



Part K:11

OBJECT PUSH PROFILE

This application profile defines the application requirements for Bluetooth devices necessary for the support of the Object Push usage model. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the Object Push usage model.



CONTENTS

1	Introduction	334
1.1	Scope	334
1.2	Bluetooth Profile Structure	334
1.3	Bluetooth OBEX-Related Specifications	335
1.4	Symbols and conventions	336
1.4.1	Requirement status symbols	336
1.4.2	Signaling diagram conventions	337
2	Profile overview	338
2.1	Profile stack.....	338
2.2	Configurations and roles	338
2.3	User requirements and scenarios	339
2.4	Profile fundamentals	339
3	User interface aspects	340
3.1	Mode selection, Push Servers	340
3.2	Function Selection, Push Clients	340
3.3	Application Usage Events	341
3.3.1	Object Push.....	341
3.3.2	Business Card Pull	341
3.3.3	Business Card Exchange	342
4	Application layer	344
4.1	Feature overview.....	344
4.2	Object Push Feature	344
4.2.1	Content Formats.....	344
4.2.2	Application Procedure	345
4.3	Business Card Pull Feature	345
4.3.1	Owner's Business Card.....	346
4.3.2	Application Procedure Business Card Pull.....	346
4.4	Business Card Exchange Feature	346
4.4.1	Owner's Business Card.....	346
4.4.2	Application Procedure Business Card Exchange.....	346
5	OBEX	348
5.1	OBEX Operations Used	348
5.2	OBEX Headers	348
5.2.1	OBEX Headers for the Object Push Feature.....	348
5.2.2	OBEX Headers for the Business Card Pull and Exchange Features.....	349
5.3	Initialization of OBEX	350
5.4	Establishment of OBEX session	350



- 5.5 Pushing Data 350
- 5.6 Pulling Data 350
- 5.7 Disconnection 350
- 6 Service Discovery 351**
 - 6.1 SD Service Records 351
 - 6.2 SDP Protocol Data Units..... 352
- 7 References..... 353**
 - 7.1 Normative references 353

FOREWORD

This document, together with the Generic Object Exchange profile and the Generic Access profile, forms the Object Push usage model.

Interoperability between devices from different manufacturers is provided for a specific service and usage model if the devices conform to a Bluetooth SIG defined profile specification. A profile defines a selection of messages and procedures (generally termed *capabilities*) from the Bluetooth SIG specifications and gives an unambiguous description of the air interface for specified service(s) and usage model(s).

All defined features are process-mandatory. This means that if a feature is used, it is used in a specified manner. Whether the provision of a feature is mandatory or optional is stated separately for both sides of the Bluetooth air interface.



1 INTRODUCTION

1.1 SCOPE

The Object Push profile defines the requirements for the protocols and procedures that shall be used by the applications providing the Object Push usage model. This profile makes use of the [Generic Object Exchange Profile \(GOEP\) \[10\]](#) to define the interoperability requirements for the protocols needed by applications. The most common devices using these usage models can be notebook PCs, PDAs, and mobile phones.

The scenarios covered by this profile are the following:

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

1.2 BLUETOOTH PROFILE STRUCTURE

In [Figure 1.1 Bluetooth Profiles](#), the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure: a profile has dependencies on the profile(s) in which it is contained – directly and indirectly. For example, the Object Push profile is dependent on Generic Object Exchange, Serial Port, and Generic Access profiles.

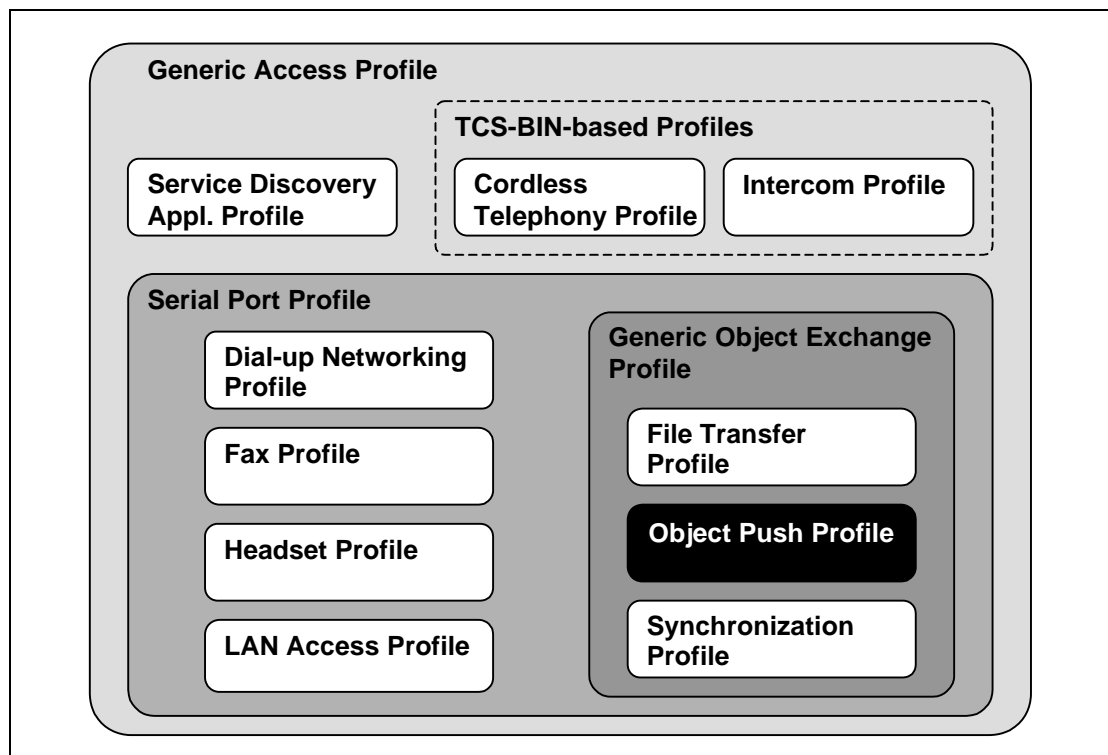


Figure 1.1: Bluetooth Profiles

1.3 BLUETOOTH OBEX-RELATED SPECIFICATIONS

Bluetooth Specification includes five separate specifications for OBEX and applications using OBEX.

1. Bluetooth IrDA Interoperability Specification [7].

- Defines how the applications can function over both Bluetooth and IrDA.
- Specifies how OBEX is mapped over RFCOMM and TCP.
- Defines the application profiles using OBEX over Bluetooth.

2. Bluetooth [Generic Object Exchange Profile](#) Specification [10]

- Generic interoperability specification for the application profiles using OBEX.
- Defines the interoperability requirements of the lower protocol layers (e.g. Baseband and LMP) for the application profiles.

3. Bluetooth [Synchronization Profile](#) Specification [15]

- Application Profile for Synchronization applications.
- Defines the interoperability requirements for the applications within the Synchronization application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

4. Bluetooth [File Transfer Profile](#) Specification [14]

- Application Profile for File Transfer applications.
- Defines the interoperability requirements for the applications within the File Transfer application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

5. **Bluetooth Object Push Profile Specification (this specification)**

- Application Profile for Object Push applications.
- Defines the interoperability requirements for the applications within the Object Push application profile.
- Does not define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

1.4 SYMBOLS AND CONVENTIONS

1.4.1 Requirement status symbols

In this document, the following symbols are used:

‘M’ for mandatory to support (used for capabilities that shall be used in the profile);

‘O’ for optional to support (used for capabilities that can be used in the profile);

‘C’ for conditional support (used for capabilities that shall be used in case a certain other capability is supported);

‘X’ for excluded (used for capabilities that may be supported by the unit but shall never be used in the profile);

‘N/A’ for not applicable (in the given context it is impossible to use this capability).

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices following this profile. Therefore, these features shall never be activated while a unit is operating as a unit within this profile.



1.4.2 Signaling diagram conventions

The following arrows are used in diagrams describing procedures:

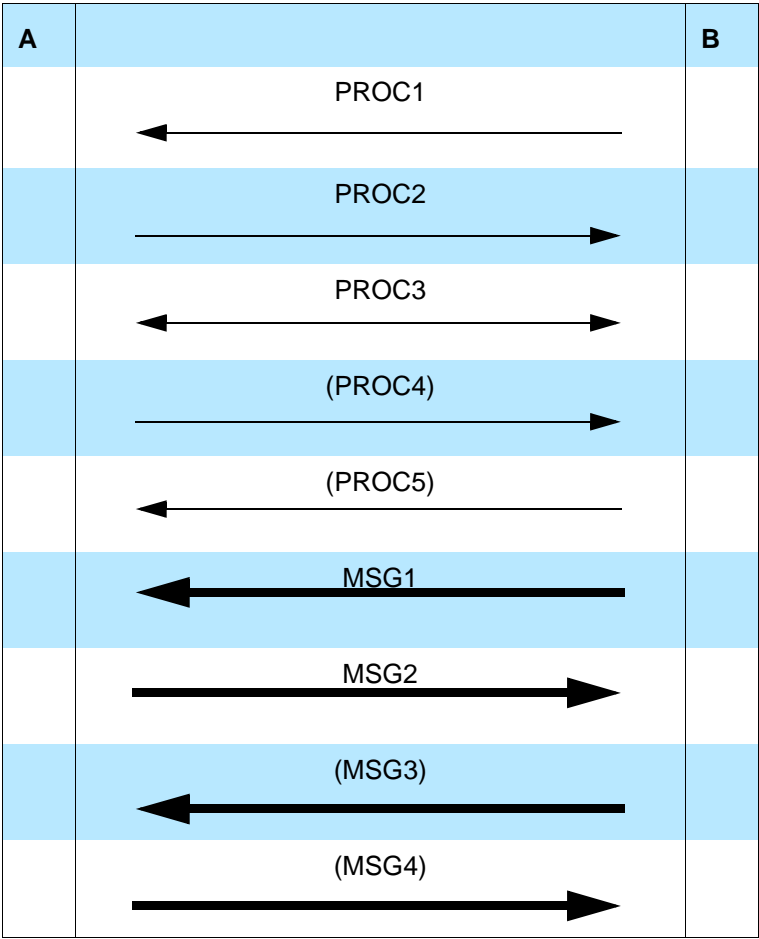


Table 1.1: Arrows used in signaling diagrams

In the table above, the following cases are shown: PROC1 is a sub-procedure initiated by B. PROC2 is a sub-procedure initiated by A. PROC3 is a sub-procedure where the initiating side is undefined (may be both A and B). PROC4 indicates an optional sub-procedure initiated by A, and PROC5 indicates an optional sub-procedure initiated by B.

MSG1 is a message sent from B to A. MSG2 is a message sent from A to B. MSG3 indicates an optional message from A to B, and MSG4 indicates an optional message from B to A.



2 PROFILE OVERVIEW

2.1 PROFILE STACK

The figure below shows the protocols and entities used in this profile.

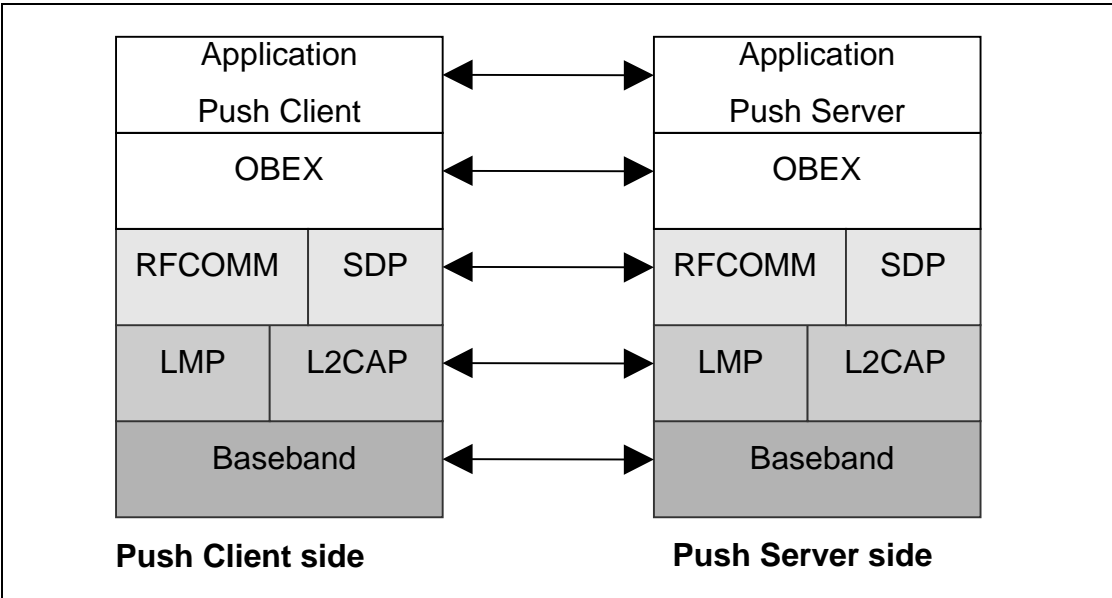


Figure 2.1: Protocol model

The Baseband [1], LMP [2] and L2CAP [3] are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM [4] is the Bluetooth adaptation of GSM TS 07.10 [5]. SDP is the Bluetooth Service Discovery Protocol [6]. OBEX [7] is the Bluetooth adaptation of IrOBEX [8].

The RFCOMM, L2CAP, LMP and Baseband interoperability requirements are defined in Section 6 in GOEP [10].

2.2 CONFIGURATIONS AND ROLES

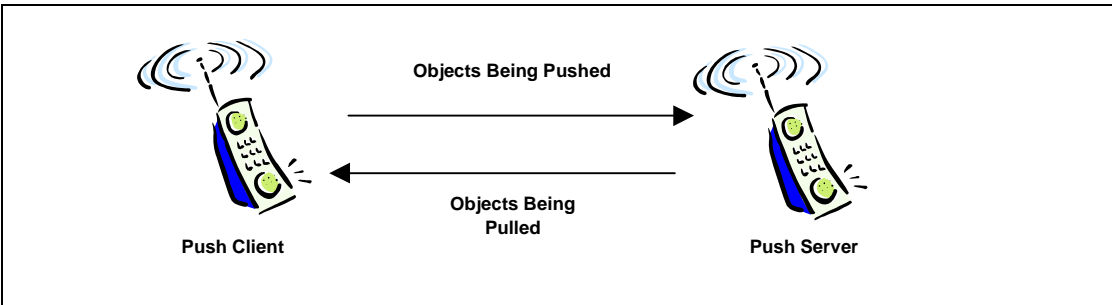


Figure 2.2: Push and Pull Example between two Mobile Phones



The following roles are defined for this profile:

Push Server – This is the server device that provides an object exchange server. In addition to the interoperability requirements defined in this profile, the Push Server must comply with the interoperability requirements for the server of the GOEP if not defined in the contrary.

Push Client – This is the client device that pushes and pulls objects to and from the Push Server. In addition to the interoperability requirements defined in this profile, the Push client must also comply with the interoperability requirements for the client of the GOEP, if not defined to the contrary.

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are:

- Usage of a Push Client to push an object to a Push Server. The object can, for example, be a business card or an appointment.
- Usage of a Push Client to pull a business card from a Push Server.
- Usage of a Push Client to exchange business cards with a Push Server.

The restrictions applying to this profile are the same as in the GOEP.

The push operation described in this profile pushes objects from the Push Client to the inbox of the Push Server.

2.4 PROFILE FUNDAMENTALS

The profile fundamentals are the same as defined in the GOEP.

Link level authentication and encryption are mandatory to support and optional to use.

Bonding is mandatory to support and optional to use.

OBEX authentication is not used.

This profile does not mandate the server or client to enter any discoverable or connectable modes automatically, even if they are able to do so. On the Push Client side, end-user interaction is always needed to initiate the object push, business card pull or business card exchange.

3 USER INTERFACE ASPECTS

3.1 MODE SELECTION, PUSH SERVERS

Object Exchange mode affects the Push Server. It enables Push Clients to push and pull objects to and from the Push Server. The Push Clients can also try to pull objects from the Push Server in this mode. The Push Server does not have to support the pulling feature, but it must be able to respond with an appropriate error message.

When entering this mode, Push Servers should:
set the device in Limited Discoverable Mode (see [Generic Access Profile](#)),
must ensure that the Object Transfer bit is set in the CoD (see [\[16\]](#)),
and must ensure that a service record is registered in the SDDB (see [Section 6](#)).

Public devices, devices that want to be visible at all times, or devices that can not supply a user interface to enable Object Exchange mode shall use General Discoverable Mode (see [\[13\]](#)) instead of Limited Discoverable Mode.

It is recommended that this mode be set and unset by user interaction.

3.2 FUNCTION SELECTION, PUSH CLIENTS

- There are three different **functions** associated with the Object Push profile:
- Object Push function
- Business Card Pull function
- Business Card Exchange function

The **Object Push function** initiates the function that pushes one or more objects to a Push Server.

The **Business Card Pull function** initiates the function that pulls the business card from a Push Server.

The **Business Card Exchange function** initiates the function that exchanges business cards with a Push Server.

The three functions should be activated by the user. They should not be performed automatically without user interaction.

When the user selects one of these functions, an inquiry procedure will be performed to produce a list of available devices in the vicinity. Requirements on inquiry procedures are discussed in [Section 6.5.1](#) of the GOEP [\[10\]](#).



3.3 APPLICATION USAGE EVENTS

In the following sections (3.3.1-3.3.3), the presented scenarios work as examples. Variations in the actual implementations are possible and allowed.

3.3.1 Object Push

When a Push Client wants to push an object to a Push Server, the following scenario can be followed. If authentication is used the user might have to enter a Bluetooth PIN at some point.

Push Client	Push Server
	The user sets the device into Object Exchange mode .
The user of the Push Client selects the Object Push function on the device.	
A list of Push Servers that may support the Object Push service is displayed to the user.	
The user selects a Push Server to push the object to. If the selected device does not support the Object Push service, the user is prompted to select another device.	
	When an object is received in the Push Server, it is recommended that the user of the Push Server be asked to accept or reject the object.
It is recommended that the user is notified of the result of the operation.	

3.3.2 Business Card Pull

When a Push Client wants to pull the business card from a Push Server the following user interaction can be followed.

If authentication is used, the user might have to enter a Bluetooth PIN at some point.

Push Client	Push Server
	The user sets the device into Object Exchange mode .
The user of the Push Client selects the Business Card Pull function on the device.	
A list of Push Servers that may support the Object Push service is displayed to the user.	
The user selects a Push Server to pull the business card from. If the selected device does not support the Object Push service, the user is prompted to select another device.	
	Some devices might ask the user whether or not to accept the request to pull the business card from his device.
It is recommended that the user is notified of the result of the operation.	

3.3.3 Business Card Exchange

When a Push Client wants to exchange business cards with a Push Server, the following user interaction can be followed.

If authentication is used, the user might have to enter a Bluetooth PIN at some point.

Push Client	Push Server
	The user sets the device into Object Exchange mode .
The user of the Push Client selects the Business Card Exchange function on the device.	
A list of Push Servers that may support the Object Push service is displayed to the user.	
The user selects a Push Server to exchange business cards with. If the selected device does not support the Object Push service, the user is prompted to select another device.	



Push Client	Push Server
	When a Push Client tries to exchange business cards with the Push Server, it is recommended that the user of the Push Server is asked to accept or reject the business card offered by the Push Client. Some devices might also ask the user whether or not to accept the request to pull the business card from his device.
It is recommended that the user is notified of the result of the operation.	



4 APPLICATION LAYER

This section describes the feature requirements on units active in the Object Push, Business Card Pull and Business Card Exchange use cases.

4.1 FEATURE OVERVIEW

Table 4.1 shows the features covered by the Object Push profile.

	Features	Support in Push Client	Support in Push Server
1.	Object Push	M*	M*.
2.	Business Card Pull	O	O†
3.	Business Card Exchange	O	O†.

Table 4.1: Application layer features

- *. Support for vCard format is mandatory. Support for other formats is optional, although the server must be able to respond with an error code on a put request even if it doesn't support the requested format.
- †. Optional, but the server must be able to respond with an error code on a pull request, even if it doesn't support this feature

4.2 OBJECT PUSH FEATURE

This feature lets a Push Client send one or more objects to a Push Server.

4.2.1 Content Formats

To achieve application level interoperability, content formats are defined for Object Push. For some applications content formats have been specified.

- Phone Book applications must support data exchange using the vCard 2.1 content format specified in [11]. The properties that are mandatory to support are listed in Chapter 7 of [9]. If a phone book application supports another content format it must still support the vCard 2.1 content format. If a device does not have a phone book application it does not have to support the vCard 2.1 content format.
- Calendar applications must support data exchange using the vCalendar 1.0 content format specified in [12]. The properties that are mandatory to support are listed in Chapter 8 of [9]. If a calendar application supports another content format it must still support the vCalendar 1.0 content format. If a device does not have a calendar application it does not have to support the vCalendar 1.0 content format.



- Messaging applications must support data exchange using the vMessage content format specified in Chapter 9 of [9]. If a messaging application supports another content format it must still support the vMessage content format as specified in Chapter 9 of [9]. If a device does not have a messaging application it does not have to support the vMessage content format.
- Notes applications must support data exchange using the vNote content format specified in Chapter 10 of [9]. If a notes application supports another content format it must still support the vNote content format as specified in Chapter 10 of [9]. If a device does not have a notes application it does not have to support the vNote content format.

It is highly recommended that a Push Client does not try to send objects of a format that the Push Server does not support. See [Section 6](#) for information on how to find out which formats the Push Server supports.

The content formats supported by a Push Server must be identified in the SDDB.

4.2.2 Application Procedure

It is mandatory for Push Servers to be able to receive multiple objects within an OBEX connection. It is not mandatory for Push Clients to be able to send multiple objects during an OBEX connection. The Push Client uses one PUT operation for each object it wants to send. It is not mandatory to support sending or receiving of multiple objects within a single PUT operation.

[Table 4.2](#) shows the application procedure required by the Push Client to push one or more objects to a Push Server.

Push Client	Details
OBEX CONNECT.	Target Header must not be used.
One or more OBEX PUTs for sending one or more objects.	
OBEX DISCONNECT	

Table 4.2: Application layer procedure for Object Push

For a detailed description of OBEX operations see [Section 5](#).

4.3 BUSINESS CARD PULL FEATURE

A Push Client can optionally supply the functionality needed to pull a business card from a Push Server.



It is optional for the Push Server to support the business card pull feature. However, it must be able to respond to pull requests with an error message, see [Section 5.6](#).

4.3.1 Owner’s Business Card

Devices that support the business card pull and business card exchange services must store the owner’s business card in the OBEX Default Get Object. Some devices (e.g. public devices) might hold information in the owner’s business card that is relevant to the device rather than to the owner of the device.

The Default Get Object does not have a name; instead it is identified by its type. To achieve the ultimate application level interoperability, both the Push Client and the Push Server must support the vCard 2.1 content format specified in [\[11\]](#).

See [\[8\]](#) for a discussion on the Default Get Object.

4.3.2 Application Procedure Business Card Pull

[Table 4.3](#) shows the application procedure required by the Push Client to perform a Business Card Pull from a Push Server.

Push Client	Details
OBEX CONNECT.	Target Header must not be used.
OBEX GET vCard of server’s business card (default get object).	Type Header must be set to “text/x-vcard”. Name Header must not be used.
OBEX DISCONNECT.	

Table 4.3: Application layer procedure for Business Card Pull

For a detailed description of OBEX operations see [Section 5](#).

4.4 BUSINESS CARD EXCHANGE FEATURE

A Push Client can optionally supply the functionality needed to exchange business cards with a Push Server.

It is optional for the Push Server to support the business card exchange feature. It must, however, be able to respond to exchange requests with an error message, see [Section 5.6](#).

4.4.1 Owner’s Business Card

See Business Card Pull feature.



4.4.2 Application Procedure Business Card Exchange

Table 4.4 shows the application procedure required by the Push Client to perform a Business Card Exchange with a Push Server.

Push Client	Details
OBEX CONNECT.	Target Header must not be used.
OBEX PUT vCard with client's business card.	
OBEX GET vCard of server's business card (default get object).	Type Header must be set to "text/x-vcard". Name Header must not be used.
OBEX DISCONNECT.	

Table 4.4: Application layer procedure for Business Card Exchange

For a detailed description of OBEX operations see [Section 5](#).

5 OBEX

5.1 OBEX OPERATIONS USED

Table 5.1 shows the OBEX operations, which are required in the Object Push profile.

Operation no.	OBEX Operation	Push Client	Push Server
1	Connect	M	M
2	Disconnect	M	M
3	Put	M	M
4	Get	O	M
5	Abort	M	M

Table 5.1: OBEX Operations

5.2 OBEX HEADERS

5.2.1 OBEX Headers for the Object Push Feature

Table 5.2 shows the specified OBEX headers which are required in the Object Push profile for the Object Push feature.

Header no.	OBEX Headers	Push Client	Push Server
1	Count	X	X
2	Name	M	M
3	Type	O	O
4	Length	M	M
5	Time	O	O
6	Description	O	O
7	Target	X	X
8	HTTP	O	O
9	Body	M	M
10	End of Body	M	M

Table 5.2: OBEX Headers used for the Object Push feature



Header no.	OBEX Headers	Push Client	Push Server
11	Who	X	X
12	Connection ID	X	X
13	Authenticate Challenge	X	X
14	Authenticate Response	X	X
15	Application Parameters	X	X
16	Object Class	X	X

Table 5.2: OBEX Headers used for the Object Push feature

5.2.2 OBEX Headers for the Business Card Pull and Exchange Features

Table 5.3 shows the specified OBEX headers which are required in the Object Push profile for the Business Card Pull and Exchange features.

Header no.	OBEX Headers	Push Client	Push Server
1	Count	X	X
2	Name	M	M
3	Type	M	M
4	Length	M	M
5	Time	O	O
6	Description	O	O
7	Target	X	X
8	HTTP	O	O
9	Body	M	M
10	End of Body	M	M
11	Who	X	X
12	Connection ID	X	X
13	Authenticate Challenge	X	X
14	Authenticate Response	X	X
15	Application Parameters	X	X
16	Object Class	X	X

Table 5.3: OBEX Headers used for the business card pull and business card exchange features

5.3 INITIALIZATION OF OBEX

Since OBEX authentication is not used by this profile, OBEX initialization is not applicable.

5.4 ESTABLISHMENT OF OBEX SESSION

See [Section 5.4.1](#), in GOEP for a description of OBEX connection establishment without authentication.

The Push Client does not use the target header when establishing an OBEX connection.

5.5 PUSHING DATA

It is highly recommended that the Push Client use the Type Header when pushing objects to the Push Server.

See [Section 5.5](#) in GOEP.

5.6 PULLING DATA

In the Object Push Profile, the Push Client only pulls data from the Push Server when it is getting the Default Get Object (owner's business card).

If there is no Default Get Object, the Push Server must respond with the error response code "NOT FOUND" [\[8\]](#). The Push Client must be able to understand this error response code.

The Push Client must use the Type Header when getting the Default Get Object from the Push Server.

The Name Header is not used when getting the Default Get Object from the Push Server. If the Push Client sends a non-empty Name header, the Push Server should respond with the response code "FORBIDDEN" [\[8\]](#).

See [Section 5.6](#) in GOEP.

5.7 DISCONNECTION

See [Section 5.7](#) in GOEP.



6 SERVICE DISCOVERY

6.1 SD SERVICE RECORDS

The SD service record for the Object Push service is defined in [Table 6.1](#).
A Push Client does not provide any SD service records.

Item	Definition	Type Size	Value *	AttrID	Status	Default Value
Service Class ID List				See [16]	M	
Service Class #0		UUID	OBEXObjectPush		M	
Protocol Descriptor List				See [16]	M	
Protocol ID #0		UUID	L2CAP		M	
Protocol ID #1		UUID	RFCOMM		M	
Param #0	Channel	Uint8	Varies		M	
Protocol ID #2		UUID	OBEX		M	
Service Name	Display-able Text name	String	Varies	See [16]	O	"OBEX Object Push"
BluetoothProfileDescriptorList				See [16]	O	
Profile ID #0	Supported profile	UUID	OBEXObjectPush			OBEX-Object-Push [16]
Version #0	Profile version	uint16	Varies			0x0100
Supported Formats List	Supported Formats List	Data Element Sequence of Uint8	Formats: 0x01 = vCard 2.1 0x02 = vCard 3.0 0x03 = vCal 1.0 0x04 = iCal 2.0 0x05 = vNote (as defined in [9]) 0x06 = vMessage (as defined in [9]) 0xFF = any type of object.	See [16]	M	

Table 6.1: Object Push Service Record

*. Values that are of the type UUID are defined in the Assigned Numbers specification [16].



6.2 SDP PROTOCOL DATA UNITS

Table 6.2 shows the specified SDP PDUs (Protocol Data Units), which are required in the Object Push profile.

PDU no.	SDP PDU	Push Client	Push Server
1	SdpErrorResponse	M	M
2	SdpServiceSearchAttributeRequest	M	M
3	SdpServiceSearchAttributeResponse	M	M

Table 6.2: SDP PDUs

7 REFERENCES

7.1 NORMATIVE REFERENCES

- [1] Bluetooth Special Interest Group, Baseband Specification, version 1.1 or later
- [2] Bluetooth Special Interest Group, LMP Specification, version 1.1 or later
- [3] Bluetooth Special Interest Group, L2CAP Specification, version 1.1 or later
- [4] Bluetooth Special Interest Group, RFCOMM with TS 07.10, version 1.1 or later
- [5] ETSI, TS 07.10, Version 6.3
- [6] Bluetooth Special Interest Group, SDP Specification, version 1.1 or later
- [7] Bluetooth Special Interest Group, IrDA Interoperability, version 1.1 or later
- [8] Infrared Data Association, IrDA Object Exchange Protocol (IrOBEX), Version 1.2 with Published Errata, April 1999
- [9] Infrared Data Association, IrMC (Ir Mobile Communications) Specification with Published Errata, Version 1.1, February 1999
- [10] Bluetooth Special Interest Group, Generic Object Exchange Profile
- [11] The Internet Mail Consortium, vCard – The Electronic Business Card Exchange Format, Version 2.1, September 1996
- [12] The Internet Mail Consortium, vCalendar – The Electronic Calendaring and Scheduling Exchange Format, Version 1.0, September 1996
- [13] Bluetooth Special Interest Group, Generic Access Profile Specification, version 1.1 or later
- [14] Bluetooth Special Interest Group, File Transfer Profile Specification, version 1.1 or later
- [15] Bluetooth Special Interest Group, Synchronization Profile Specification, version 1.1 or later
- [16] Bluetooth Special Interest Group, Assigned Numbers specification
<http://www.bluetooth.org/assigned-numbers.htm>