAS_NEXT_IND	✓	✓										
CSR_BT_PAS_NEXT_RES	✓	✓				✓	✓	✓			✓	✓

Table 17: CSR_BT_PAS_PULL_VCARD_ENTRY Primitives

Description

This signal is used for retrieving a single phone book entry from the application. In a normal scenario will navigate through the phone book using CSR_BT_PAS_PULL_VCARD_LIST and CSR_BT_PAS_SET_FOLDER, and single phone book single entries using CSR_BT_PAS_PULL_VCARD_ENTRY. The signals CSR_BT_PAS_PULL_VCARD_ENTRY_IND/RES is used for transferring the information about the phone book object, where the CSR_BT_PAS_NEXT_IND/RES are used for transferring the actual phone book object. The last CSR_BT_PAS_NEXT_RES (or CSR_BT_PAS_PULL_VCARD_ENTRY_RES if there is no body) must set finalFlag TRUE to indicate that the end of the operation.

Parameters

type
Signal identity, CSR_BT_PAS_PULL_VCARD_ENTRY_IND/RES, CSR_BT_PAS_NEXT_IND/RES.

connectionId
The connection Id for this session, the PAS client will use this Id in the request.

ucs2nameOffset
Payload relative offset of a null terminated 16 bit Unicode text string (UCS2) containing the (file) name of the object.

The function “CsrUcs2ByteString2Utf8” can be used for converting a null terminated UCS2 text string into a null terminated UTF8 text string

filter[8]
Indicate the attributes requested by the client; see the detailed description in [PBAP].

The filter is a 64 bit long flag store in an array of 8 CsrUint8. The lower part of the flag; filter[0] contains bit 0 – 7, filter[1] contains bit 8 – 15, filter[2] contains bit 16 – 23, filter[3] contains bit 24 – 31, filter[4] contains bit 32 – 39, filter[5] contains bit 40 – 47, filter[6] contains bit 48 – 55 and filter[7] contains bit 56 – 63.

format
Used for indicating the requested format (vCard 2.1 or 3.0) to be returned in the operation.

responseCode
The valid response codes are defined (in csr_bt__obex.h). For success in the request the code is CSR_BT_OBEX_SUCCESS_RESPONSE_CODE, any other response code indicates a failure.

bodyLength
The length of the body (object).

*body
The object itself. "body" is a CsrUint8 pointer to the object.

payloadLength	Number of bytes in the payload structure.
*payload	OBEX payload data. Offsets are relative to this pointer.
finalFlag	Indicates that the body (Object) fits the whole objects or that it is the last part.
smpOn	Reserved for future use. Set to FALSE.

4.18 CSR_BT_PAS_ABORT

Parameters / Primitives	type	connectionId	descriptionOffset	descriptionLength	payloadLength	*payload
CSR_BT_PAS_ABORT_IND	✓	✓	✓	✓	✓	✓

Table 18: CSR_BT_PAS_ABORT Primitives

Description

This signal is indicating that the OBEX PBAP client has terminated an operation (such as PULL_PB), before it would normally end the session.

Parameters

type	Signal identity, CSR_BT_PAS_ABORT_IND.
connectionId	The connection Id for this session, the PBAP client will use this Id in the request.
descriptionOffset	Payload relative offset of a null terminated 16 bit Unicode text string (UCS2) containing the reason for the abort. The function “CsrUcs2ByteString2Utf8” can be used for converting a null terminated UCS2 text string into a null terminated UTF8 text string
descriptionLength	Length of the abort description string.
payloadLength	Number of bytes in the payload structure.
*payload	OBEX payload data. Offsets are relative to this pointer.

4.19 CSR_BT_PAS_DISCONNECT

Parameters Primitives	type	connectionId	deviceAddr	reasonCode	reasonSupplier
CSR_BT_PAS_DISCONNECT_IND	✓	✓	✓	✓	✓

Table 19: CSR_BT_PAS_DISCONNECT Primitives

Description

This signal is indicating that the OBEX file transfer session is finished, and is ready for a new one.

Parameters

type	Signal identity, CSR_BT_PAS_DISCONNECT_IND.
connectionId	The connection Id for this session, the PBAP client will use this Id in the request.
deviceAddr	The Bluetooth address which is connected to the device.
reasonCode	The reason code of the operation. Possible values depends on the value of reasonSupplier. If eg. the reasonSupplier == CSR_BT_SUPPLIER_CM then the possible reason codes can be found in csr_bt_cm_prim.h. All values which are currently not specified are the respective prim.h files or csr_bt_obex.h is regarded as reserved and the application should consider them as errors.
reasonSupplier	This parameter specifies the supplier of the reason given in reasonCode. Possible values can be found in csr_bt_result.h

4.20 CSR_BT_PAS_SECURITY_IN

Parameters / Primitives	type	appHandle	secLevel	resultCode	resultSupplier
CSR_BT_PAS_SECURITY_IN_REQ	✓	✓	✓		
CSR_BT_PAS_SECURITY_IN_CFM	✓			✓	✓

Table 20: CSR_BT_PAS_SECURITY_IN Primitives

Description

Applications that wish to change the enforcement to a specific profile security level, i.e. authentication, encryption and/or authorisation, can use this API to set up the security level for *new* connections. Note that this API is for the local device only and can be used from within any state.

The *CSR_BT_SECURITY_IN_REQ* signal sets up the security level for new incoming connections. Already established or pending connections are not altered.

Note, that any attempts to set security to a less secure level than the mandatory security level will be rejected. See csr_bt_profiles.h for mandatory security settings. The default settings used by CSR Synergy Bluetooth are set to require authentication and encryption.

Note that if MITM protection is requested and the remote device does not have the required IO capabilities, pairing/bonding will fail and connections to the remote device *cannot* be made. See [SC] for further details.

Parameters

type Signal identity CSR_BT_PAS_SECURITY_IN_REQ/CFM.

appHandle Application handle to which the confirm message is sent.

secLevel The application must specify one of the following values:

- CSR_BT_SEC_DEFAULT : Use default security settings
- CSR_BT_SEC_MANDATORY : Use mandatory security settings
- CSR_BT_SEC_SPECIFY : Specify new security settings

If CSR_BT_SEC_SPECIFY is set the following values can be OR'ed additionally:

- CSR_BT_SEC_AUTHORISATION: Require authorisation
- CSR_BT_SEC_AUTHENTICATION: Require authentication
- CSR_BT_SEC_ SEC_ENCRYPTION: Require encryption (implies authentication)
- CSR_BT_SEC_MITM: Require MITM protection (implies encryption)

resultCode The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.

resultSupplier This parameter specifies the supplier of the result given in resultCode. Possible values can be found in csr_bt_result.h

CSR Synergy Bluetooth 18.2.0 PAS API

5 Document References

Document	Reference
Phone Book Access Profile Version 1.0 27 April 2006	PBAP
IrDA Object Exchange Protocol - IrOBEX Version 1.3 3 January 2003	OBEX
Specifications for Ir Mobile Communications (IrMC) Version 1.1 01 March 1999	IRMC
CSR Synergy Bluetooth. CM – Connection Manager API Description, doc. no. api-0101-cm	CM
CSR Synergy Bluetooth, SC – Security Controller API Description, Document no. api-0102-sc	SC

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
OBEX	Object Exchange Protocol
PAC	OBEX Phone Book Access Profile Client side
PAS	OBEX Phone Book Access Profile Server side
PBAP	Phone Book Access Profile
SDS	Service Discovery Server
SIG	Special Interest Group
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.