BLUETOOTH® DOC	Date / Year-Month-Day 2010-08-26	Approved Adopted	Revision V12r00	Document No OPP_SPEC
Prepared By OBEX WG	E-mail Address obex-feedback@bluetooth.org			N.B.

OBJECT PUSH PROFILE

Abstract:

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.

Revision History

Revision	Date	Comments
Version 1.1	22 February 2001	Version released with Bluetooth Core Specification V1.1.
D12r00	15 August 2005	Changed document numbering to conform with Bluetooth Document Naming Procedure V10r00.
D12r01	14 September 2005	Editorial updates
D12r02	31 October 2005	Editorial Updates
D12r03	15 November 2005	Editorial Updates
D12r04	15 December 2006	Editorial Updates
D12r05	14 July 2006	Edits from LLO
D12r06	01 March 2007	Edits from LLO
D12r07	15 March 2007	Edits from LLO, IEEE updates (shall/may)
D12r08	31 July 2007	Editorial updates (minor)
D12r09	9 July 2009	Initial version implementing functionality in OBEX Application Enhancements DIPD
D112r09	14 July 2009	Moved L2CAP AMP text into informative appendix; other changes from WG feedback
D12r10	17 July 2009	More feedback from WG
D12r11	20 July 2009	Rectify Object Transfer CoD requirement; expand SDDB acronym, other edits from WG members
D12r12	13 August 2009	Tweaked encryption requirements, PICS reqs
D12r13	20 August 2009	Erratum 3085, 2447, 940 included
D12r14	19-10-2009	Including feedback from BARB review
D12r15	30-10-2009	Further review from BARB
D12r16	30-10-2009	Correct “session” to be “connection”
D12r17dh	18-05-2010	Added errata #385, #498
D12r18	20-05-2010	Corrected references and punctuation
D12r19	01-06-2010	Further editorial, layout tweaks after OBEX WG review. Include ESR errata.
D12r20	02-6-2010	Cleanup for BARB submission
D12r21	04-6-2010	Included Draft spec BARB review comments
D12r22	04-6-2010	Update incorrect erratum number in revision history
D12r23	07-6-2010	Include review comments from Burch. Correct the Appendix wording from the BIP PS, fix up cross-references
V12r00	08-26-2010	Adopted by the Bluetooth SIG Board of Directors

Contributors

Name	**Company**
David Kammer	3Com
Patrik Olsson (editor)	Ericsson Mobile Communications AB
David Suvak	Counterpoint Systems Foundry
Apratim Purakayastha	IBM Corporation
Aron Walker	IBM Corporation
Jon Inouye	Intel Corporation
Tim Howes	Nokia
Stephane Bouet	Nokia Mobile Phones
Riku Mettälä	Nokia Mobile Phone
James Scales	Nokia Mobile Phones
Steve Rybicki	Puma Technology
Shaun Astarabadi	Toshiba Corporation
Katsuhiro Kinoshita	Toshiba Corporation

Disclaimer and Copyright Notice

The copyright in this specification is owned by the Promoter Members of *Bluetooth*® Special Interest Group (SIG), Inc. ("*Bluetooth* SIG"). Use of these specifications and any related intellectual property (collectively, the "Specification"), is governed by the Promoters Membership Agreement among the Promoter Members and *Bluetooth* SIG (the "Promoters Agreement"), certain membership agreements between *Bluetooth* SIG and its Adopter and Associate Members (the "Membership Agreements") and the *Bluetooth* Specification Early Adopters Agreements (1.2 Early Adopters Agreements) among Early Adopter members of the unincorporated *Bluetooth* SIG and the Promoter Members (the "Early Adopters Agreement"). Certain rights and obligations of the Promoter Members under the Early Adopters Agreements have been assigned to *Bluetooth* SIG by the Promoter Members.

Use of the Specification by anyone who is not a member of *Bluetooth* SIG or a party to an Early Adopters Agreement (each such person or party, a "Member"), is prohibited. The legal rights and obligations of each Member are governed by their applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of the applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in termination of the applicable Membership Agreement or Early Adopters Agreement and other liability permitted by the applicable agreement or by applicable law to *Bluetooth* SIG or any of its members for patent, copyright and/or trademark infringement.

THE SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

Each Member hereby acknowledges that products equipped with the *Bluetooth* technology ("*Bluetooth* products") may be subject to various regulatory controls under the laws and regulations of various governments worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of *Bluetooth* products. Examples of such laws and regulatory controls include, but are not limited to, airline regulatory controls, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their *Bluetooth* Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their *Bluetooth* products related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses**. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.**

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST *BLUETOOTH* SIG AND ITS PROMOTER MEMBERS RELATED TO USE OF THE SPECIFICATION.

Bluetooth SIG reserve the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate.

Document Terminology

The Bluetooth SIG has adopted Section 13.1 of the IEEE Standards Style Manual, which dictates use of the words ``shall'', ``should'', ``may'', and ``can'' in the development of documentation, as follows:

- The word shall is used to indicate mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (shall equals is required to).
- The use of the word must is deprecated and shall not be used when stating mandatory requirements; must is used only to describe unavoidable situations.
- The use of the word will is deprecated and shall not be used when stating mandatory requirements; will is only used in statements of fact.
- The word should is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited (should equals is recommended that).
- The word may is used to indicate a course of action permissible within the limits of the standard (may equals is permitted).
- The word can is used for statements of possibility and capability, whether material, physical, or causal (can equals is able to)

Contents

1 Introduction 8
1.1 Scope 8
1.2 Bluetooth Profile Structure 8
1.3 OBEX-Related Specifications Used by this Profile 9
1.4 Symbols and conventions 9
1.4.1 Requirement Status Symbols 9
1.4.2 Signaling Diagram Conventions 10
2 Profile Overview 11
2.1 Profile Stack 11
2.2 Configurations and Roles 11
2.2 User Requirements and Scenarios 12
2.3 Profile Fundamentals 12
3 User Interface Aspects 13
3.1 Mode Selection, Push Servers 13
3.2 Function Selection, Push Clients 13
3.3 Application Usage Events 13
3.3.1 Object Push 14
3.3.2 Business Card Pull 15
3.3.3 Business Card Exchange 16
4 Application Layer 17
4.1 Feature Overview 17
4.2 Object Push Feature 17
4.2.1 Content Formats 17
4.2.2 Application Procedure 18
4.3 Business Card Pull Feature 18
4.3.1 Owner's Business Card 18
4.3.2 Application Procedure Business Card Pull 19
4.4 Business Card Exchange Feature 19
4.4.1 Owner's Business Card 19
4.4.2 Application Procedure Business Card Exchange 19
5 OBEX 21
5.1 OBEX Operations Used 21
5.2 OBEX Headers 22
5.2.1 OBEX Headers for the Object Push Feature 22
5.2.2 OBEX Headers for the Business Card Pull and Exchange Features 22
5.3 Initialization of OBEX 23
5.4 Establishment of OBEX connection 23
5.5 Pushing Data 23
5.6 Pulling Data 23
5.7 Disconnection 24
5.8 Single response mode 24
5.9 Reliable Sessions 24
6 Service Discovery 25
6.1 SDP Service Records 25
7 GOEP Interoperability Requirements 26
8 Normative References 27
9 Appendix A: OPP over AMP (Informative) 28
9.1 OPP using OBEX over L2CAP 28
9.2 OPP using OBEX over RFCOMM 28

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 (OPP) 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. Common devices using these models are notebook PCs, PDAs, and mobile phones.

The scenarios covered by this profile are the following:

- Use of a Bluetooth device to push an object to the inbox of another Bluetooth device. The object can for example be a business card or an appointment.
- Use of a Bluetooth device to pull a business card from another Bluetooth device.
- Use of a Bluetooth device to exchange business cards with another Bluetooth device. Exchange is 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, 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 (for compatibility with devices implementing earlier versions of this profile), and Generic Access profile.

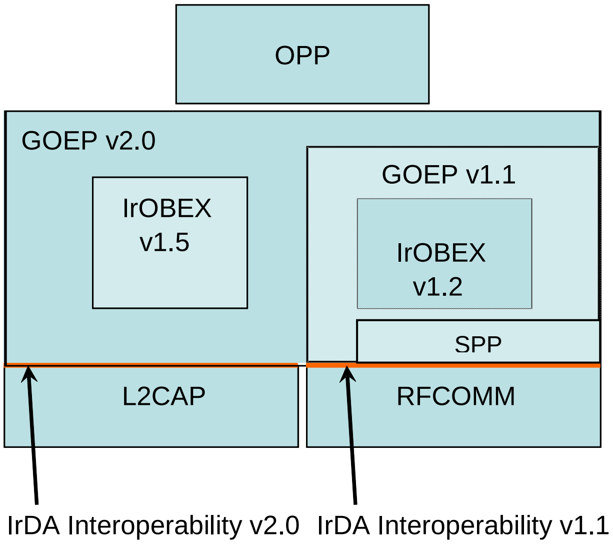

Figure 1.1: Bluetooth Profiles and Protocols

1.3 OBEX-Related Specifications Used by this Profile

This profile refers to the following two specifications:

1. IrDA Interoperability Specification v2.0 [11].
   - Defines how the applications can function over both Bluetooth and IrDA.
   - Specifies how OBEX is mapped over L2CAP and TCP.
2. Generic Object Exchange Profile Specification v2.0 [10]
   - Generic interoperability specification for the application profiles using OBEX over L2CAP.
   - Defines the interoperability requirements of the lower protocol layers (e.g. Baseband and LMP) for the application profiles.

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:

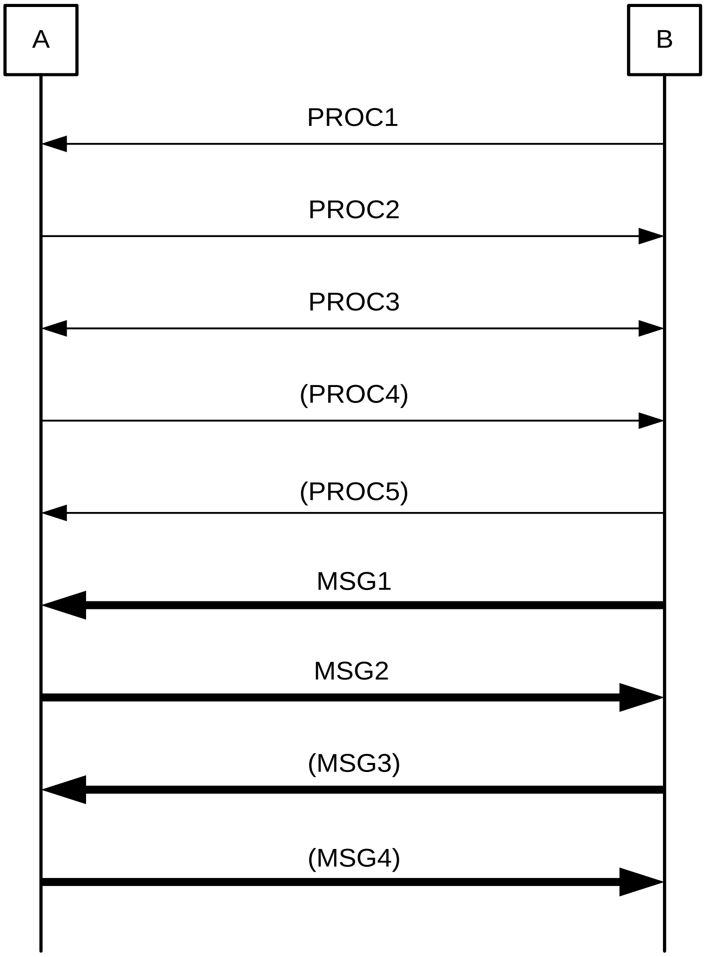

Figure 1.2: Arrows Used in Signaling Diagrams

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

Figure 2.1below shows the protocols and entities used in this profile.

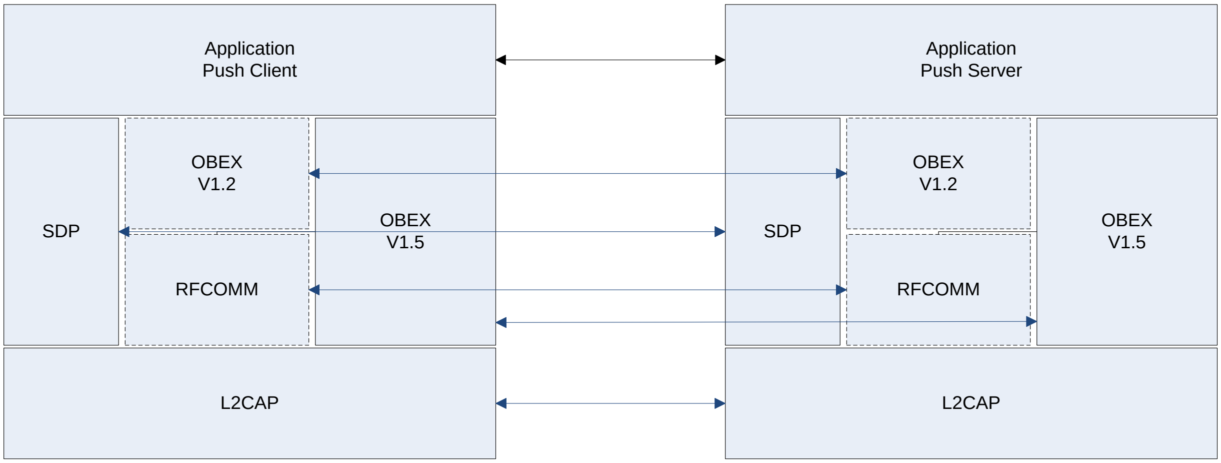

Figure 2.1: Protocol model

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

The L2CAP interoperability requirements are defined in GOEP v2.0 [10]. This profile requires backwards compatibility with previous versions of OPP based on GOEP v1.1. The procedures for backward compatibility defined in section 6.2 of GOEP v2.0 shall be used.

2.2 Configurations and Roles

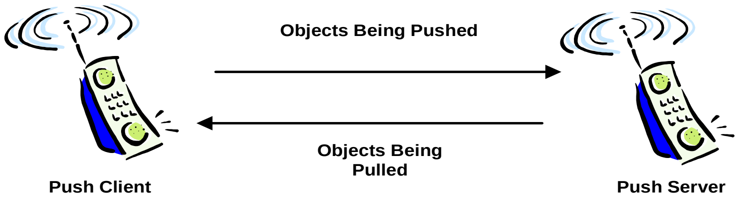

Figure 2.2: Push and Pull Example between Two Mobile Phones

The following roles are defined for this profile:

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

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

2.2 User Requirements and Scenarios

The scenarios covered by this profile are:

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

2.3 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. When using Core Specification v2.1 or later, encryption is mandatory to use; otherwise its use is optional.

Bonding is mandatory to support and optional to use.

OBEX authentication shall not be 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 an 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 may try to pull objects from the Push Server in this mode. If the Push Server does not support the pulling feature, it shall respond with an appropriate error message (see [4]).

When entering this mode, Push Servers

- shall set the Object Transfer bit in the CoD (see [9])
- should set the device into Limited Discoverable Mode (see Generic Access Profile [8])
- shall register a service record in the Service Discovery database (see Section 6).

Devices that want to be visible at all times, or devices that cannot supply a user interface to enable Object Exchange mode should use General Discoverable Mode (see [8]) instead of Limited Discoverable Mode. Devices in General Discoverable Mode may still enter Limited Discoverable Mode for limited time periods. Entry into the Limited Discoverable Mode may be triggered by a button-press or user interface menu option.

It is recommended that Limited Discoverable 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 a 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.

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

3.3 Application Usage Events

In the following sections 3.3.1through 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 GAP authentication is used the user may have be required to enter a Bluetooth passkey or compare two 6-digit numbers at some point during the service level connection establishment.

Push Client	Push Server
	The user sets the device into Object Exchange mode.
The user of the Push Client selects the Object Push function.	
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 either the Object Push service or the type of object being sent, the user is prompted to select another device. Service discovery is used to determine both the services available and the object types supported for an Object Push service on a 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 GAP authentication is used, the user may be required to enter a Bluetooth passkey or compare two 6-digit numbers at some point during the service level connection establishment.

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.	
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 either the Object Push service or the type of object being sent, the user is prompted to select another device. Service discovery is used to determine both the services available and the object types supported for an Object Push service on a device.	
	Some devices might ask the user to accept or reject the request to pull the business card.
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 GAP authentication is used, the user may have be required to enter a Bluetooth passkey or compare two 6-digit numbers at some point during the service level connection establishment.

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.	
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 either the Object Push service or the type of object being sent, the user is prompted to select another device. Service discovery is used to determine both the services available and the object types supported for an Object Push service on a device.	
	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 to accept or reject the request to pull the business card.
It is recommended that the user is notified of the result of the operation.	

4 Application Layer

This section describes the feature requirements for devices that implement 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 v2.1 format is mandatory. Support for other formats is optional. The server shall respond with an error code even if it does not support the requested format.

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. Note that other OBEX application profiles have richer services for transfer and management for some of these content formats. Dependent on the usage models implementers are encouraged to use those profiles in addition to this profile. For some applications content formats have been specified.

Phone Book applications shall support data exchange using the vCard 2.1 content format specified in [6]. The properties that are mandatory to support are listed in Chapter 7 of [5]. If a phone book application supports another content format it shall 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. Whatever the vCard format requested, the character set used to encode vCard attribute content should be UTF-8. The CHARSET property parameter may be used in a vCard to override the default character set; however this is not recommended as it may lead to reduced interoperability of this profile.

Calendar applications shall support data exchange using the vCalendar 1.0 content format specified in [7]. The properties that are mandatory to support are listed in Chapter 8 of [4]. If a calendar application supports another content format it shall 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 shall support data exchange using the vMessage content format specified in Chapter 9 of [5]. If a messaging application supports another content format it shall still support the vMessage content format as specified in Chapter 9 of [5]. If a device does not have a messaging application it does not have to support the vMessage content format.

Notes applications shall support data exchange using the vNote content format specified in Chapter 10 of [5]. If a notes application supports another content format it shall still support the vNote content format as specified in Chapter 10 of [5]. If a device does not have a notes application it does not have to support the vNote content format.

Push Clients should not 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 object formats supported by a Push Server shall be identified in the Service Discovery database – see section 6.

4.2.2 Application Procedure

Push Servers shall be able to receive multiple objects within an OBEX connection. Push Clients may send multiple objects during an OBEX connection. The Push Client uses one PUT operation for each object it wants to send.

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 shall 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

Although Push Client implementations may support the functionality to pull a business card from a Push Server, Push Servers are not required to support the business card pull feature. Push Servers that do not support business card pull shall 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 shall 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 provide application level interoperability, both the Push Client and the Push Server shall support the vCard 2.1 content format specified in [6].

See [4] 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 shall not be used.
OBEX GET vCard of server's business card (default get object).	Type Header shall be set to "text/x-vcard" (case insensitive) . Name Header may be empty or left out entirely, but shall not be issued with an actual value
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 shall, however, be able to respond to exchange requests with an error message if it does not support business card exchange, 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 shall not be used.
OBEX PUT vCard with client's business card.	
OBEX GET vCard of server's business card (default get object).	Type Header shall be set to "text/x-vcard" (case insensitive). Name Header may be empty or left out entirely, but shall not be issued with an actual value.

Push Client	Details
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.

OBEX Operation	Push Client	Push Server	Prohibited when using RFCOMM
Connect	M	M	
Disconnect	M	M	
Put	M	M	
Get	O	M	
Abort	M	M	
Action	X	X	Y
Session	O	O*	Y
SetPath	X	X	Y

Table 5.1: OBEX Operations

*Although Optional, the Server must (according to GOEP) respond with an error if the feature is not supported.

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.

OBEX Headers	Push Client	Push Server	Prohibited when using RFCOMM
Count	O	O	
Name	M	M	
Type	O	O	
Length	M	M	
Time	O	O	
Description	O	O	
HTTP	O	O	
Body	M	M	
End of Body	M	M	
Single Response Mode	O	O	Y
Single Response Mode Parameters	C1	C1	Y
Session Parameters	C2	C2	Y
Session Sequence Number	C2	C2	Y

Table 5.2: OBEX Headers used for the Object Push Feature

C1. Mandatory if Single Response Mode supported, otherwise excluded
C2. Mandatory if Session operation supported, otherwise excluded
All other OBEX headers not shown in the above table are excluded from this 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.

OBEX Headers	Push Client	Push Server	Prohibited when using RFCOMM
Count	O	O	
Name	M	M	
Type	M	M	
Length	M	M	
Time	O	O	
Description	O	O	
HTTP	O	O	
Body	M	M	

OBEX Headers	Push Client	Push Server	Prohibited when using RFCOMM
End of Body	M	M	
Single Response Mode	O	O	Y
Single Response Mode Parameters	C1	C1	Y
Session Parameters	C2	C2	Y
Session Sequence Number	C2	C2	Y

Table 5.3: OBEX Headers Used for the Business Card Pull and Business Card Exchange Features

C1. Mandatory if Single Response Mode supported, otherwise excluded
C2. Mandatory if Session operation supported, otherwise excluded
All other OBEX headers not shown in the above table are excluded from this feature.

5.3 Initialization of OBEX

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

5.4 Establishment of OBEX connection

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

The Push Client should use the Type Header when pushing objects to the Push Server. Push Servers should accept supported objects if sent without a Type header.

The server may send Single Response Mode Parameters (SRMP) headers according to the “user accept” scenario defined in GOEP.

See Section 5.5 in GOEP.

If the Push Server does not support the object format sent in the Push operation, the Push Server should respond with the error code "Unsupported Media Type". Similarly, if the Push Server supports the object format but cannot handle the size of the object being sent in the Push operation, the Push Server should respond with the error code "Requested entity is too large.

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 *shall* respond with the error response code “NOT FOUND” [4]. The Push Client *shall* be able to understand this error response code.

The Push Client shall 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” [4].

The server may send SRMP headers according to the “user accept” scenario defined in GOEP.

See Section 5.6 in GOEP.

5.7 Disconnection

See Section 5.12 in GOEP.

5.8 Single response mode

When transferring “large” objects and OBEX over L2CAP Single response mode (SRM) should be used. The definition of “large” is implementation-dependent. The Server should use the OBEX Length header to deduce the size of the object being transferred.

5.9 Reliable Sessions

When transferring “large” objects Reliable sessions should be used. The definition of “large” is implementation-dependent. The Server should use the OBEX Length header to deduce the size of the object being transferred.

6 Service Discovery

6.1 SDP Service Records

The SDP service record for the Object Push service is defined in Table 6.1. A Push Client does not provide any SDP service records.

Item	Definition	Type Size	Value*	AttrID	Status	Default Value
Service Class ID List				See [9]	M	
Service Class #0		UUID	OBEXObjectPush		M	
Protocol Descriptor List				See [9]	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	Displayable Text name	String	Varies	See [9]	O	“OBEX Object Push”
BluetoothProfile DescriptorList				See [9]	M	
Profile ID #0	Supported profile	UUID	OBEXObjectPush			OBEXObjectPush [16]
Version #0	Profile version	Uint16	Varies			0x0102
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 [9]	M	
GoepL2CapPsm	L2CAP PSM	Uint16	Varies	See [9]	M	

Table 6.1: Object Push Service Record

*Values that are of the type UUID are defined in the Assigned Numbers specification [9].

7 GOEP Interoperability Requirements

This section defines the requirements to interoperate with different versions of GOEP.

Client devices implementing this profile shall implement GOEP v2.0 or later, and follow the GOEP SDP Interoperability Requirements to determine the version of GOEP on the Server device.

The following table shows which IrOBEX version will be used between devices implementing different versions of this profile:

Client	**v1.2 or later**	**Earlier than v1.2**
Server		
v1.2 or later	IrOBEX v1.5	IrOBEX v1.2
Earlier than v1.2	IrOBEX v1.2	IrOBEX v1.2

Table 7.1: IrOBEX versions used in this profile

When GOEP v2.0 or later is used, RFCOMM shall not be used to convey OBEX. When GOEP v1.1 is used, L2CAP shall not directly be used to convey OBEX; and features from IrOBEX later than v1.2 shall not be used

8 Normative References

[1] Core Specification v3.0 or later

[2] ETSI, TS 07.10, Version 6.3

[3] SDP Specification

[4] Infrared Data Association, IrDA Object Exchange Protocol (IrOBEX), Version 1.5 or later

[5] Infrared Data Association, IrMC (Ir Mobile Communications) Specification with Published Errata, Version 1.1, February 1999

[6] The Internet Mail Consortium, vCard – The Electronic Business Card Exchange Format, Version 2.1, September 1996

[7] The Internet Mail Consortium, vCalendar – The Electronic Calendaring and Scheduling Exchange Format, Version 1.0, September 1996

[8] Generic Access Profile Specification

[9] Bluetooth SIG Assigned Numbers – Service Discovery

[10] Generic Object Exchange Profile v2.0

[11] IrDA Interoperability v2.0

9 Appendix A: OPP over AMP (Informative)

This appendix provides guidance to implementers when using OPP in devices conforming to Bluetooth v3.0 + HS or later.

9.1 OPP using OBEX over L2CAP

When FTP is using OBEX over L2CAP devices should use a high speed AMP when transferring “large” files or many “small” objects. SRM should be used to fully utilize HS capabilities of the devices.

The implementer of an FTP device should consider the use of a high speed AMP when transferring files; the Initiator can initiate a connection over AMP and both the Initiator and the Responder can initiate an L2CAP Move to AMP procedure based on the size of the file.

9.2 OPP using OBEX over RFCOMM

When FTP is using OBEX over RFCOMM, the RFCOMM entity may also be transporting traffic from other profiles. This means the L2CAP Move procedure should be used only if all profiles running over RFCOMM are capable of running over AMP and accept a move onto another logical link. It is recommended that the L2CAP Move procedure not be performed when using OBEX over RFCOMM. Devices should use OBEX over L2CAP as specified in GOEP v2.0 to obtain the benefit of the high speed AMP when using this profile.