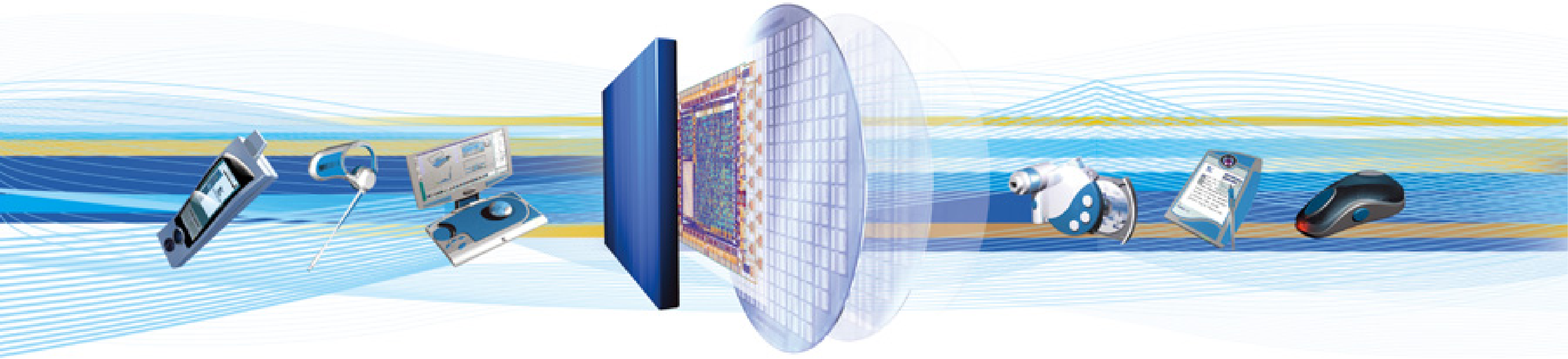
CSR Synergy®

CSR Synergy Bluetooth 18.2.0

OBEX Basic Printing Profile Server

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

Bluetooth®

Contents

List of Figures

List of Tables

1 Introduction

1.1 Introduction and Scope

This document describes the message interface provided by the OBEX Basic Printing Profile Server (BPPS). BPPS conforms to the server side of the Basic Printing Feature, ref. [BPP].

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. BPPS and the application
- The BPPS shall only handle one request at a time
- The client only authenticates BPPS doing a connect session

2 Description

2.1 Introduction

The scenario covered by this profile is the following:

- Usage of a Bluetooth® device e.g. a camera or a PDA that supports BPP client to send one or more files (images, documents etc.) to a Bluetooth® enabled printer supporting the BPP server

The OBEX Basic Printing Profile server (BPPS) must be activated by the application. When it is activated it is able to provide the application with the ability to receive objects for printing and the ability to retrieve referenced objects. Furthermore, it enables the application to communicate both printer and job attributes.

The BPPS provides Service Discovery handling.

The BPPS is handling the interpretation of the OBEX packet.

The application is responsible for handling the indications from the BPPS and sending the correct responses. The response codes used are described in the IrOBEX Specification [OBEX]. The BPPS 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. [BPP] and [OBEX].

2.2 Reference Model

The BPPS interfaces to the Connection Manager (CM).

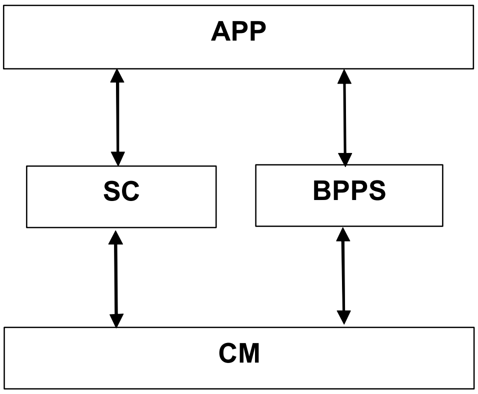


Figure 1: Reference model

2.3 Sequence Overview

BPPS starts up being in an IDLE state. When the application activates BPPS, the server enters an 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 an IDLE state.

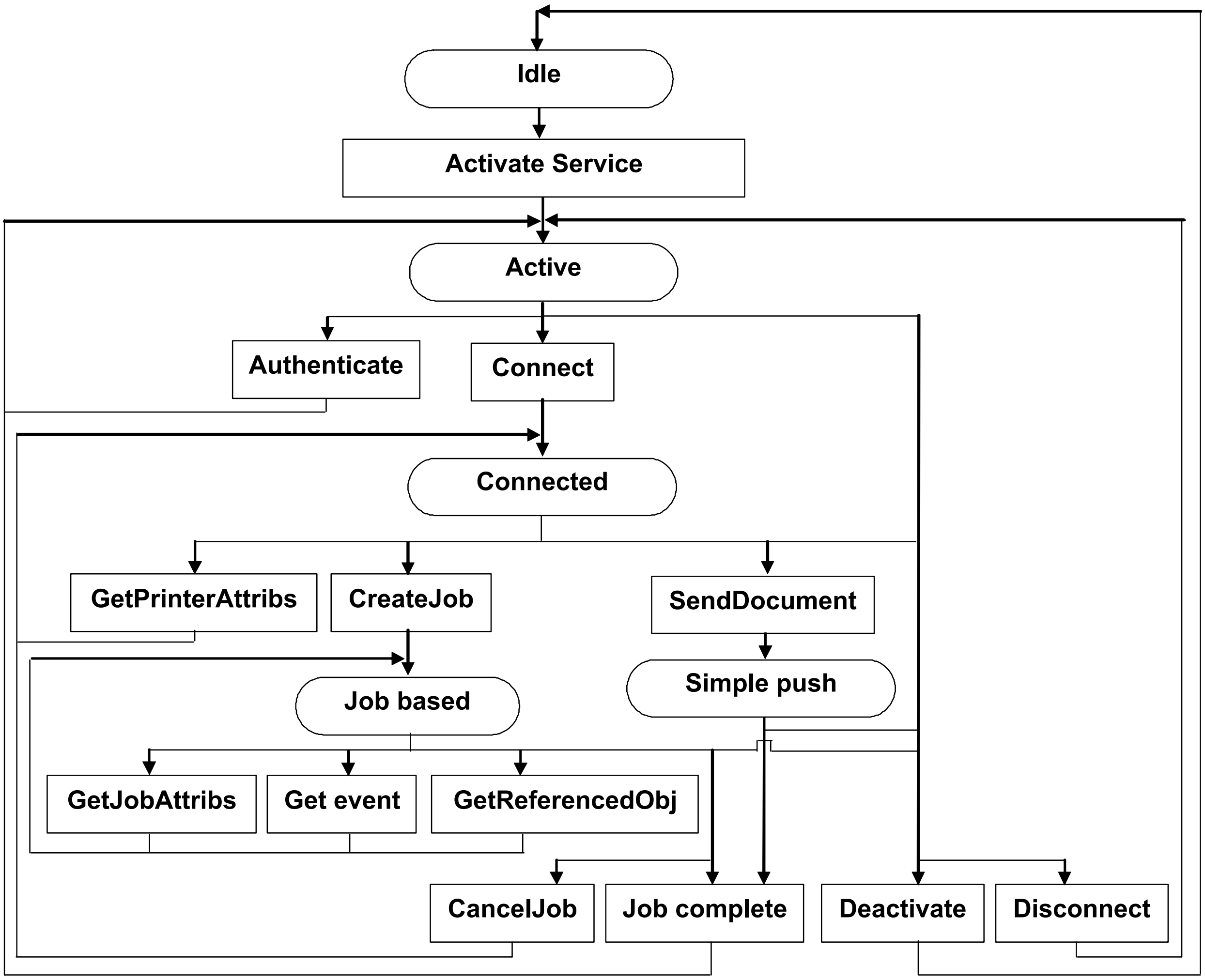


Figure 2: BPPS state diagram

3 Interface Description

3.1 Activation

Sending a CSR_BT_BPPS_ACTIVATE_REQ activates BPPS. BPPS then registers a Service Record in the Service Discovery Server and makes it connectable. When BPPS is ready to handle incoming requests, a CSR_BT_BPPS_ACTIVATE_CFM is sent to the application.

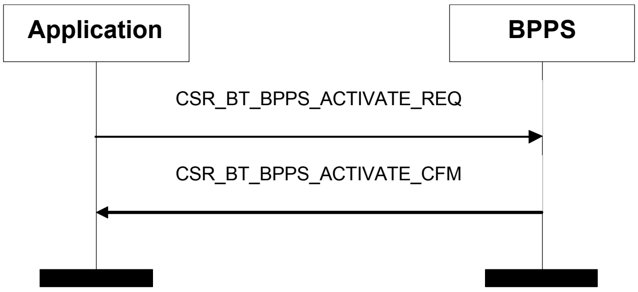


Figure 3: Activation

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 without Authentication

When the client is making an OBEX connect request against the server, the first message the application receives is CSR_BT_BPPS_CONNECT_IND, which the application must respond with a CSR_BT_BPPS_CONNECT_RES message with the appropriate result code.

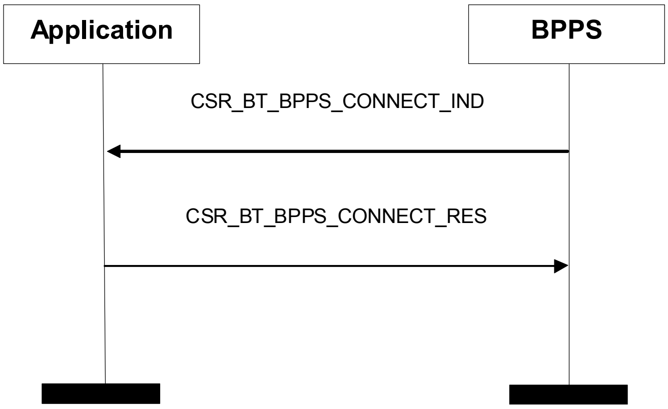


Figure 4: Connection handling without authentication

3.3 Connect with Authentication

If the client doing the OBEX connect request has authenticated the server, the application will receive a CSR_BT_BPPS_AUTHENTICATE_IND, which the application must respond with a CSR_BT_BPPS_AUTHENTICATE_RES message.

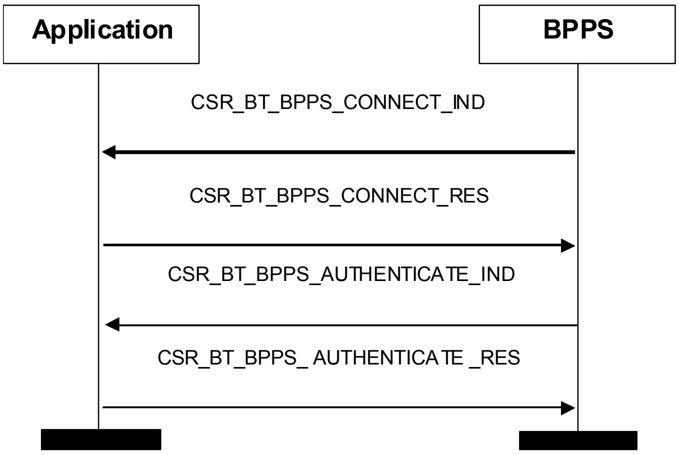


Figure 5: Connection handling with authentication from client

If the application requires that a client authenticates, a CSR_BT_BPPS_AUTHENTICATE_REQ can be sent to the BPPS, to which the BPPS responds with a CSR_BT_BPPS_AUTHENTICATE_CFM to the application. This is followed by a new CSR_BT_BPPS_CONNECT_IND which must be responded appropriately.

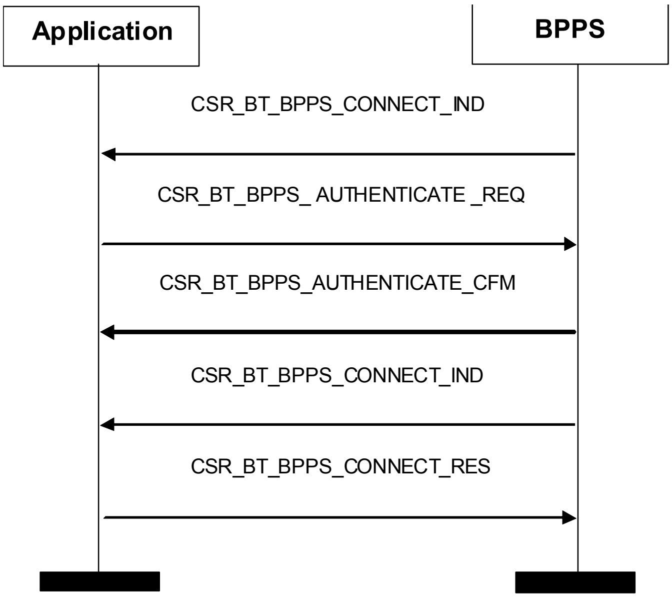


Figure 6: Connection handling with authentication from server

3.4 Get Printer Attributes

When the OBEX connection has been made the remote client can request the printer attributes. The request will be processed by BPPS and BPPS sends a CSR_BT_BPPS_GET_PRINTER_ATTRIBS_IND message to the application. The message will indicate which attributes the application must include in the CSR_BT_BPPS_GET_PRINTER_ATTRIBS_RES message.

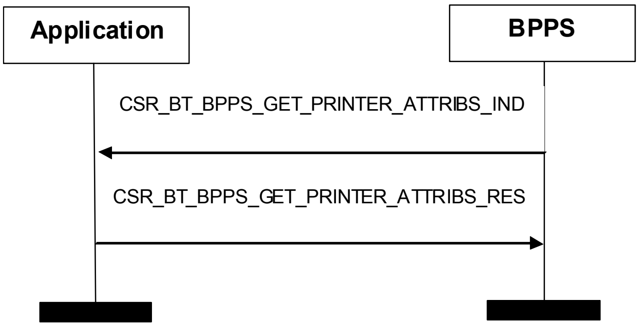

Application
BPPS
CSR_BT_BPPS_GET_PRINTER_ATTRIBS_IND
CSR_BT_BPPS_GET_PRINTER_ATTRIBS_RES

Figure 7: Get Printer Attributes Indication

3.5 Create Job

When an OBEX connection has been made the remote client can create a printer specific job by sending so that a document can be pushed to the printer using the job-based transfer method from a BPP client.

When the BPP client is making a create job request against the server, the application receives a CSR_BT_BPPS_CREATE_JOB _IND message. The application must respond with a CSR_BT_BPPS_CREATE_JOB_RES message with the appropriate job id number and printer operation status value.

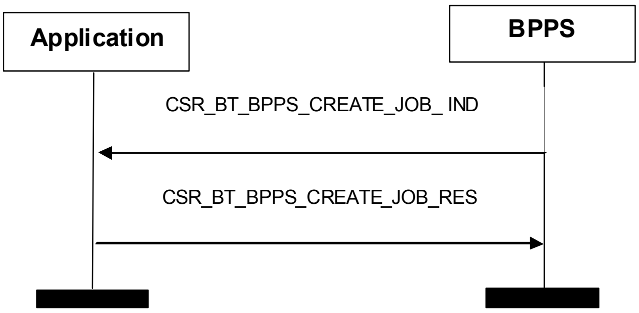


Figure 8: Create Job handling

3.6 Get Job Attributes

When the remote client has made an OBEX connection and created a job, the client can request the job attributes of that particular job to get the current status. This can only happen in conjunction with the BPP job-based transfer mechanism.

When the BPP client is making a get job attributes request against the server, the application receives a CSR_BT_BPPS_GET_JOB_ATTRIBS_IND message. The application must respond with a CSR_BT_BPPS_GET_JOB_ATTRIBS_RES message with the current job id, job state, job name, job's originating user name, job media sheets completed, number of intervening jobs and printer operation status.

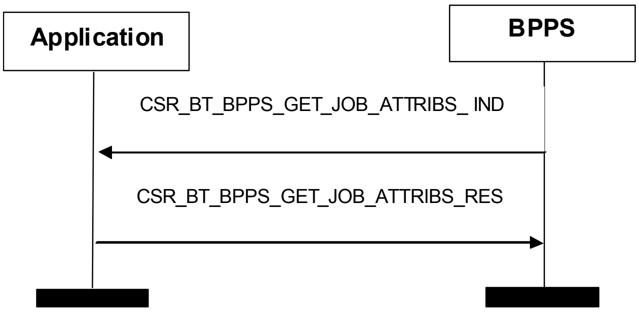

Application
BPPS
CSR_BT_BPPS_GET_JOB_ATTRIBS_ IND
CSR_BT_BPPS_GET_JOB_ATTRIBS_RES

Figure 9: Get Job Attributes handling

3.7 Cancel Job

When the remote client has made an OBEX connection and created a job, the client can cancel the current job. This is used in conjunction with the BPP job-based transfer mechanism.

When the BPP client is making a cancel job request against the server, the application receives a CSR_BT_BPPS_CANCEL_JOB_IND message. The application must respond with a CSR_BT_BPPS_CANCEL_JOB_RES message with the current job id that is being cancelled and printer operation status.

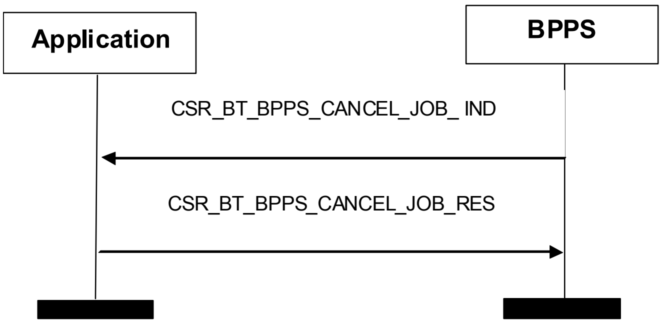


Figure 10: Cancel Job handling

3.8 Send Document

When the remote client has made an OBEX connection it can send a document for printing. The document will be delivered from BPPS to the application by a series of messages, starting with the CSR_BT_BPPS_SEND_DOCUMENT_IND message. The document to be printed can be divided into fragments and delivered in multiple messages. The BPPS will still start the sequence with the CSR_BT_BPPS_SEND_DOCUMENT_IND message. When the BPPS receives additional fragments of the document it will send CSR_BT_BPPS_NEXT_IND messages to the application. The application must respond with a CSR_BT_BPPS_NEXT_RES message when it has received the CSR_BT_BPPS_NEXT_IND message. To signal that the last part of the data has been sent, the CSR_BT_BPPS_NEXT_IND has its *finalFlag* parameter set to TRUE.

The files transferred can be in any format and it is the application's duty to process the file appropriately.

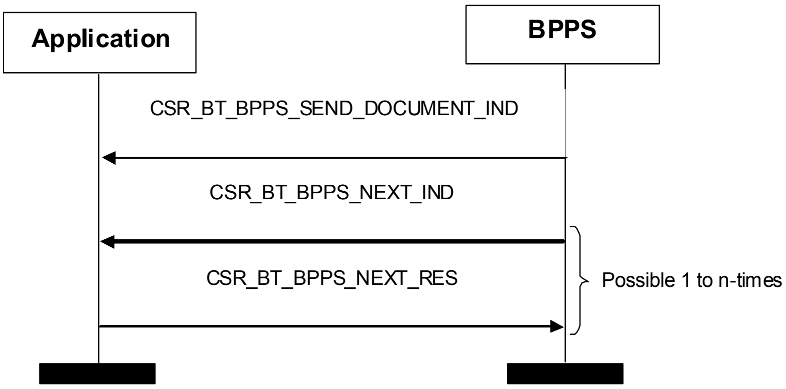


Figure 11: Send Document handling

3.9 Get Referenced Object

If the application encounters a referenced object it can send a CSR_BT_BPPS_GET_REFERENCED_OBJ_REQ message to BPPS. The application has the option to request either the entire referenced object or only a fragment of the referenced object. If the entire object must be retrieved count should be set to "-1" else count is used for indicating the amount of bytes to retrieve in the fragment. The support for retrieving only a fragment of the referenced object enables the application to retrieve parts of different referenced objects in a multiplexing manner. The requested fragments will be delivered to the application by the CSR_BT_BPPS_GET_REFERENCED_OBJ_IND message and the CSR_BT_BPPS_GET_REFERENCED_OBJ_CFM message when the last part of the fragment arrives. The application must respond with a CSR_BT_BPPS_GET_REFERENCED_OBJ_RES message to the CSR_BT_BPPS_GET_REFERENCED_OBJ_IND message, when it is ready to retrieve the next part of the referenced object fragment. The CSR_BT_BPPS_GET_REFERENCED_OBJ_IND/ CSR_BT_BPPS_GET_REFERENCED_OBJ_RES sequence can occur from zero to n times, depending on the number of OBEX packets needed to transfer the requested object or object fragment.

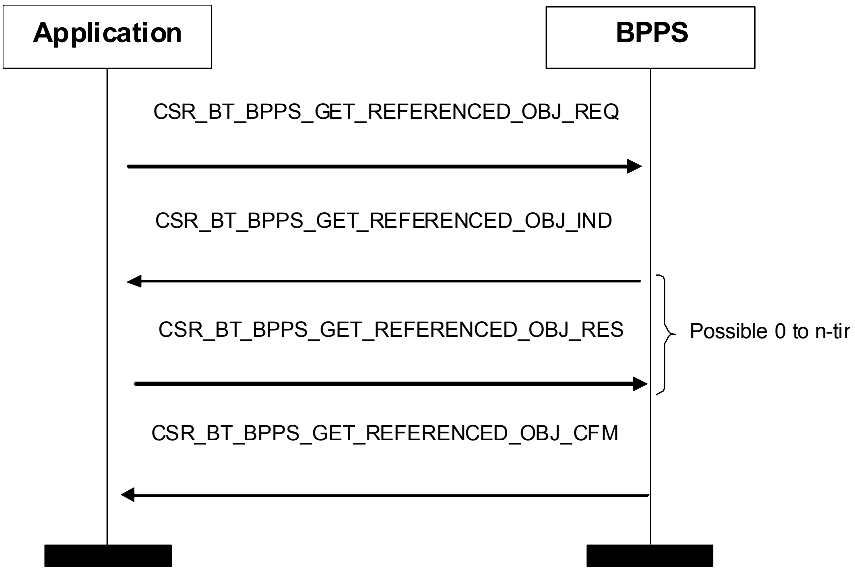

Application
BPPS
CSR_BT_BPPS_GET_REFERENCED_OBJ_REQ
CSR_BT_BPPS_GET_REFERENCED_OBJ_IND
CSR_BT_BPPS_GET_REFERENCED_OBJ_RES
Possible 0 to n-tir
CSR_BT_BPPS_GET_REFERENCED_OBJ_CFM

Figure 12: Get referenced object handling

3.10 Get Event

The remote client has the option to request to be updated every time the print job or printer changes state. When this is requested BPPS will signal to the application that it should inform BPPS about any state changes of the printer and the print job. BPPS uses the CSR_BT_BPPS_GET_EVENT_IND message to tell the application when to start and stop sending state updates. When the application has been instructed to send update, it should send these updates in CSR_BT_BPPS_GET_EVENT_RES messages.

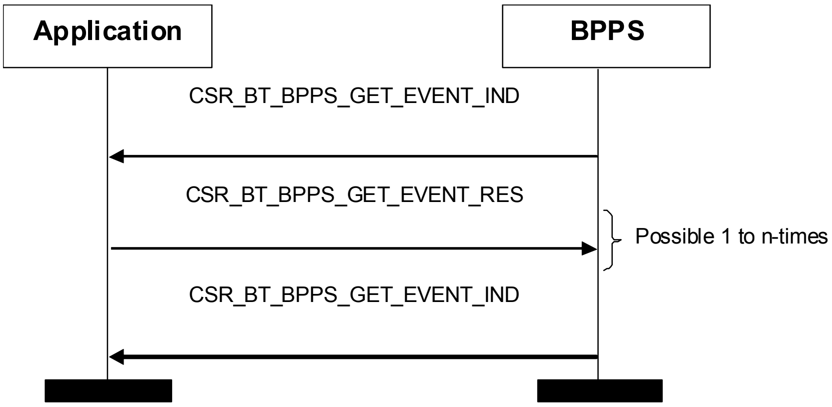

Application
BPPS
CSR_BT_BPPS_GET_EVENT_IND
CSR_BT_BPPS_GET_EVENT_RES
Possible 1 to n-times
CSR_BT_BPPS_GET_EVENT_IND

Figure 13: Get Event handling

3.11 Job Complete

When the application has completed a print job it must inform BPPS. This is done by the CSR_BT_BPPS_JOB_COMPLETE_REQ message. Upon reception of that message the BPPS will initiate an internal job completion routine and prepare to receive a new OBEX session. BPPS will respond with a CSR_BT_BPPS_JOB_COMPLETE_CFM message when it has completed the internal preparation for new sessions.

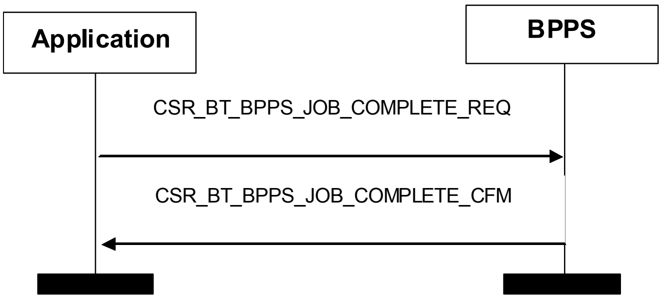


Figure 14: Job Complete handling

3.12 Abort Handling

The orderly sequence of request (from an OBEX client) followed by response (from an OBEX server) has one exception. An abort operation may come in the middle of a request/response sequence. It cancels the current operation. If the client makes an abort request in order to terminate a multi-packet operation before it normally ends, BPPS will send a CSR_BT_BPPS_ABORT_IND message to the application.

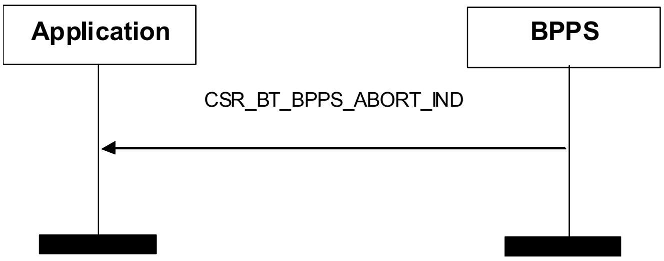


Figure 15: Abort handling

3.13 Disconnect

The CSR_BT_BPPS_DISCONNECT_IND message signals the end of an OBEX session.

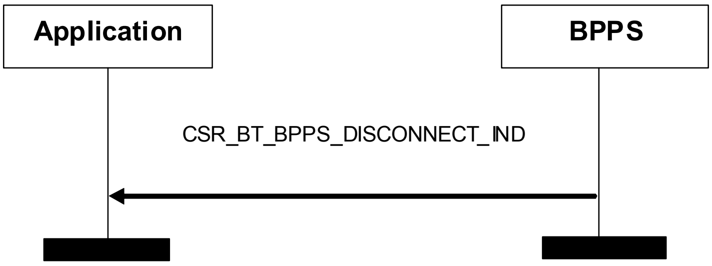


Figure 16: Disconnect

3.14 Deactivation

Sending a CSR_BT_BPPS_DEACTIVATION_REQ message deactivates BPPS. This procedure may take some time depending on the current BPPS activity. When deactivated, the BPPS confirms a CSR_BT_BPPS_DEACTIVATE_CFM message.

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

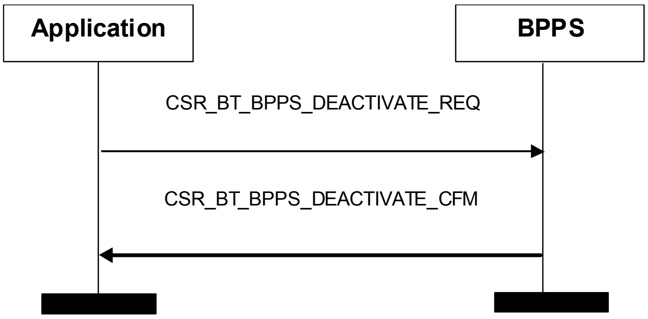


Figure 17: Deactivation request handling

3.15 Payload Encapsulated Data

3.15.1 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.

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.

3.15.2 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.

Likewise the profile will free any pointers received as parameters in an API function or primitive.

4 OBEX Basic Printing Profile Responder Primitives

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

4.1 List of All Primitives

Primitives:	Reference:
CSR_BT_BPPS_ACTIVATE_REQ	See section 4.2
CSR_BT_BPPS_ACTIVATE_CFM	See section 4.2
CSR_BT_BPPS_DEACTIVATE_REQ	See section 0
CSR_BT_BPPS_DEACTIVATE_CFM	See section 0
CSR_BT_BPPS_CONNECT_IND	See section 4.4
CSR_BT_BPPS_CONNECT_RES	See section 4.4
CSR_BT_BPPS_AUTHENTICATE_REQ	See section 4.5
CSR_BT_BPPS_AUTHENTICATE_RES	See section 4.5
CSR_BT_BPPS_AUTHENTICATE_CFM	See section 4.5
CSR_BT_BPPS_AUTHENTICATE_IND	See section 4.5
CSR_BT_BPPS_NEXT_IND	See section 4.7
CSR_BT_BPPS_NEXT_RES	See section 4.7
CSR_BT_BPPS_GET_PRINTER_ATTRIBS_IND	See section 4.7
CSR_BT_BPPS_GET_PRINTER_ATTRIBS_REQ	See section 4.7
CSR_BT_BPPS_GET_PRINTER_ATTRIBS_RES	See section 4.8
CSR_BT_BPPS_CREATE_JOB_IND	See section 4.9
CSR_BT_BPPS_CREATE_JOB_REQ	See section 4.9
CSR_BT_BPPS_CREATE_JOB_RES	See section 4.9
CSR_BT_BPPS_GET_JOB_ATTRIBS_IND	See section 4.10
CSR_BT_BPPS_GET_JOB_ATTRIBS_REQ	See section 4.10
CSR_BT_BPPS_GET_JOB_ATTRIBS_RES	See section 4.10
CSR_BT_BPPS_CANCEL_JOB_IND	See section 4.11
CSR_BT_BPPS_CANCEL_JOB_REQ	See section 4.11
CSR_BT_BPPS_CANCEL_JOB_RES	See section 4.11
CSR_BT_BPPS_SEND_DOCUMENT_IND	See section 4.12
CSR_BT_BPPS_ABORT_IND	See section 4.13
CSR_BT_BPPS_DISCONNECT_IND	See section 4.14
CSR_BT_BPPS_GET_REFERENCED_OBJ_REQ	See section 4.15
CSR_BT_BPPS_GET_REFERENCED_OBJ_IND	See section 4.15
CSR_BT_BPPS_GET_REFERENCED_OBJ_RES	See section 4.15
CSR_BT_BPPS_GET_REFERENCED_OBJ_CFM	See section 4.15
CSR_BT_BPPS_GET_EVENT_IND	See section 4.16
CSR_BT_BPPS_GET_EVENT_RES	See section 4.16
CSR_BT_BPPS_JOB_ COMPLETE_REQ	See section 0
CSR_BT_BPPS_JOB_ COMPLETE_CFM	See section 0
CSR_BT_BPPS_SECURITY_IN_REQ	See section 4.18

CSR_BT_BPPS_SECURITY_IN_CFM	See section 4.18

Table 1: List of all primitives

4.2 CSR_BT_BPPS_ACTIVATE

Parameters Primitives	type	appHandle	*documetFormatsSupported	characterRepertoiresSupportedLength	*characterRepertoiresSupported	*imageFormatsSupported	*ieee1284Id	obexMaxPacketSize	windowSize	srmEnable
CSR_BT_BPPS_ACTIVATE_REQ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CSR_BT_BPPS_ACTIVATE_CFM	✓									

Table 2: CSR_BT_BPPS_ACTIVATE Primitives

Description

This signal is used for activating BPPS and making it accessible from a remote device. The process includes:

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

BPPS will remain activated until a CSR_BT_BPPS_DEACTIVATE_REQ is received.

Parameters

type
Signal identity, CSR_BT_BPPS_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.

*documentFormatsSupported
Appended to the Service Discovery Record before it is registered with the Connection Manager (CM). This utf-8 string describes every data format, page description language (PDL) and image format that is supported by the printer in US-ASCII.

For a detailed description of this value refer to the Basic Printing Profile (BPP) specification section 12.2.2 entitled “Document Formats Supported”.

characterRepertoiresSupportedLength
Length of the characterRepertoiresSupported parameter in bytes.

*characterRepertoiresSupported
Appended to the Service Discovery Record before it is registered with the Connection Manager (CM). This 128 byte value enables a sender to determine which characters or glyphs a printer supports for access from XHTML-Print and the optional Basic Text, vCard, vCalendar and vMessage formats. Each bit in this 128 byte value that is set to 1 indicates that all characters of the corresponding character repertoire can be printed. A bit set to 0 indicates that a part or all of the characters in the corresponding character repertoire cannot be printed.

For a detailed description of this value please refer to the Basic Printing Profile [BPP] specification section 12.2.2 entitled “Document Formats Supported”.

*imageFormatsSupported

Append to the Service Discovery Record before it is registered with the Connection Manager (CM). This utf-8 string describes the XHTML-Print image formats that are supported by the printer in US-ASCII.

For a detailed description of this value please refer to the Basic Printing Profile [BPP] specification section 12.2.2 entitled “Document Formats Supported”.

*ieee1284Id

Appended to the Service Discovery Record before it is registered with the Connection Manager (CM). This value contains the IEEE 1284 printer identification ASCII string that describes the particular model of printer.

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_BPPS_DEACTIVATE

Parameters Primitives	type
CSR_BT_BPPS_DEACTIVATE_REQ	✓
CSR_BT_BPPS_DEACTIVATE_CFM	✓

Table 3: CSR_BT_BPPS_DEACTIVATE Primitives

Description

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

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

The signal will stop any ongoing transaction.

Parameters

type	Signal identity, CSR_BT_BPPS_DEACTIVATE_REQ/CFM.

4.4 CSR_BT_BPPS_CONNECT

Parameters / Primitives	type	connectionId	obexPeerMaxPacketSize	deviceAddr	responseCode	length	count	btConnId
CSR_BT_BPPS_CONNECT_IND	✓	✓	✓	✓		✓	✓	✓
CSR_BT_BPPS_CONNECT_RES	✓	✓			✓			

Table 4: CSR_BT_BPPS_CONNECT Primitives

Description

This signal is indicating that the BPP client is starting a basic printing session. The application can accept or deny the request and has to return the connectionId received in the indication.

Parameters

type — Signal identity, CSR_BT_BPPS_CONNECT_IND/RES.

connectionId — The connection Id tells the recipient of the request which OBEX connection this request belongs to. The connection Id received in the indication must be returned in the response.

obexPeerMaxPacketSize — Indicates the maximum size OBEX packet that is allowed to send to the client.

deviceAddr — The Bluetooth address which is connected to the device

responseCode — For accepting an OBEX connection the code is:
CSR_BT_OBEX_SUCCESS_RESPONSE_CODE.

The following response codes reject the OBEX connection request:
CSR_BT_OBEX_SERVICE_UNAVAILABLE_RESPONSE_CODE
CSR_BT_OBEX_BAD_REQUEST_RESPONSE_CODE
CSR_BT_OBEX_FORBIDDEN_RESPONSE_CODE

The responseCodes are defined in (csr_bt_obex.h) with the following type CsrBtObexResponseCode and can also be found in IrDA Object Exchange Protocol.

The meaning of the response codes for the Basic Printing Profile is described in [BPP].

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_BPPS_AUTHENTICATE

Parameters Primitives	type	options	realmLength	* realm	deviceAddr	passwordLength	*password	*userId	resultCode	resultSupplier
CSR_BT_BPPS_AUTHENTICATE_IND	✓	✓	✓	✓	✓					
CSR_BT_BPPS_AUTHENTICATE_REQ	✓		✓	✓		✓	✓	✓		
CSR_BT_BPPS_AUTHENTICATE_RES	✓					✓	✓	✓		
CSR_BT_BPPS_AUTHENTICATE_CFM	✓								✓	✓

Table 5: CSR_BT_BPPS_AUTHENTICATE Primitives

Description

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

Parameters

type

Signal identity, CSR_BT_BPPS_AUTHENTICATE_IND/REQ/RES/CFM.

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 arguments for_BT_BPPS_AUTHENTICATE_RES message. If bit 0 is not set the application can just zero terminated the first character of the user Id string.

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_BPPS_AUTHENTICATE_IND and in the CSR_BT_BPPS_AUTHENTICATE_REQ the 'realmLength' 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_BPPS_AUTHENTICATE_IND and in the CSR_BT_BPPS_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

passwordLength
The length of the response password.

*password
Containing the response password of the OBEX authentication. This is a pointer which shall be allocated by the application.

*userId
Zero terminated string (ASCII) containing the userId for the authentication.
This is a pointer which shall be allocated by the application.
Note in the CSR_BT_BPPS_AUTHENTICATE_REQ the userId is ignored right now.

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. If the resultSupplier == CSR_BT_SUPPLIER_OBEX then the possible result codes can be found in csr_bt_obex.h. All values which are currently not specified in the respective prim.h files or csr_bt_obex.h 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

4.6 CSR_BT_BPPS_NEXT

Parameters / Primitives	type	connectionId	finalFlag	*body	bodyLength	smpOn
CSR_BT_BPPS_NEXT_RES	✓	✓				✓
CSR_BT_BPPS_NEXT_IND	✓	✓	✓	✓	✓	

Table 6: CSR_BT_BPPS_NEXT Primitives

Description

If the document being sent to the BPPS cannot fit into a single CSR_BT_BPPS_SEND_DOCUMENT_IND message, it must be fragmented into one CSR_BT_BPPS_SEND_DOCUMENT_IND message and one or more CSR_BT_BPPS_NEXT messages.

The application will receive a CSR_BT_BPPS_NEXT_IND when the BPPS has received more data. The application must respond wit a CSR_BT_BPPS_NEXT_RES message.

To signal that the last part of the data has been sent, the *finalFlag* must be set to TRUE.

Parameters

type	Signal identity, CSR_BT_BPPS_NEXT_IND/RES.
connectionId	The connection Id tells the recipient of the request which OBEX connection this request belongs to. The connection Id received in the indication must be returned in the response.
finalFlag	This flag is set if the message contains the last part of the data being sent.
*body	The actual data. This may be the whole or part of the entire data being transferred.
bodyLength	Length in bytes of the body pointer.
smpOn	Reserved for future use. Set to FALSE.

4.7 CSR_BT_BPPS_GET_PRINTER_ATTRIBS_REQ + _IND

Parameters Primitives	type	printerName	printerLocation	printerState	printerStateReasons	documentFormatsSupported	colorSupported	maxCopiesSupported	sidesSupported	numberUpSupported	orientationsSupported	mediaSizesSupported	mediaTypesSupported	mediaLoaded	printQualitySupported	queuedJobCount	imageFormatsSupported	basicTextPageWidth	basicTextPageHeight	printerGeneralCurrentOperator
CSR_BT_BPPS_GET_PRINTER_ATTRIBS_IND	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 7: CSR_BT_BPPS_GET_PRINTER_ATTRIBS_REQ + _IND Primitives

Description

The printer attributes request is sent from the client to the BPP server in the form of an OBEX SOAP encoded string. Within this SOAP encoded string are all the individual printer attributes that are requested. This string is decoded within the BPP server and for each attribute a Boolean value is passed to the application in the CSR_BT_BPPS_GET_PRINTER_ATTRIBS_IND.

For a detailed description of the Get Printer Attributes request and response please refer to the Basic Printing Profile [BPP] specification section 7.1.1 entitled “GetPrinterAttributes”.

Parameters

type — Signal identity, CSR_BT_BPPS_GET_PRINTER_ATTRIBS_IND/REQ.

printerName — If the value is TRUE, the application must respond with a string value that holds the administratively assigned user friendly name of the printer.

printerLocation — If the value is TRUE, the application must respond with a string value that indicates the location of the device.

printerState — If the value is TRUE, the application must respond with a string value that identifies the current state of the printer. The values held by this string can be “idle”, “processing” and “stopped”.

printerStateReasons — If the value is TRUE, the application must respond with a string value that indicates additional information about why the printer is in its current state.

documentFormats — If the value is TRUE, the application must respond with an array of strings values that specifies the document formats supported as MIME media types.

colorSupported — If the value is TRUE, the application must respond with a Boolean true or false value to indicate whether or not the device is capable of full colour printing.

maxCopiesSupported — If the value is TRUE, the application must respond with an integer value that holds a number which identifies the largest value of the Copies parameter that the printer supports in the CreateJob actions.

sidesSupported — If the value is TRUE, the application must respond with a string value that identifies the set of values of the sides parameter that the printer supports in the CreateJob action.

numberUpSupported — If the value is TRUE, the application must respond with an integer value that identifies the maximum value of the NumberUp parameter that the printer supports in the CreateJob action.

orientationsSupported
If the value is TRUE, the application must respond with a string value that identifies the set of values of the OrientationRequested parameter that the printer supports in CreateJob actions.

mediaSizesSupported
If the value is TRUE, the application must respond with and array of string values that identifies the set of all values of the MediaSize parameter that the printer can support in CreateJob operations.

mediaTypesSupported
If the value is TRUE, the application must respond with an array of string values that identifies the types of paper that the printer can support.

mediaLoaded
If the value is TRUE, the application must respond with an array of arrays which identifies a collection of medium type/size pairs that are currently loaded in the printer.

printQualitySupported
If the value is TRUE, the application must respond with an array of string values that identifies the set of values of the PrintQuality parameter that the printer supports in CreateJob actions.

queuedJobCount
If the value is TRUE, the application must respond with an integer value that indicates the number of jobs in the printer's queue.

imageFormatsSupported
If the value is TRUE, the application must respond with an array of strings values that identifies the types of images that are supported when embedded in an XHTML-Print document.

basicTextPageWidth
If the value is TRUE, the application must respond with an integer value which holds the number of characters of the default mono-spaced font that will fit the width of the page for the default paper size. This value is applicable for XHTML-Print and Basic Text document formats.

basicTextPageHeight
If the value is TRUE, the application must respond with an integer value which holds the number of characters of the default mono-spaced font that will fit the height of the page for the default paper size. This value is applicable for XHTML-Print and Basic Text document formats.

printerGeneralCurrentOperator
If the value is TRUE, the application must respond with a string value that provides general information about who to contact (and how) to report printer support requirements.

4.8 CSR_BT_BPPS_GET_PRINTER_ATTRIBS_RES

Parameters / Primitives	type	*printerName	*printerLocation	*printerState	*printerStateReasons	*colorSupported	*maxCopiesSupported	*numberUpSupported	*mediaLoaded	mediaLoadedLength	mediaLoadedCount	*queuedJobCount	*basicTextPageWidth	*basicTextPageHeight	*printerGeneralCurrentOperator	*documentFormatsSupported	smpOn
CSR_BT_BPPS_GET_PRINTER_ATTRIBS_RES	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 8: CSR_BT_BPPS_GET_PRINTER_ATTRIBS_RES Primitive

Parameters / Primitives	documentFormatsSupportedCount	documentFormatsSupportedLength	*sidesSupported	sidesSupportedCount	sidesSupportedLength	*orientationsSupported	orientationsSupportedCount	orientationsSupportedLength	*mediaSizesSupported	mediaSizesSupportedCount	mediaSizesSupportedLength	*mediaTypesSupported	mediaTypesSupportedCount	mediaTypesSupportedLength	*printQualitySupported	printQualitySupportedCount	printQualitySupportedLength	*imageFormatsSupported	imageFormatsSupportedCount	imageFormatsSupportedLength	operationStatus
CSR_BT_BPPS_GET_PRINTER_ATTRIBS_RES	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 9: CSR_BT_BPPS_GET_PRINTER_ATTRIBS_RES Primitive, continued

Description

After reception of a CSR_BT_BPPS_GET_PRINTER_ATTRIBS_IND the application must respond with a CSR_BT_BPPS_GET_PRINTER_ATTRIBS_RES message. Depending on the values of the indication signal, the different values in the response signal must be filled.

For a detailed description of the Get Printer Attributes request and response please refer to the Basic Printing Profile (BPP) specification section 7.1.1 entitled “GetPrinterAttributes”.

Parameters

type	Signal identity, CSR_BT_BPPS_GET_PRINTER_ATTRIBS_RES.
*printerName	Assigned user friendly name of the printer.
*printerLocation	String value that indicates the location of the device.
*printerState	String value that identifies the current state of the printer. The values held by this string can be “idle”, “processing” and “stopped”.
*printerStateReasons	String value that indicates additional information about why the printer is in its current state.

*documentFormatsSupported	If requested by the client, the application must respond with an array of strings values that specifies the document formats supported as MIME media types.
documentFormatsSupportedCount	Number of strings in documentFormats.
documentFormatsSupportedLength	Total length of documentFormat
*colorSupported	If requested by the client, the application must respond with a string holding true or false to indicate whether or not the device is capable of full color printing.
*maxCopiesSupported	If requested by the client, the application must respond with a string value that holds a number which identities the largest value of the Copies parameter that the printer supports in the CreateJob actions.
*sidesSupported	If requested by the client, the application must respond with a string value that identifies the set of values of the sides parameter that the printer supports in the CreateJob action.
sidesSupportedCount	Number of strings in sidesSupported.
sidesSupportedLength	Total length of sidesSupported
*numberUpSupported	If requested by the client, the application must respond with an integer value that identifies the maximum value of the NumberUp parameter that the printer supports in the CreateJob action.
*orientationsSupported	If requested by the client, the application must respond with a string value that identifies the set of values of the OrientationRequested parameter that the printer supports in CreateJob actions.
orientationsSupportedCount	Number of strings in orientationsSupported
orientationsSupportedLength	Total length of orientationsSupported
*mediaSizesSupported	If requested by the client, the application must respond with and array of string values that identifies the set of all values of the MediaSize parameter that the printer can support in CreateJob operations.
mediaSizesSupportedCount	Number of strings in mediaSizesSupported
mediaSizesSupportedLength	Total length of mediaSizesSupported
*mediaTypesSupported	If requested by the client, the application must respond with an array of string values that identifies the types of paper that the printer can support.
mediaTypesSupportedCount	Number of strings in mediaTypesSupported
mediaTypesSupportedLength	Total length of mediaTypesSupported
*mediaLoaded	If requested by the client, the application must respond with a string which identifies a collection of medium type/size pairs that are currently loaded in the printer.
mediaLoadedCount	Number of strings in mediaLoadedCount.
mediaLoadedLength	Total length of mediaLoadedCount
*printQualitySupported	If requested by the client, the application must with an array of string values that identifies the set of values of the PrintQuality parameter that the printer supports in CreateJob actions.
printQualitySupportedCount	Number of strings in printQualitySupported
printQualitySupportedLength	Total length of printQualitySupported

*queuedJobCount	If requested by the client, the application must respond with a string that indicates the number of jobs in the printer's queue.
*imageFormatsSupported	If requested by the client, the application must respond with an array of strings values that identifies the types of images that are supported when embedded in an XHTML-Print document.
imageFormatsSupportedCount	Number of strings in imageFormatsSupported
imageFormatsSupportedLength	Total length of imageFormatsSupported
*basicTextPageWidth	If requested by the client, the application must respond with a string which holds the number of characters of the default mono-spaced font that will fit the width of the page for the default paper size. This value is applicable for XHTML-Print and Basic Text document formats.
*basicTextPageHeight	If requested by the client, the application must respond with a string which holds the number of characters of the default mono-spaced font that will fit the height of the page for the default paper size. This value is applicable for XHTML-Print and Basic Text document formats.
*printerGeneralCurrentOperator	If requested by the client, the application must respond with a string value that provides general information about who to contact (and how) to report printer support requirements.
operationStatus	OperationStatus of the get printer attrib operation. The operationStatus are defined in (csr_bt_obex.h) with the following type CsrBtObexOperationStatusCode and can also be found in the Basic Printing Profile [BPP] specification section 16.2 entitled "Operation Status Codes" for defined code values.
smpOn	Reserved for future use. Set to FALSE.

4.9 CSR_BT_BPPS_CREATE_JOB

Primitives \ Parameters	type	*jobNmae	*jobOriginatingUserName	*documentFormat	*copies	*sides	*numberUp	*orientationRequested	*mediaSize	*mediaType	*printQuality	*cancelOnLostLink	jobId	operationStatus	smpOn
CSR_BT_BPPS_CREATE_JOB_IND	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
CSR_BT_BPPS_CREATE_JOB_RES	✓												✓	✓	✓

Table 10: CSR_BT_BPPS_CREATE_JOB Primitives

Description

This Indication and response signal is used when the remote client is requested to create a new printer job as part of the job-based transfer method for pushing a document from the client to the BPP server. The create job information is sent from the client to the BPP server in the form of an OBEX SOAP encoded string. Within this SOAP encoded string are all the individual create job attributes that are required by the server. This string is decoded within the BPP server and each value is passed on to the application where it may be used for processing the new printer job. The results of this action are passed back to the BPP server where the SOAP encoded string is constructed for the response sent back to the client.

For a detailed description of the Create Job indication and response please refer to the Basic Printing Profile [BPP] specification section 7.1.2 entitled “CreateJob”.

Parameters

type — Signal identity, CSR_BT_BPPS_CREATE_JOB_IND/REQ/RES.

*jobName — String value that specifies the user-friendly name of the job being created.

*jobOriginatingUserName — String value that specifies the name or identifier of the user that submitted the job to be created. This name is either supplied by the client or by the security infrastructure if it exists.

*documentFormat — String value that specifies the document format of the job being created as a MIME media-type and any applicable version.

*copies — String that specifies the number of copies of the job being created that is to be printed.

*sides — String value that specifies how pages are to be imposed upon the sides of a selected medium for the job being created. Values for this parameter can be “one-sided”, “two-sided-long-edge” or “two-sided-short-edge”.

*numberUp — String that indicates the number of print-stream pages to impose upon a single side of an instance of a selected medium for the job being created.

*orientationRequested — String value that indicates the desired orientation for printed pages of the job being created. Values for this parameter can be “portrait”, “landscape”, “reverse-landscape” and “reverse-portrait”.

*mediaSize — String value that identifies the size of medium to for the job being created.

*mediaType — String value that identifies the type of medium to be used for the job being created.

*printQuality String value that specifies the print quality requested for the job being created. Values for this parameter can be “draft”, “normal” and “high”.

*cancelOnLostLink String holding true or false which indicates whether the job being created shall be cancelled by the printer if the Bluetooth radio link is lost or closed.

jobId Integer value that specifies the job identifier of the job that has now been created and for which the printer can accept print data in a subsequent SendDocument command. Note that the jobId value will be used in the subsequent SendDocument command.

operationStatus Status of the CreateJob operation.
The operationStatus are defined in (csr_bt_obex.h) with the following type CsrBtObexOperationStatusCode and can also be found in the Basic Printing Profile [BPP] specification section 16.2 entitled “Operation Status Codes” for defined code values.

smpOn Reserved for future use. Set to FALSE.

4.10 CSR_BT_BPPS_GET_JOB_ATTRIBS

Parameters Primitives	type	jobId	*jobAttributes	jobAttribsCount	jobAttribsLength	*jobState	*jobName	*jobOriginatingUserName	jobMediaSheetsCompleted	numberOfInterveningJobs	operationStatus	smpOn
CSR_BT_BPPS_GET_JOB_ATTRIBS_IND	✓	✓	✓	✓	✓							
CSR_BT_BPPS_GET_JOB_ATTRIBS_RES	✓	✓				✓	✓	✓	✓	✓	✓	✓

Table 11: CSR_BT_BPPS_GET_JOB_ATTRIBS Primitives

Description

This indication and response signal is used when the client has requested to get attributes of the current printer job that is currently in existence. The get job attributes are sent from the client to the BPP server in the form of an OBEX SOAP encoded string. Within this SOAP encoded string are all the individual job attributes that are requested by the client. This string is decoded within the BPP server and each value requested is passed to the application where it may be used for processing the get job attributes request. The results of this action are passed back to the BPP server where the SOAP encoded string is constructed for the response sent back to the client.

For a detailed description of the Get Job indication and response, please refer to the Basic Printing Profile [BPP] specification section 7.1.4 entitled “CreateJob”.

Parameters

type
Signal identity, CSR_BT_BPPS_GET_JOB_ATTRIBS_IND/REQ/RES.

jobId
Integer value that specifies the job identifier of the job attributes that has now been requested. The returned jobId from the BPP server will be the same as the original jobId requested by the BPP client.

*jobAttribs
Data chunk of concatenated zero-terminated strings with values that specify each of the job attributes that are requested by the BPP client.

jobAttribsCount
Number of strings in jobAttribs

jobAttribsLength
Total length of jobAttribs

*jobState
String value that specifies the state of the current print job. The description can be on of the following values: - “printing”, “waiting”, “stopped”, “completed”, “aborted”, “cancelled” or “unknown”.

*jobName
String value that specifies the user-friendly name of the specified job. The BPP server may return an empty string if this attribute was not set in the CreateJob operation by the BPP client.

*jobOriginatingUserName
String value that specified the name of the user that submitted the specified job.

jobMediaSheetsCompleted
Integer value that specifies the number of media sheets completed for the specified job so far.

numberOfInterveningJobs
Integer value that specifies the number of unique print jobs that may be printed ahead of the print job specified by the jobId.

operationStatus	Status of the get job attribs operation. The operationStatus are defined in (csr_bt_obex.h) with the following type CsrBtObexOperationStatusCode and can also be found in the Basic Printing Profile [BPP] specification section 16.2 entitled “Operation Status Codes” for defined code values.
smpOn	Reserved for future use. Set to FALSE.

4.11 CSR_BT_BPPS_CANCEL_JOB

Parameters Primitives	type	jobId	operationStatus
CSR_BT_BPPS_CANCEL_JOB_IND	✓	✓	
CSR_BT_BPPS_CANCEL_JOB_RES	✓	✓	✓

Table 12: CSR_BT_BPPS_CANCEL_JOB Primitives

Description

This indication and response signal is used where the client has requested to cancel a current printer job that is currently in existence. The cancel job message is sent from the client to the BPP server in the form of an OBEX SOAP encoded string. Within this SOAP encoded string is the jobId attribute of the job that is to be cancelled. This string is decoded within the BPP server and the jobId value is passed on to the application where it may be used for processing the cancel job request. The results of this action are passed back to the BPP server where the SOAP encoded string is constructed for the response sent back to the client.

For a detailed description of the cancel job indication and response please refer to the Basic Printing Profile [BPP] specification section 7.1.5 entitled “CancelJob”.

Parameters

type Signal identity, CSR_BT_BPPS_CANCEL_JOB_IND/REQ/RES.

jobId Integer value that specifies the job identifier of the job attributes that has now been requested. The returned jobId from the BPP server will be the same as the original jobId requested by the BPP client.

operationStatus Status of the cancel job operation.
The operationStatus are defined in (csr_bt_obex.h) with the following type CsrBtObexOperationStatusCode and can also be found in the Basic Printing Profile [BPP] specification section 16.2 entitled “Operation Status Codes” for defined code values.

4.12 CSR_BT_BPPS_SEND_DOCUMENT_IND

Parameters / Primitives	type	documentTypeLength	*documentType	jobId	*fileName	*documentBody	documentBodyLength	*docTypeDependentInfo	docTypeDependentInfoLength
CSR_BT_BPPS_SEND_DOCUMENT_IND	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 13: CSR_BT_BPPS_SEND_DOCUMENT_IND Primitives

Description

This signal indicates that the client has transferred a file to the BPP server and that the file can be processed in the appropriate manner in the BPPS application. The file could have been transferred either with the simple document push or job-based document transfer method as specified in the BPP specification. The value of the jobId parameter indicates which method the BPP client used.

For a detailed description of the cancel job request and response, please refer to the Basic Printing Profile [BPP] specification section 7.1.3 entitled “SendDocument”.

Parameters

type — Signal identity, CSR_BT_BPPS_SEND_DOCUMENT_IND.

documentTypeLength — Length in bytes of the documentType parameter.

*documentType — A null terminated UTF8 string with 8 bit ASCII text describing the MIME type of the object, such as vCard, vCal, etc.

jobId — Integer value that specifies the job identifier of the job that the sent document is associated with. If this parameter is set to a value greater than zero then this indicates that the job-based document transfer method is being used. If this value is set to zero then this indicates that the simple document push transfer model

*fileName — Utf8 String value that specifies the name of the file for the document that has just been transferred from the BPP client to the server.

*documentBody — Pointer to a buffer that holds the entire document that has just been transferred from the BPP client to the server.

documentBodyLength — Integer value that holds the length in bytes of the file that holds the document that has just been transferred from the BPP client to the server.

*docTypeDependentInfo — Pointer to a 0-terminated UTF8 string that holds the document type dependent information that has just been transferred from the BPP client to the server.

It is optional to specify any document type dependent information. If no document type dependent information is specified, *docTypeDependentInfo is set to NULL. Note that the application must CsrPfree the data pointed to by *docTypeDependentInfo.

NOTICE: This parameter will be implemented in a future release of CSR Synergy Bluetooth. Which release the parameter will be implemented in, has not been

	planned yet. This parameter will be set to NULL until it is implemented.
docTypeDependentInfoLength	Integer value that holds the length in bytes of the document type dependent information. NOTICE: This parameter will be implemented in a future release of CSR Synergy Bluetooth. Which release the parameter will be implemented in, has not been planned yet. This parameter will be set to 0 until it is implemented.

4.13 CSR_BT_BPPS_ABORT

Parameters / Primitives	type	connectionId
CSR_BT_BPPS_ABORT_IND	✓	✓

Table 14: CSR_BT_BPPS_ABORT Primitives

Description

This signal is used when a client decides to terminate a multi-packet operation before it normally ends.

Please notice that the orderly sequence of request (from a client) followed by a response (from a server) has one exception. The ABORT operation may come in the middle of a request/response sequence.

Parameters

type	Signal identity, CSR_BT_BPPS_ABORT_IND.
connectionId	The connection Id for this session, the BPP client must use this Id in the request.

4.14 CSR_BT_BPPS_DISCONNECT

Parameters Primitives	type	connectionId	reasonCode	reasonSupplier
CSR_BT_BPPS_DISCONNECT_IND	✓	✓	✓	✓

Table 15: CSR_BT_BPPS_DISCONNECT Primitives

Description

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

Parameters

type	Signal identity, CSR_BT_BPPS_DISCONNECT_IND.

connectionId	The connection Id for this session, the BPP client must use this Id in the request.

reasonCode	The reason code of the operation. Possible values depends on the value of reasonSupplier. If e.g. 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

4.15 CSR_BT_BPPS_GET_REFERENCED_OBJ

Parameters Primitives	type	*objName	offset	count	getFileSize	*payload	payloadLength	filesize	bodyOffset	bodyLength	resultCode	resultSupplier	smpOn
CSR_BT_BPPS_GET_REFERENCED_OBJ_REQ	✓	✓	✓	✓	✓								✓
CSR_BT_BPPS_GET_REFERENCED_OBJ_IND	✓					✓	✓	✓	✓	✓			
CSR_BT_BPPS_GET_REFERENCED_OBJ_RES	✓												✓
CSR_BT_BPPS_GET_REFERENCED_OBJ_CFM	✓					✓	✓	✓	✓	✓	✓	✓	

Table 16: CSR_BT_BPPS_GET_REFERENCED_OBJ Primitives

Description

If the application encounters a referenced object it can send a get referenced object request to BPPS. It is possible to request retrieval of either the entire referenced object or only a fragment of the referenced object. If the entire object must be retrieved count should be set to “-1” else count is used for indicating the amount of bytes to retrieve in the fragment. The support for retrieving only a fragment of the referenced object enables the application to retrieve parts of different referenced objects in a multiplexing manner. The requested fragments will be delivered to the application by the CSR_BT_BPPS_GET_REFERENCED_OBJ_IND message and the CSR_BT_BPPS_GET_REFERENCED_OBJ_CFM message when the last part of the fragment arrives. The application must respond with a CSR_BT_BPPS_GET_REFERENCED_OBJ_RES message to the CSR_BT_BPPS_GET_REFERENCED_OBJ_IND message, when it is ready to retrieve the next part of the referenced object fragment.

For a detailed description of the Get referenced object operation, please refer to the Basic Printing Profile [BPP] specification section 7.1.6 entitled “GetReferencedObjects”.

Parameters

type Signal identity, CSR_BT_BPPS_GET_JOB_ATTRIBS_REQ/IND/RES/CFM.

jobId Integer value that specifies the job identifier of the job attributes that has now been requested. The returned jobId from the BPP server will be the same as the original jobId requested by the BPP client.

*objName Pointer to a string containing the name of the referenced object.

offset The byte offset into the referenced object. The first byte of the object is byte zero. It is important that the application keeps track of the offset into the referenced object when the object is retrieved in multiple fragments.

count The number of bytes to retrieve from the referenced object. Set to -1 if the entire object should be retrieved.

getFileSize Boolean value, set to true if the remote client should return the file size of the referenced object if that information is available to the remote client.

*payload Pointer to the OBEX packet, which contains information and data from the remote client.

payloadLength Length, i.e. number of bytes, pointed to by the *payload* pointer.

filesize
Size of the referenced object, set to -1 if the remote client can not provide this information.

bodyOffset
Offset value of the body of the referenced object. This value must be added to the *payload* value, which will result in a pointer to the data of the referenced object.

bodyLength
Length of the body-fragment of the referenced object.

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. If the resultSupplier == CSR_BT_SUPPLIER_OBEX then the possible result codes can be found in csr_bt_obex.h. All values which are currently not specified in the respective prim.h files or csr_bt_obex.h 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

smpOn
Reserved for future use. Set to FALSE.

4.16 CSR_BT_BPPS_GET_EVENT

Primitives \ Parameters	type	jobId	updateEvents	*jobState	*printerState	*printerStateReasons	operationStatus	smpOn
CSR_BT_BPPS_GET_EVENT_IND	✓	✓	✓					
CSR_BT_BPPS_GET_EVENT_RES	✓	✓		✓	✓	✓	✓	✓

Table 17: CSR_BT_BPPS_GET_EVENT Primitives

Description

The remote client can request to be continuously updated on the printer and job states. BPPS informs the application about this by sending a CSR_BT_BPPS_GET_EVENT_IND message with the updateEvents parameter set to TRUE. After the application has received such a message it is obligated to send a CSR_BT_BPPS_GET_EVENT_RES message every time either of the jobState, printerState or operationStatus changes state. All parameters of the response message must be included in every response. The application must keep on sending the response messages to BPPS until the application receives a CSR_BT_BPPS_GET_EVENT_IND message with the updateEvents parameter set to FALSE or until the application sends a CSR_BT_BPPS_JOB_COMPLETE_REQ message to indicate to BPPS that the print job is completely finished.

For a detailed description of the Get event operation, please refer to the Basic Printing Profile [BPP] specification section 7.1.7 entitled “GetEvent (Event Notification)”.

Parameters

type — Signal identity, CSR_BT_BPPS_GET_EVENT_IND/RES.

jobId — Integer value that specifies the job identifier of the job attributes that has now been requested. The returned jobId from the BPP server will be the same as the original jobId requested by the BPP client.

UpdateEvents — This Boolean value indicates whether the application should send state change information to BPPS or not. When the value is TRUE the application must continuously update the BPPS. If the value is FALSE the application should stop sending CSR_BT_BPPS_GET_EVENT_RES messages.

*jobState — Pointer to a string that specifies the state of the current print job. The description can be on of the following values: - “printing”, “waiting”, “stopped”, “completed”, “aborted”, “cancelled” or “unknown”.

*printerState — Pointer to a string that identifies the current state of the printer. The values held by this string can be “idle”, “processing” and “stopped”.

*printerStateReasons — Pointer to a string that indicates additional information about why the printer is in its current state. The following values are defined in the Basic Printing Profile [BPP] specification: “none” , “attention-required” , “media-jam” , “paused” , “door-open” , “media-low” , “media-empty” , “output-area-almost-full” , “out-put-area-full” , “marker supply-low” , “marker-supply-empty”, “marker-failure”

operationStatus — Status of the get event operation.
The operationStatus are defined in (csr_bt_obex.h) with the following type CsrBtObexOperationStatusCode and can also be found in the Basic Printing Profile [BPP] specification section 16.2 entitled “Operation Status Codes” for defined code values.

smpOn Reserved for future use. Set to FALSE.

4.17 CSR_BT_BPPS_JOB_COMPLETE

Parameters / Primitives	type
CSR_BT_BPPS_JOB_ COMPLETE_REQ	✓
CSR_BT_BPPS_JOB_ COMPLETE_CFM	✓

Table 18: CSR_BT_BPPS_JOB_COMPLETE Primitives

Description

The CSR_BT_BPPS_JOB_ COMPLETE_REQ message must be sent from the application to BPPS when the print job has been completed. BPPS then internally prepares for the next OBEX connection. when BPPS has completed this preparation it sends a CSR_BT_BPPS_JOB_ COMPLETE_CFM message to the application in order to signal that it is ready for new connections.

Parameters

type Signal identity, CSR_BT_BPPS_JOB_COMPLETE_REQ/CFM.

4.18 CSR_BT_BPPS_SECURITY_IN

Parameters / Primitives	type	appHandle	secLevel	resultCode	resultSupplier
CSR_BT_BPPS_SECURITY_IN_REQ	✓	✓	✓		
CSR_BT_BPPS_SECURITY_IN_CFM	✓			✓	✓

Table 19: CSR_BT_BPPS_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 the BPPS status and object channels do not use security as required in [BPP], hence it is not possible to change the security levels of these channels. However, the job channel is always the first to be established, and the status and object channel will only accept connections (both incoming and outgoing) to addresses already connected on the job channel.

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_BPPS_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. If the resultSupplier == CSR_BT_SUPPLIER_OBEX then the possible result codes can be found in csr_bt_obex.h. All values which are currently not specified in the respective prim.h files or csr_bt_obex.h 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

5 Document References

Document	Reference
Basic Printing Profile Interoperability Specification 15 November 2002	[BPP]
Generic Object Exchange Profile Version 1.1 Profile section K:10 22 February 2001	[GOEP]
IrDA Object Exchange Protocol – IrOBEX Version 1.2 18 March 1999	[OBEX]
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
BPPS	OBEX Basic Printing Profile (server)
CSR	Cambridge Silicon Radio
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.