BLUETOOTH® DOC	Date / Year-Month-Day 2006-04-27	Approved Adopted	Revision V10r00	Document No PBAP_SPEC
Prepared Car Working Group	e-mail address Car-feedback@bluetooth.org			N.B.

PHONE BOOK ACCESS PROFILE

Abstract

The Phone Book Access Profile (PBAP) specification defines the procedures and protocols to exchange Phone Book objects between devices. It is especially tailored for the automotive Hands-Free use case where an onboard terminal device (typically a Car-Kit installed in the car) retrieves Phone Book objects from a Mobile device (typically a mobile phone or an embedded phone). This profile can however also be used for other use cases that require the exchange of Phone Book objects between two devices.

Revision History

Revision	Date	Comments
D05r00	04-05-20	First preliminary draft
D05r01	04-05-22	Draft 0.5 prepared for BARB review
D05r02	04-06-22	BSTS Editorial Review
D05r04	04-08-04	Editorial improvements
D05r05	04-08-18	0.5 level BARB comments reflected
D05r06	04-08-20	Addition of SIM cards handling. Clarification of the Phone Book Download feature.
D05r07	04-08-25	Editorial comments and corrections included. S. Raxter
D07r01	04-09-21	Result of the NissanUK Face to Face meeting
D07r03	04-10-26	Editorial comments and corrections included. S. Raxter
D09r01	04-11-29	0.7 level BARB comments reflected
D09r02	04-11-29	Denso comments taken into account
D09r03	04-11-30	Other BARB comments processed
D09r04	04-12-24	Cosmetic improvements.
D09r05	05-02-22	Changes from BARB review.
D09r06	05-02-28	Comments included. GP.
D09r07	05-03-07	Search mechanism + comments adressed
D09r08	05-03-08	Fix format and spelling errors. SR.
D09r09	05-03-16	More format changes and review comments- SR
D09r10	05-04-10	Change mandatory Char Set requirement- SR
D09r11	05-04-21	Changed Section 3.4 for IEEE language.SR
D09r12	05-05-13	More feedback from the WG included. SB
D09r13	05-06-25	Feedback from IOP in UPF21. SB
D09r14	05-06-27	More Feedback included
D09r15	06-01-12	Errata included. Additional comments from CWG review.
D10r03	06-03-15	Editorial updates
D10r04	06-04-12	Inclusion of errata from tech review.
D10r05	06-14-13	Inclusion of additional errata from tech review
V10r00	06-04-27	Adopted by the Bluetooth Board of Directors

Contributors

Name	Company
Souichi SAITO	Denso
Don LIECHTY	Extended Systems
Stephen RAXTER	Johnson Controls
Michael CARTER	Motorola
Leonard HINDS	Motorola
Tony MANSOUR	Motorola
Stephane BOUET(owner)	Nissan
Patrick CLAUBERG	Nokia
Jamie MCHARDY	Nokia
Jurgen SCHNITZLER	Nokia
Brian TRACY	Nokia
Nicolas BESNARD	Parrot
Guillaume POUJADE	Parrot
Terry BOURK	RF Micro Devices
Dmitri TOROPOV	Siemens
Erwin WEINANS	Sony Ericsson
Tim REILLY	Stonestreet One
Kentaro NAGAHAMA	Toshiba
Robert MALING	Toyota
Akira MIYAJIMA	Toyota
Ryan BRUNER	Visteon

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.

Contents

Figures

Tables

Foreword

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 Phone Book Access Profile (PBAP) defines the protocols and procedures that shall be used by devices for the retrieval of phone book objects. It is based on a Client-Server interaction model where the Client device pulls phone book objects from the Server device.

This profile is especially tailored for the Hands-Free usage case (i.e. implemented in combination with the “Hands-Free Profile” or the “SIM Access Profile”). It provides numerous capabilities that allow for advanced handling of phone book objects, as needed in the car environment. In particular, it is much richer than the Object Push Profile (that could be used to push vCard formatted phone book entry from one device to another).

This profile can also be applied to other usage cases where a Client device is to pull phone book objects from a Server device.

Note however that this profile only allows for the consultation of phone book object (read-only). It is not possible to alter the content of the original phone book object (read/write).

1.2 Profile Dependencies

A profile is dependent upon another profile if it re-uses parts of that profile, by explicitly referencing it. A profile has dependencies on the profile(s) in which it is contained – directly and indirectly.

Phone Book Access Profile is dependent upon the Generic Object Exchange Profile, the Serial Port Profile and the Generic Access Profile.

1.3 Symbols and conventions

1.3.1 Requirement status symbols

In this document, the following symbols are used:

"M" for mandatory to support

"O" for optional to support

"X" for excluded (used for capabilities that may be supported by the unit but shall never be used in this use case)

"C" for conditional to support

"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 in this use case. Therefore, these features shall never be activated while a unit is operating as a unit within this use case.

1.3.2 Signaling diagram conventions

The signaling diagrams in this specification are informative only. Within the diagrams, the following conventions are used to describe procedures:

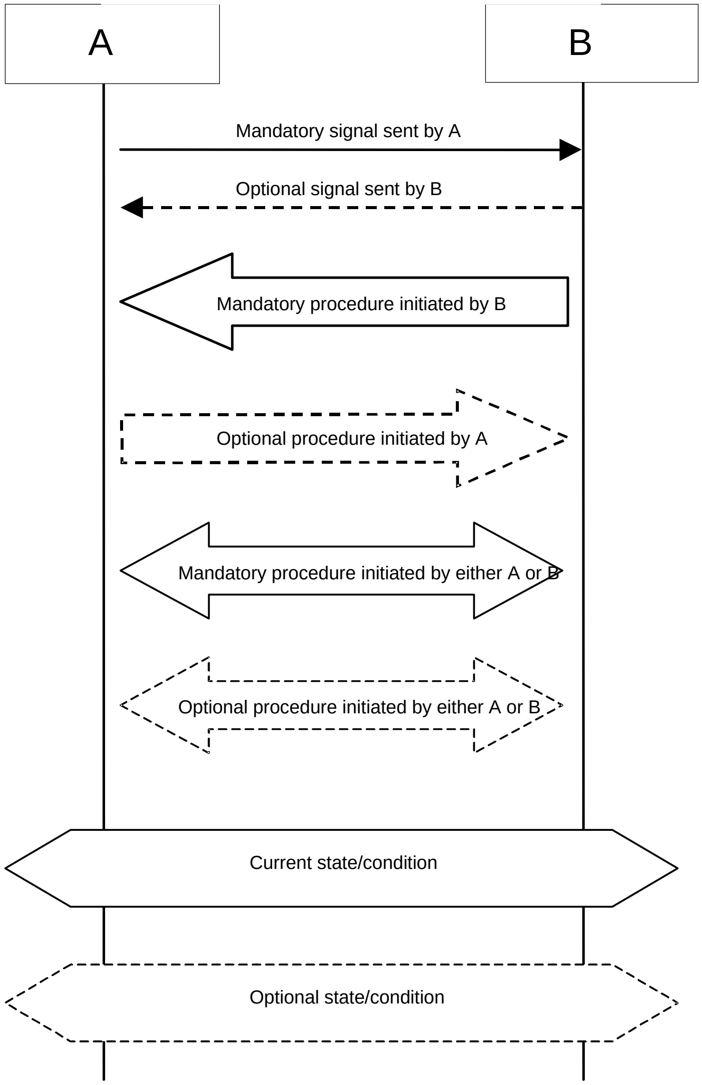
A
B
Mandatory signal sent by A
Optional signal sent by B
Mandatory procedure initiated by B
Optional procedure initiated by A
Mandatory procedure initiated by either A or B
Optional procedure initiated by either A or B
Current state/condition
Optional state/condition

Figure 1.1: Conventions used in signalling diagrams

2 Profile Overview

2.1 Profile stack

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

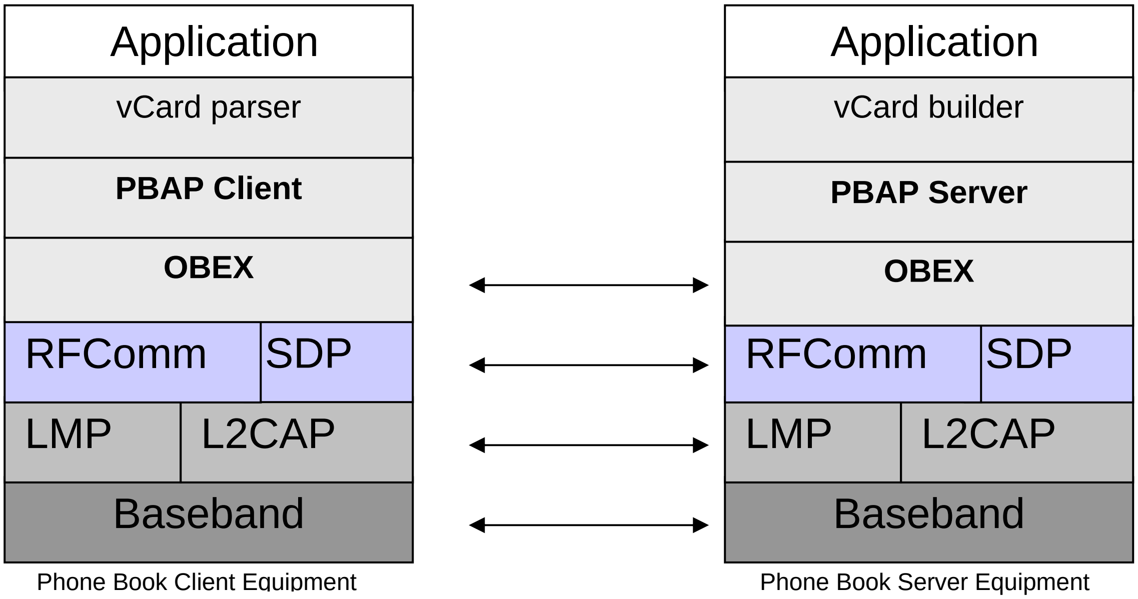

Figure 2.1: Profile Stack

The Baseband, LMP and L2CAP are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM is the Bluetooth serial port emulation entity. SDP is the Bluetooth Service Discovery Protocol. See [14] for more details on these topics.
Compatibility to the current Bluetooth Core specification 1.2 and later (see [14]) is mandated.
The PBAP session is defined as the underlying OBEX session between the client and the server opened with the PBAP Target UUID [See section 6.4].

2.2 Configuration and roles

The figure below shows a typical configuration of devices for which the Phone Book Access Profile is applicable:

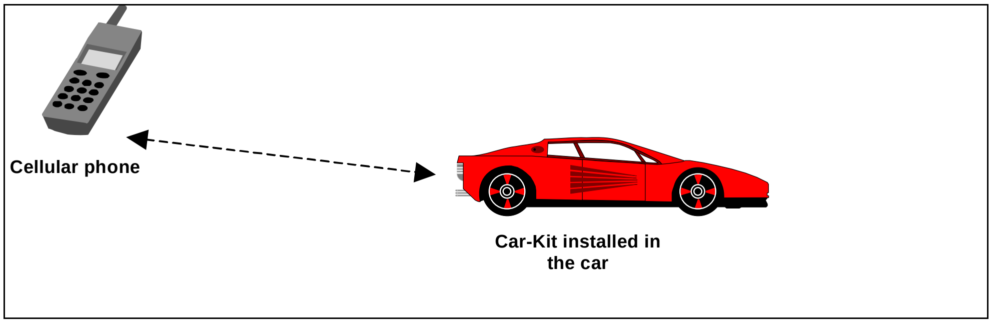

Figure 2.2: Phone Book Access Profile applied to the Hands-Free use case

The following roles are defined for this profile:

Phone Book Server Equipment (PSE) – This is the device that contains the source phone book objects.

Phone Book Client Equipment (PCE) – This is the device that retrieves phone book objects from the Server Equipment.

These terms are used in the rest of this document to designate these roles.

For the Hands-Free use case, a typical configuration would be a mobile phone as PSE and a Hands-Free car kit as PCE.

2.3 User requirements and scenarios

The following are some of the main scenarios that are covered by this profile:

The PCE to access the list of phone book entries stored in the PSE

The PCE to down load one or several phone book entries from the PSE

The PCE to access the call histories stored in the PSE

The PCE to access the Subscriber number information stored in the PSE

2.4 Profile fundamentals

The Phone Book Client Equipment may be able to use the services of the Phone Book Server Equipment only after a successful creation of a secure connection. Before a Phone Book Client Equipment can use the services of Phone Book Server Equipment for the first time, the two devices shall bond. Initialization includes exchanging a PIN code (Bluetooth Passkey), creation of link keys, encryption and service discovery. Either the PSE or PCE may initiate bonding. As a minimum the PSE shall support Inquiry in order to initiate bonding. Both PSE and PCE shall support Inquiry Scan Mode in order to accept bonding

2.5 Bluetooth Security

The two devices shall create a secure connection using the GAP authentication procedure as described in Section 5.1 of the Generic Access Profile [6]. This procedure may include entering a PIN code and will include creation of link keys. A fixed PIN code may also be used during the GAP authentication procedure.

The Phone Book Access Profile mandates the use of several Bluetooth security features:

Bonding – The PCE and PSE shall be bonded before setting up a Phone Book Access Profile connection. Either security mode 2 or 3 shall be used for the Phone Book Access Profile connection.

Encryption - The link between PCE and PSE shall be encrypted using Bluetooth encryption.

Bluetooth Passkey - The PCE and PSE shall prohibit the use of a zero-length Bluetooth passkey.

Furthermore, the following issues are mandated for devices complying with the Phone Book Access Profile:

Link keys – Combination keys shall be used for Phone Book Access Profile connections.

Encryption key length - The length of the encryption key shall be at least 64 bits. For increased security, use of the maximum length allowed given regional regulation is encouraged.

User confirmation - The PSE user shall have to confirm at least the first Phone Book Access Profile connection from each new PCE.

2.6 Conformance

If conformance to this profile is claimed all capabilities indicated as mandatory for this profile shall be supported in the specified manner (process mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory, optional and conditional capabilities, for which support is indicated, are subject to verification as part of the Bluetooth certification program.

3 Application layer

3.1 Phone Book Access Profile Objects and Formats

3.1.1 Phone Book Repositories

There might be several repositories for phone book objects. A typical example is that of a GSM mobile phone where one phone book is stored locally in the phone's memory and another phone book is stored on the phone's SIM card.

3.1.2 Phone Book Objects

There are 5 types of phone book objects:

- The main phone book object (pb) corresponds to the user phone book of the current repository. In case the PSE is a mobile phone, pb is the content of the contact list stored in the phone, whereas for a SIM, it is the contact list stored in the SIM Card.
- The Incoming Calls History object (ich) corresponds to a list of the M most recently received calls. The number of entries in this list, M, is dependent on the implementation of the connected device.
- The Outgoing Calls History object (och) corresponds to a list of the P most recently made calls. The number of entries in this list, P, is dependent on the implementation of the connected device.
- The Missed Calls History object (mch) corresponds to a list of the Q most recently missed calls. The number of entries in this list, Q, is dependent on the implementation of the connected device.
- The Combined Calls History object (cch) corresponds to the combination of ich, och and mch. The number of entries in the list, R, is dependent on the implementation of the connected device.

More information on pb, ich, och and mch can be found in the IrMC specification [13]. Cch is an extension to IrMC specific to the present profile.

3.1.3 Phone Book object representations

Each phone book object has 2 representations:

- File representation:
  In this representation, the phone book object is represented as one single file that contains all of the corresponding phone book entries. This representation corresponds to the IrMC [13] level 2 information exchange.
- Folder representation:
  In this representation, the phone book object is presented as a virtual folder that contains the corresponding phone book entries, each being represented as an individual file. This representation corresponds to the IrMC [13] level 3 information exchange.

Note that the folder representation of ich, och, mch and cch are extensions to IrMC.

3.1.4 Phone Book Entries format

Each individual entry in a phone book object is presented under the vCard format.

The PSE shall support both vCard 2.1 and vCard 3.0 versions and deliver the Entries to the PCE under the format version that is requested by the PCE.

Whatever the vCard format requested, the character set used to encode vCard attribute content shall be UTF-8.

Whenever the CHARSET property parameter is used in a vCard to override default character set, UTF-8 is the only accepted value for this parameter in this profile.

3.1.4.1 Call History extension

The time of each call found in och, ich, mch and cch folder, can be shown using the IrMC [13] defined X-IRMC-CALL-DATETIME property that extends the vCard specification. This attribute can be used in combination with three newly created property parameters:

- MISSED
- RECEIVED
- DIALED

These are used to indicate the nature of the call that is time-stamped with X-IRMC-CALL-DATETIME.

For instance, a call that was missed on March 20th, 2005 at 10 am would be stamped:

X-IRMC-CALL-DATETIME;MISSED:20050320T100000

It is strongly recommended to use this property parameter whenever possible. They are especially useful in vCards that are retrieved from the cch folder (see 3.1.2).

Note that it is legal to use this property with no data ie,

X-IRMC-CALL-DATETIME;MISSED:

This scenario may occur if the device did not have the time/date set when the call was received. The phone number would be recorded but no date/time could be attached to it. It will still need to be added to the vCard as the cch log needs it to indicate the type of call that the record identifies.

3.1.5 PBAP virtual folders structure

Phone book information is organized under the virtual folders architecture described in Figure 3.1.

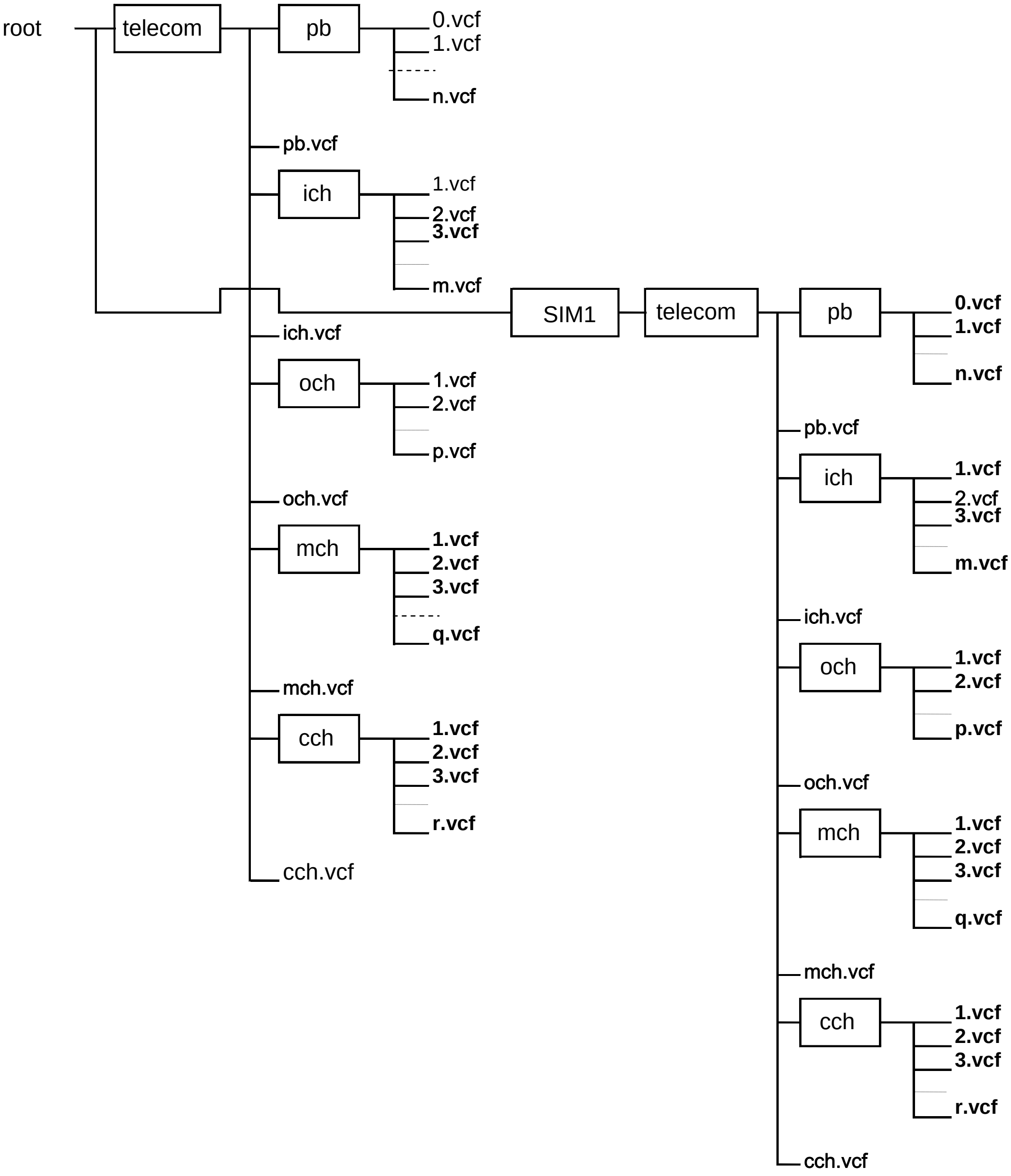

Figure 3.1: PBAP virtual folders architecture

3.1.5.1 Handles

All the vCards of the PSE are identified by their handle (<**handle.vcf**>). A handle is 32 bit value, represented as up to 8 hexidecimal digits.

The present profile has been developed on the assumption that the handles are not modified during the lifespan of a given PBAP session i.e. that the internal index and the actual content of each vCard does not change. If for some reason the implementation allows for handle modifications (including possibly deletion operations) during an active PBAP session, it shall support either of the following 2 behaviours:

Error reporting: If any of the handles are modified/deleted during an active PBAP session, then the PSE shall return Error "Precondition failed" to all read requests until the PCE has updated the vCard-listing.

Change tracking: The PSE keeps track of all the changes that are brought to the handles during the active PBAP session.

3.1.5.2 Local Phone Books: PB and SIM1

The local phone book information is located under the telecom folder. When the PSE also contains a SIM card, SIM card phone book information shall be located under the SIM1/telecom/ folder.

The special handle **0.vcf** in the pb folder is reserved for the owner card. It shall always be present and contain at least the mobile number of the PSE, whenever the PSE has the knowledge of this number. When the owner card is not known, handle 0.vcf can be an empty vCard, or a vCard that contains the mobile number of the PSE, if the PSE has a mobile number and does know it. The 0.vcf shall always be present in the pb.vcf file.

The vCards located in the pb folder are in ascendant handles numbering order. When the vCard-listing object for pb is requested, the default order shall be ascendant handles numbering order (see section 3.1.6)

The vcf files located directly under the “telecom” and/or ”SIM1/telecom” folders contain all the vCards of the corresponding phone book object and can be used to retrieve the entire object in one operation (see section 4.2, Phone Book Download Feature).

3.1.5.3 Call Histories

The handles of each call history vCard are specific to their folder. For instance, even if the same number/contact can be found in the och and ich folder, both handles, the one in ich and in och, are independent and don’t have to be the same.

Here is an example where the same number can be found in several call histories:

Ich folder	Handle	Och folder	Handle
+33642084141 – 02/18/2005 16:00	1.vcf	+4425724268 – 02/15/2005 17:30	1.vcf
+4425724268 – 02/18/2005 15:00	2.vcf	+33149046174 – 02/15/2005 17:00	2.vcf
		+33642084141 – 02/14/2005 09:00	3.vcf

The handles should be attributed in the PSE in such a way that once sorted by increasing handles; the most recent call event in the listed folder has the handle 1.vcf. The order is therefore chronological.

As opposed to the Local Phone Books, the Call Histories are using dynamic handles that shall be updated in real time as calls are received and made.

Neither search nor sorting on call histories folders is mandatory to support on the PSE. Specific search or sort request from the PCE should be replied with a vCard listing in chronological order.

If the number of a call history entry can be linked to a local PSE phone book entry (in pb or SIM1), the PSE should provide the contact information available in that corresponding phone book entry when the call history entry is retrieved. However, the TEL attribute of the call history entry shall contain only the phone number that was used for that call.

If the number could not be linked to a locally present PSE phone book entry, then the call history object shall contain attributes VERSION, N (in that case, N would be an empty field), FN for a vCard 3.0 only (in that case, FN would be an empty field), TEL if requested, and X-IRMC-CALL-DATETIME if requested and supported.

3.1.6 vCard-Listing Object (x-bt/vcard-listing)

The vCard-listing object is an XML object and is encoded in UTF-8 by default. The default character set may be omitted.

The vCard-Listing object is defined according to the following DTD:

```
<!DTD for the PBAP vCard-Listing Object-->

<!ELEMENT vcard-listing ( card )* >
<!ATTLIST vcard-listing version CDATA #FIXED “1.0”>

<!ELEMENT card EMPTY>
<!ATTLIST card
 handle CDATA #REQUIRED
 name CDATA #IMPLIED >
```

Example of vCard listing object:

```
<?xml version=“1.0”?>
<!DOCTYPE vcard-listing SYSTEM “vcard-listing.dtd”>
<vCard-listing version=“1.0”>
 <card handle = ”0.vcf” name = ”Miyajima;Andy”/>
 <card handle = ”1.vcf” name = ”Poujade;Guillaume”/>
 <card handle = ”2.vcf” name = ”Hung;Scott”/>
 <card handle = ”3.vcf” name = ”Afonso;Arthur”/>
 <card handle = ”6.vcf” name = ”McHardy;Jamie”/>
 <card handle = ”7.vcf” name = ”Toropov;Dmitri”/>
 <card handle = ”10.vcf” name = ”Weinans;Erwin”/>
</vCard-listing>
```

3.1.6.1 Name attribute format

The name attribute of the vCard-listing DTD has the same definition as the name attribute of a vCard - i.e., the N attribute.. Therefore the format to be used is the fields structure : “LastName;FirstName;MiddleName;Prefix;Suffix”

Regardless of the native structure used internally for storing name information, all PSEs shall be able to format the name information of the name attribute of the vCard-listing object according to one of the following methods:

- The PSE provides all the information in the LastName field of the name attribute and the other fields are empty.
- The internal name structure maps directly into the “LastName; FirstName; MiddleName;Prefix;Suffix” structure of the name attribute and no conversion is required. Some fields may be empty.
- From its native structure, the PSE constructs a new representation of the name information that fits into the “LastName; FirstName; MiddleName;Prefix;Suffix” structure of the name attribute. Some fields may be empty.

In the case of the vcard-listing of one of the calls history phonebook objects, and when a call history entry cannot be linked to a local entry in the main phonebook object of the PSE, the PSE may return the phone number corresponding to the call history entry as a string in the name property.

4 Phone Book Access Features

4.1 Phone Book Access Profile Features

The current profile is composed of 2 features. For a device to comply with this specification, it shall observe the following implementation requirements table:

Feature	Support by the PCE	Support by the PSE
Download	C1	M
Browsing	C1	M

C1: at least one of those shall be supported.

The tables in 4.2 and 4.3 map each feature to the functions used for that feature. The columns indicate the level of support required for each function.

4.2 Phone Book Download Feature

This feature is used to download the entire content of a phone book object. It is especially designed for devices with relatively large storage capacity that are typically attempting to download and locally store the entire phone book.

Note that this feature is very basic. In particular, phone book entries are delivered in the ascendant index order only. No sorting is possible.

	Function	Support by the PCE	Support by the PSE
Phone book Download feature	PullPhonebook	M	M

The PullPhonebook function is used to download the phone book object of interest.

A typical function sequence for the Phone Book Download feature is illustrated in Figure 4.1:

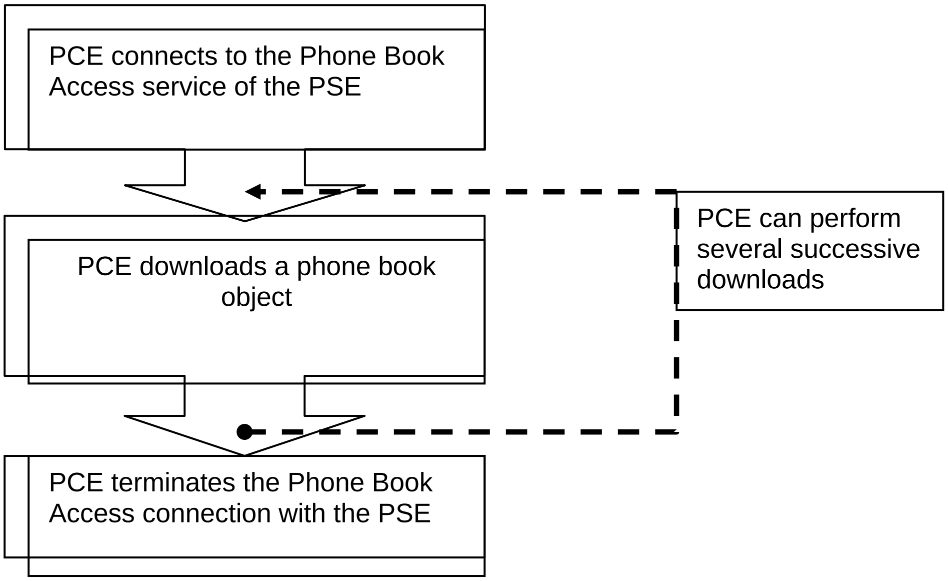
PCE connects to the Phone Book Access service of the PSE
PCE downloads a phone book object
PCE can perform several successive downloads
PCE terminates the Phone Book Access connection with the PSE

Figure 4.1: Phone Book Download Sequence

4.3 Phone Book Browsing Feature

This feature is especially suitable for applications where there is a need to scroll the phone book.

	Function	Support by the PCE	Support by the PSE
Phone Book Browsing Feature	SetPhonebook	M	M
	PullvCardListing	M	M
	PullvCardEntry	M	M

The SetPhonebook allows for the PCE to select the phone book object of interest.

The PullvCardListing function allows for the PCE to retrieve a list that represents the content of a given phone book object. The list can be tailored so that it contains only the entries of interest and/or is sorted in the order of interest.

The PullvCardEntry is used by the PCE to retrieve an individual entry from the phone book object of interest.

A typical function sequence for the Phone Book Browsing feature is illustrated in Figure 4.2:

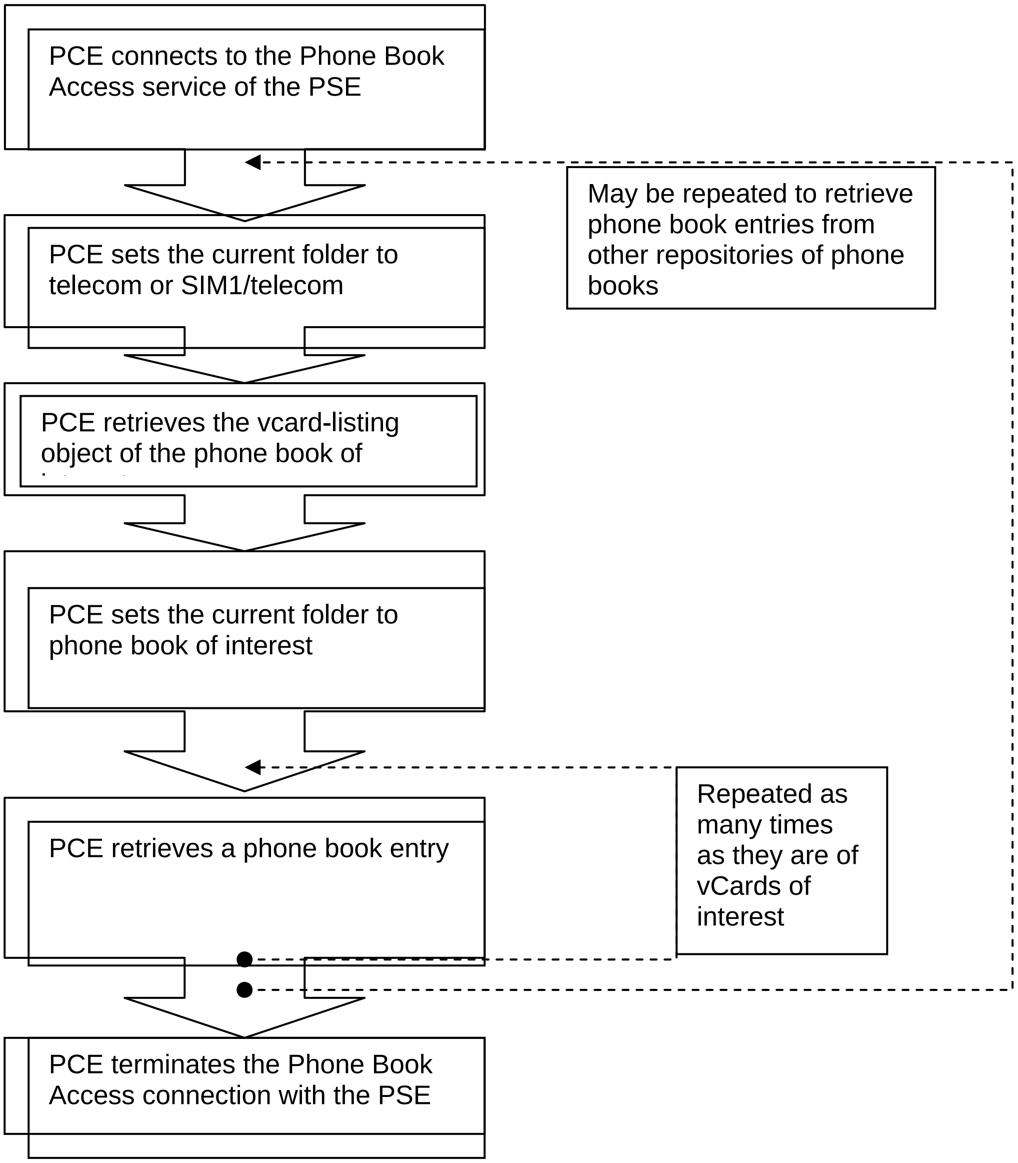
PCE connects to the Phone Book Access service of the PSE
May be repeated to retrieve phone book entries from other repositories of phone books
PCE sets the current folder to telecom or SIM1/telecom
PCE retrieves the vcard-listing object of the phone book of
PCE sets the current folder to phone book of interest
Repeated as many times as they are of vCards of interest
PCE retrieves a phone book entry
PCE terminates the Phone Book Access connection with the PSE

Figure 4.2: Phone Book Browsing Sequence

5 Phone Book Access Profile Functions

5.1 PullPhoneBook Function

This function retrieves an entire phone book object from the object exchange server. The Pull Phone Book function is typically used from the root folder. However, when an implementation is using both phone book downloading and phone book browsing services, Pull Phone Book can be used from whichever virtual folder the PSE is currently set to.

The request is formatted as follows:

Fields	Opcode Packet Length	Get Length of the packet
Headers	Connection ID Name Type Application parameters	Connection Identifier Object name (*.vcf) x-bt/phonebook Filter Format MaxListCount ListStartOffset

The vCard objects in the response shall only contain the attributes as indicated using the **Filter** parameter and shall be presented using the Format as indicated using the **Format** parameter.

The response is formatted as follows

Fields	Response Code Packet Length	Success or error code Length of the packet
Headers	Body/EndOfBody Application Parameters	vcard object PhonebookSize NewMissedCalls

The vCard object to be returned in the Body header contains the file representation of the phone book object of interest.

5.1.1 Connection ID

The connection Id, received during the connection establishment shall be used to identify the connection. See the OBEX specification for details about the usage of the Connection ID header.

5.1.2 Name

The Name header shall contain the absolute path in the virtual folders architecture of the PSE, appended with the name of the file representation of one of the Phone Book Objects.

Example: telecom/pb.vcf or SIM1/telecom/pb.vcf for the main phone book objects.

5.1.3 Type

The type header shall be used to indicate the type of object to be received. In this function its content shall be: **<x-bt/phonebook>** for phone book objects.

5.1.4 Application Parameters Header

This header is an extendable header to be specialized by the application above the OBEX layer. The TagID of the different parameters are described in 6.2.1.
Refer to [10], [11], [12] for the complete specification of the vCard formats.

5.1.4.1 Filter {AttributeMask 64-bit value)}

This header is used to indicate the attributes contained in the requested vCard objects. The PCE can use this header to receive only the relevant content of the requested vCard. All attributes of the vCard shall be returned if this header is not specified or carries the value 0x00000000.

Mandatory attributes for vCard 2.1 are VERSION ,N and TEL.

Mandatory attributes for vCard 3.0 are VERSION, N, FN and TEL

Mandatory here means that the PSE shall always return the attributes VERSION, N and TEL for a vCard 2.1 or VERSION, N, FN and TEL for a vCard 3.0.

With the exception of the above attributes, the attributes that are requested by the PCE can be ignored by the PSE in case the PSE doesn’t support them. However, all the attributes that are indeed supported by the PSE shall be returned if their value is not null. For instance, if the PSE supports EMAIL and TITLE (in addition to the mandatory VERSION, N and TEL in case of vCard 2.1) and the PCE requests the EMAIL and BDAY attributes, the PSE shall include the EMAIL attribute in the vCard objects that are returned to the PCE whenever the EMAIL attribute’s value is not null.

The PSE is not allowed to return any other attributes than those requested by the PCE.

In the case of multiple instances of the same attribute (for instance when more than 1 telephone number is present in a phone book entry), all the instances shall be returned by the PSE.

bit 0	VERSION	vCard Version
Bit 1	FN	Formatted Name
bit 2	N	Structured Presentation of Name
bit 3	PHOTO	Associated Image or Photo
bit 4	BDAY	Birthday
bit 5	ADR	Delivery Address
bit 6	LABEL	Delivery
bit 7	TEL	Telephone Number
bit 8	EMAIL	Electronic Mail Address
bit 9	MAILER	Electronic Mail
bit 10	TZ	Time Zone
bit 11	GEO	Geographic Position
bit 12	TITLE	Job
bit 13	ROLE	Role within the Organization
bit 14	LOGO	Organization Logo
bit 15	AGENT	vCard of Person Representing
bit 16	ORG	Name of Organization
bit 17	NOTE	Comments
bit 18	REV	Revision
bit 19	SOUND	Pronunciation of Name
bit 20	URL	Uniform Resource Locator
bit 21	UID	Unique ID
bit 22	KEY	Public Encryption Key
bit 23	NICKNAME	Nickname
bit 24	CATEGORIES	Categories
bit 25	PROID	Product ID
bit 26	CLASS	Class information
bit 27	SORT-STRING	String used for sorting operations
bit 28	X-IRMC-CALL-DATETIME	Time stamp
bit 29~38		Reserved for future use
bit 39	Proprietary Filter	Indicates the usage of a proprietary filter
bit 40 ~ 63		Reserved for proprietary filter usage

Table 5.1: Attribute Mask

Bit = 1 indicates that the value shall be present in the requested vCard if available.

The reserved bits must be set to 0.

Bit 39 is used to indicate the presence of a proprietary AttributesMask. A proprietary AttributeMask is a mask that describes the requirements of the PCE for proprietary vCard attributes. Proprietary vCard attributes are vendor specific. For the proprietary

AttributeMask to be properly used, it is necessary that the PCE identify the PSE as being a device that uses the same set of proprietary vCard attributes.

5.1.4.2 Format { vCard2.1 | vCard3.0 }

This header is used to indicate the requested format (vCard 2.1 or 3.0) to be returned in the operation. The format vCard 2.1 shall be the default format if this header is not specified.

5.1.4.3 MaxListCount

This header is used to indicate the maximum number of entries of the **<x-bt/phonebook>** object that the PCE can handle. This header shall always contain a value between 0 and 65535. The value 65535 means that the number of entries is not restricted.

MaxListCount = 0 signifies to the PSE that the PCE wants to know the number of used indexes in the phone book of interest.

When MaxListCount = 0, the PSE shall ignore all other application parameters that may be present in the request. The response shall not contain any Body header.

5.1.4.4 ListStartOffset

This header shall be used to indicate the offset of the first entry of the **<x-bt/phonebook>** object compared to the first entry of the Phonebook object that would be returned if the ListStartOffset parameter was not specified in the request. The offset shall be 0 if this header is not specified.

5.1.4.5 PhonebookSize

This application parameter is used in the response when the value of MaxListCount in the request is 0. It shall contain the number of indexes in the phone book of interest that are actually used (i.e. indexes that correspond to non-null entries).

5.1.4.6 NewMissedCalls

This application parameter shall be used in the response when and only when the phone book object is mch. It indicates the number of missed calls that have not been checked on the PSE at the point of the request.

5.2 SetPhoneBook Function

This function sets the current folder in the virtual folder architecture.

The request is formatted as follows:

Fields	Opcode Packet Length	SetPath Length of the packet
Headers	Connection ID Name	Connection Identifier Name of the folder

The response is formatted as follows:

Fields	Response Code Packet Length	Success or error code Length of the packet
Headers		

Note that the OBEX SetPath Command (the basis for the SetPhoneBook function) allows only for setting the current folder to the root, parent or a child folder. For example in order to set the current phone book folder to "pb", from the root, it is necessary to apply SetPath twice: the first is necessary to change into "telecom" and the second is necessary to change into "pb".

The usage of the flags and Name header is summarized hereunder:

5.3 PullvCardListing Function

	Go back to root	Go down 1 level	Go up 1 level
Flags / bit 0 :	0	0	1
bit 1 :	1	1	1
bit 2~7 :	0	0	0
Name header:	empty	name of the child folder	optional

This function retrieves the PSE’s Phonebook-listing object.

The request is formatted as follows:

Fields	Opcode Packet Length	Get Length of the packet
Headers	Connection ID	Connection Identifier
	Type	x-bt/vcard-listing
	Name	Name of the folder
	Application parameters	Order
		SearchValue
		SearchAttribute
		MaxListCount
		ListStartOffset

The response is formatted as follows:

Fields	Response Code Packet Length	Success or error code Length of the packet
Headers	Body/EndOfBody Application Parameters	vcard-listing object PhonebookSize NewMissedCalls

The vCard handles in the vCard-listing object shall be ordered as directed in the **Order** parameter. Only vCards matching the **SearchValue** for the attribute as indicated in the **SearchAttribute** shall be contained in the listing object. The vCard-listing object shall contain up to **MaxListCount** vCard entries, starting with an offset as indicated using the **ListStartOffset** parameter.

5.3.1 Connection ID

The connection Id, received during the connection establishment shall be used to identify the connection.

5.3.2 Type

The type header shall be used to indicate the type of object to be received. In this function its content shall be: **<x-bt/vcard-listing>** for phone book folder listing object.

5.3.3 Name

This is the name of the folder to be retrieved. The value shall not include any path information, since the PullvCardListing function uses relative paths. An empty name header may be sent to retrieve the folder that is the current directory.

5.3.4 Application Parameters

5.3.4.1 Order { Alphabetical | Indexed | Phonetical }

This header is used to indicate to the Server, which sorting order shall be used for the **<x-bt/vcard-listing>** listing object. The ordering is always ascendant.

Alphabetical order: The sorting operation should be based on the N attribute of the selected folder entries. Sorting may be performed on LastName, then Firstname, then MiddleName; but other sorting algorithms are also allowed. In particular, it is possible for the PSE to apply the same algorithm as the one it uses for the sorting function of its native user interface.It is recommended to perform the sorting operation after conversion of the Name attribute into UTF-8.

Indexed order: The vCards are ordered by increasing handle order.

Phonetically ordered: The sorting algorithm should be based on the SOUND attribute. Note that sorting on the Sound attribute makes sense only when the SOUND attribute values of the phonebook objects on the PSE are in textual form. It is recommended to perform the sorting operation after conversion of the Sound attribute into UTF-8.

If this application parameter is not specified, the default order is “Indexed”.

5.3.4.2 SearchAttribute {Name | Number | Sound }

This header shall be used to indicate to the Server, which vCard attribute the search operation shall be carried out on. If this header is not specified, the search shall be performed on the “Name” attribute.

Note that searching on the Sound attribute makes sense only when the Sound attribute values of the phonebook objects on the PSE are in textual form.

5.3.4.3 SearchValue {<text string>}

This header shall be used to indicate to the Server, which vCards shall be contained in the **<x-bt/vcard-listing>** listing object. The vCard is listed only if the <text string> matches the value of the attribute indicated using the Search/Attribute. All the vCards shall be returned if this header is not specified.

The UTF-8 character set shall be used for <text_string> in this header.

The matching routine is implementation specific.

5.3.4.4 MaxListCount

This header shall be used to indicate the maximum number of entries of the **<x-bt/vcard-listing>** listing object. The maximum number of entries shall be 65535 if this header is not specified.

5.3.4.5 ListStartOffset

This header shall be used to indicate the offset of the first entry of the **<x-bt/vcard-listing>** listing object compared to the first entry of the vcard-listing object that would be returned if the MaxListCount parameter was not specified in the request. The offset shall be 0 if this header is not specified.

5.3.4.6 PhonebookSize

This application parameter shall be used in the response when the value of MaxListCount in the request is 0. MaxListCount=0 signifies to the PSE that the PCE wants to know the number of indexes in the phone book of interest that are actually used (i.e. indexes that correspond to non-null entries). When MaxListCount = 0, the PSE shall ignore all other application parameters that may be present in the request. The response shall not contain any Body header.

5.3.4.7 NewMissedCalls

This application parameter shall be used in the response when and only when the phone folder is mch. It indicates the number of missed calls that have not been checked on the AG at the point of the request.

5.4 PullvCardEntry Function

This function retrieves a specific vCard from the object exchange server.

The request is formatted as follows:

Fields	Opcode Packet Length	Get Length of the packet
Headers	Connection ID	Connection Identifier
	Name Type Application parameters	Object name (*.vcf) x-bt/vcard Filter Format

The vCards shall only contain the attributes as indicated using the **Filter** parameter and shall be presented using the Format as indicated using the **Format** parameter.

The response is formatted as follows

Fields	Response Code Packet Length	Success or error code Length of the packet
Headers	Body/EndOfBody	vcard object

5.4.1 Connection ID

The connection Id, received during the connection establishment shall be used to identify the connection. See the OBEX specification for details about the usage of the Connection ID header.

5.4.2 Name

The Name header shall be used to indicate the name of the object to be retrieved. No path information shall be included since the PullvCardEntry function uses relative path information.

5.4.3 Type

The type header shall be used to indicate the type of object to be received. In this function its content shall be: **<x-bt/vcard>** for Generic vCard object.

5.4.4 Application parameters

Refer to [10], [11], [12] for the complete specification of the vCard formats.

5.4.4.1 Filter {AttributeMask (64-bit value)}

This header is used to indicate the attributes contained in the requested vCard objects. The PCE can use this header to receive only the relevant content of the requested vCard. All attributes of the vCard shall be returned if this header is not specified or carries the value 0x00000000.

Mandatory attributes for vCard 2.1 are VERSION, N and TEL.

Mandatory attributes for vCard 3.0 are VERSION, N,FN and TEL.

Mandatory here means that the PSE shall always return the attributes VERSION, N and TEL for a vCard 2.1 or VERSION, N, FN and TEL for a vCard 3.0.

With the exception of the above attributes, the attributes that are requested by the PCE can be ignored by the PSE in case the PSE doesn’t support them. However, all the attributes that are indeed supported by the PSE shall be returned if their value is not null. For instance, if the PSE supports EMAIL and TITLE (in addition to the mandatory VERSION, N and TEL in case of vCard 2.1) and the PCE requests the EMAIL and BDAY attributes, the PSE shall include the EMAIL attribute in the vCard objects that are returned to the PCE whenever the EMAIL attribute’s value is not null.

The PSE is not allowed to return any other attributes than those requested by the PCE.

In the case of multiple instances of the same attribute (for instance when more than 1 telephone number is present in a phone book entry), all the instances shall be returned by the PSE.

bit 0	VERSION	vCard Version
Bit 1	FN	Formatted Name
bit 2	N	Structured Presentation of Name
bit 3	PHOTO	Associated Image or Photo
bit 4	BDAY	Birthday
bit 5	ADR	Delivery Address
bit 6	LABEL	Delivery
bit 7	TEL	Telephone Number
bit 8	EMAIL	Electronic Mail Address
bit 9	MAILER	Electronic Mail
bit 10	TZ	Time Zone
bit 11	GEO	Geographic Position
bit 12	TITLE	Job
bit 13	ROLE	Role within the Organization
bit 14	LOGO	Organization Logo
bit 15	AGENT	vCard of Person Representing
bit 16	ORG	Name of Organization
bit 17	NOTE	Comments
bit 18	REV	Revision
bit 19	SOUND	Pronunciation of Name
bit 20	URL	Uniform Resource Locator
bit 21	UID	Unique ID
bit 22	KEY	Public Encryption Key
bit 23	NICKNAME	Nickname
bit 24	CATEGORIES	Categories

bit 0	VERSION	vCard Version
bit 25	PROID	Product ID
bit 26	CLASS	Class information
bit 27	SORT-STRING	String used for sorting operations
bit 28	X-IRMC-CALL-DATETIME	Time stamp
bit 29~38		Reserved for future use
bit 39	Proprietary Filter	Indicates the usage of a proprietary filter
bit 40 ~ 63		Reserved for proprietary filter usage

Table 5.2 Attribute Mask

Bit = 1 indicates that the value shall be present in the requested vCard if available.

The reserved bits must be set to 0.

Bit 39 is used to indicate the presence of a proprietary AttributesMask. A proprietary AttributeMask is a mask that describes the requirements of the PCE for proprietary vCard attributes. Proprietary vCard attributes are vendor specific. For the proprietary AttributeMask to be properly used, it is necessary that the PCE identify the PSE as being a device that uses the same set of proprietary vCard attributes. The procedure how the PCE can identify the PSE is out of scope of the present specification.

5.4.4.2 Format { vCard2.1 | vCard3.0 }

This header is used to indicate the requested format (vCard 2.1 or 3.0) to be returned in the operation. The format vCard 2.1 shall be the default format if this header is not specified.

6 OBEX

6.1 OBEX Operations Used

Table 6.1 lists the OBEX operations required by the Phone Book Access Profile.

OBEX operation	**Ability to send**	**Ability to respond**
	PCE	**PSE**
Connect	M	M
Disconnect	M	M
Get	M	M
Abort	M	M
SetPath	C1	M

C1: support required only if the phone book Browsing feature is implemented.

Table 6.1 OBEX Operations

6.2 OBEX Headers

Table 6.2 lists the OBEX headers required by the Phone Book Access Profile.

OBEX header	**PCE**	**PSE**
Name	M	M
Type	M	M
Body	M	M
End of Body	M	M
Target	M	M*
Who	M*	M
Connection ID	M	M
Authentication Challenge	M	M
Authentication Response	M	M
Application Parameters	M	M

Table 6.2 OBEX Headers

* Ability to parse only (ability to send is not required).

Note that the profile does not exclude the headers that are not listed in Table 6.2. Some implementations might choose to use additional headers to enable added value services. Therefore unknown or unsupported headers shall always be ignored.

6.2.1 Application Parameters Header

The tag IDs used in the Application Parameters header are listed below. All of the Application Parameter header values use big-endian byte ordering.

Value	Tag ID	Length	Possible Values
Order	0x01	1 byte	0x00 = indexed 0x01 = alphanumeric 0x02 = phonetic
SearchValue	0x02	variable	Text
SearchAttribute	0x03	1 byte	0x00= Name 0x01= Number 0x02= Sound
MaxListCount	0x04	2 bytes	0x0000 to 0xFFFF
ListStartOffset	0x05	2 bytes	0x0000 to 0xFFFF
Filter	0x06	8 bytes	64 bits mask
Format	0x07	1 byte	0x00 = 2.1 0x01 = 3.0
PhonebookSize	0x08	2 bytes	0x0000 to 0xFFFF
NewMissedCalls	0x09	1 byte	0x00 to 0xFF

Table 6.3 Application Parameter Headers

6.2.2 OBEX Headers in Multi-Packet Responses

In the case of multi-packet responses, to encourage interoperability, the Phone Book Access Profile requires that all the headers other than the BODY header shall be placed in the first packet. The object body is sent on the subsequent packets, but can start from the first packet if there is room left in that packet and the request phase of the GET operation is complete (i.e. the GET packet had the final bit set).

In case of a multi-packet request, all OBEX request messages shall be sent as one OBEX packet containing all the headers. I.e. OBEX GET with opcode 0x68 shall always be used. OBEX GET with opcode 0x03 shall never be used.

6.2.3 OBEX Error Codes

The only mandatory error codes for the PSE are:

- Bad Request.
- Not implemented.
- Not Found
- Service unavailable

However, the PSE can use all the error codes listed in Table 6-4 to provide a more detailed error report to the PCE:

Error Code	PCE(interprets the Error Codes)	PSE(informs of Errors)	Meaning in the Phone Book Access Profile
Bad Request	M*	M	Function not recognized or ill-formatted
Not implemented	M*	M	Function recognized but not supported

Error Code	PCE(interprets the Error Codes)	PSE(informs of Errors)	Meaning in the Phone Book Access Profile
Unauthorized	M*	O	In operations with actual exchange of an object in the body header (either in the request or the response), indicates that the function was recognized, but that the object to be handled is protected and access is not authorized (either temporarily or permanently).
Precondition Failed	M*	C1	The function was recognized but there is a problem with one of the request's parameter values
Not Found	M*	M	The function was recognized and all the parameters are proper, but the vCard handle or the phone book object could not be found.
Not Acceptable	M*	O	The request is recognized and all the parameter values are legal, but there is a problem with a parameter value that indicates a request that cannot be met by the Server.
Service unavailable	M*	M	The function was recognized and is normally executable, but a system condition prevents it from being performed. It could be for instance that the phone book object is unavailable when the PSE is engaged in a call.
Forbidden	M*	O	Function recognized and correctly formatted but temporarily barred

Table 6.4 Error Codes

* Indicates that the Client shall recognize this response code as an error code.
C1 Indicates that the error code is mandatory if the PSE does not support the change tracking mechanism described in 3.1.5.1

On the PCE side, all of the response codes listed above must be recognized as error codes; how to handle these error codes is left to the implementer's discretion.

Support for response codes other than Bad Request, Not Implemented, Service unavailable and Not found is optional; it is recommended, however, that as many of the others as possible be supported because they are more informative and give the PCE a better indication of the nature of an error; this permits better error reporting. The other OBEX error codes shall not be used.

Any attempt at performing an OBEX PUT operation during the lifetime of the PBAP session shall be rejected, with Bad Request as error code.

The "x complements y" relationship between response codes is illustrated in Figure 6.1:

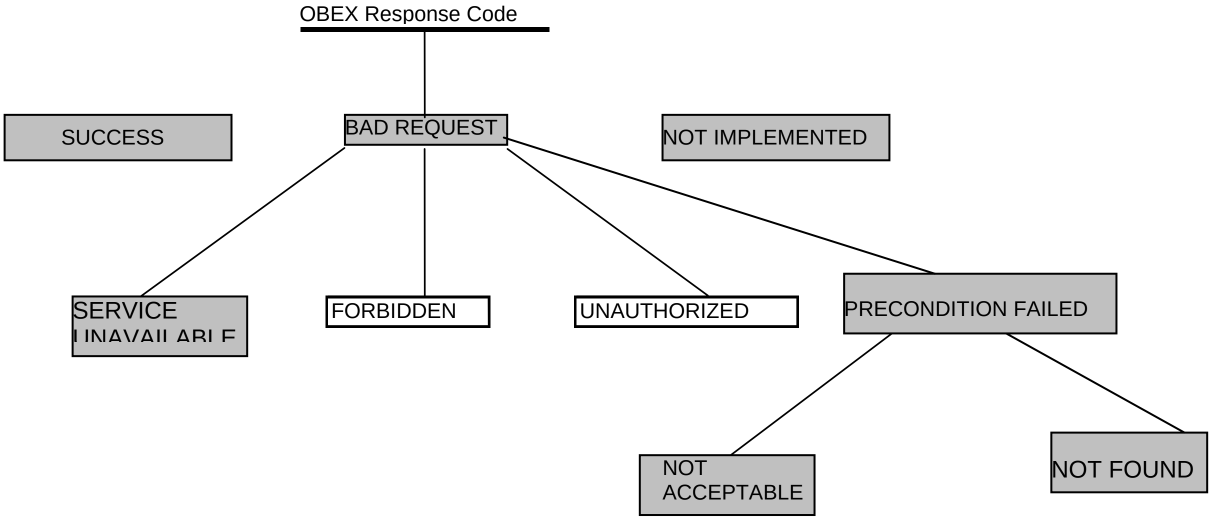

Figure 6.1: logical tree of the PBAP error codes:

The response codes in gray are the ones that the PSE shall support. The ones in white are optional for the PCE to support. "Precondition Failed" is conditionally mandatory.

When multi-packet responses are used, response codes must be returned as early as possible, preferably in the first response packet. In some cases – for example, Service Unavailable – it is possible that an error condition won't arise until the operation is underway, in which case it is acceptable to return a response code in a packet other than the first one.

6.3 Initializing OBEX

The initialization procedure is defined in Section 5.3 in [7].

Support for the OBEX authentication is mandatory, including support for OBEX user Ids as described in the IrOBEX specification, section 3.5.2.2. Whether or not it is actually used is left to the implementer's discretion.

Note that 2 devices with fixed OBEX PIN codes cannot be expected to connect if OBEX authentication is used.

When OBEX authentication is used, it is recommended to use alphanumerical PIN codes. However, depending on the application, it is allowed to use digits-only PIN codes.

The OBEX authentication can be either:

- mutual, with the first authentication requested by the PSE
- mutual, with the first authentication requested by the PCE
- one way, the PCE identifying the PSE.

6.4 Establishing an OBEX Session

See section 5.4 in the Bluetooth Generic Access Profile specification for a description of OBEX connection establishment without authentication.
The use of the Target header is mandatory in the Phone Book Access Profile. The

UUID to be used in the Target header is:
796135f0-f0c5-11d8-0966-0800200c9a66

6.5 Terminate an OBEX Session

The session can be terminated by the PCE by issuing a DISCONNECT command. When the OBEX session is terminated and the RFComm channel used for that session is not used by any other service, the RFComm channel shall be terminated, as described in [7].

It is also possible for the PSE to terminate the OBEX session; this is done in an indirect way, by closing the RFComm channel used for the OBEX session. The PCE shall be able to properly sustain such termination of the RFComm session.

7 Service Discovery

7.1 SDP Interoperability Requirements

The following service records are defined for the Phone Book Access Profile. There is one service record applicable to the PCE and another for the PSE.

7.1.1 SDP record for the PCE device

Item	Definition	Type	Value	Status	Default
ServiceClassID List				M	
ServiceClass #0		UUID	Phonebook Access Client	M	
ServiceName	Displayable Text name	String	Service-provider defined	O	"Phonebook Access PCE"
Bluetooth Profile Descriptor List				M	
Profile #0	Supported Profiles	UUID	Phonebook Access	M	
Param #0	Profile Version	Uint16		M	0x0100

This SDP record can be used in a multi-profiles environment: for instance, when a device A that supports, among other profiles, the PBAP PSE role attempts to pair with a device B that supports a multitude of profiles (including PBAP PCE role), it might want to inform its user that device B might request PBAP services and seek user agreement before allowing for link keys to be created and stored.

7.1.2 SDP record for the PSE device

Item	Definition	Type	Value	Status	Default
ServiceClassID List				M	
ServiceClass #0		UUID	Phonebook Access Server	M	
Protocol Descriptor List					
Protocol #0		UUID	L2CAP		
Protocol #1		UUID	RFCOMM		
Param #0	Channel number	Uint8	N= Channel number		
Protocol #2		UUID	OBEX		
ServiceName	Displayable Text name	String	Service-provider defined	O	"Phonebook Access PSE
Bluetooth Profile Descriptor List				M	
Profile #0	Supported Profiles	UUID	Phonebook Access	M	
Param #0	Profile Version			M	0x0100
Supported Repositories		Uint16 Uint8	Bit 0 = Local Phonebook Bit 1 = SIM card Bit 2~7 reserved for future use	M	

7.1.3 Class Of Device/Service Field

There is no obvious correlation between the Class of Device/Service Field and the support for the Phone Book Access Profile since it is expected that many different types of devices will implement the Phone Book Access Profile.

When Phone Book Access Profile is activated on the PSE, the PSE should ensure that the Object Transfer Bit is set in its CoD and register a service record in its SDDB.

8 Generic Access Profile

This profile requires compliance to the Generic Access Profile.

This section defines the support requirements for the capabilities as defined in the "Generic Access Profile".

8.1 Modes

The table shows the support status for GAP Modes in this profile.

	Procedure	Support in PCE	Support in PSE
1	**Discoverability modes**		
	Non-discoverable mode	O	O
	Limited discoverable mode	O	O
	General discoverable mode	M	M
2	**Connectability modes**		
	Non-connectable mode	N/A	N/A
	Connectable mode	O	M
3	**Pairing modes**		
	Non-pairable mode	O	O
	Pairable mode	M	M

Table 8.1: Modes

8.2 Security Aspects

The table shows the support for Security aspects within this profile.

	Procedure	Support in PCE	Support in PSE
1	Authentication	M	M
2	Security modes		
	Security mode 1	X	X
	Security mode 2	C1	C1
	Security mode 3	C1	C1
3	Encryption	M	M
C1: Support for at least one of the security modes 2 and 3 is mandatory.			

Table 8.2: Security Aspects

8.3 Idle Mode Procedures

The following table shows the support status for Idle mode procedures within this profile:

	Procedure	Support in PCE	Support in PSE
1	General inquiry	O	M
2	Limited inquiry	O	O
3	Name discovery	O	O
4	Device discovery	O	O
5	Bonding	M	M

Table 8.3: Idle Mode Procedures

8.3.1 Bonding

It is mandatory for the PSE to support initiation of bonding, and for the PCE to accept bonding.

9 References

[1] Bluetooth Specification [vol2]; Core System Package [controller volume]; Part B Baseband Specification.

[2] Bluetooth Specification [vol2]; Core System Package [controller volume]; Part C Link Manager Protocol.

[3] Bluetooth Specification [vol2]; Core System Package [controller volume]; Part E Host Controller Interface function Specification.

[4] Bluetooth Specification [vol3]; Core System Package [host volume]; Part A Logical Link Control and Adaptation Protocol specification

[5] Bluetooth Specification [vol3]; Core System Package [host volume]; Part B Service Discovery Protocol specification

[6] Bluetooth Specification [vol3]; Core System Package [host volume]; Part C Generic Access Profile specification

[7] Specification of the Bluetooth System, Bluetooth Generic Object Exchange Profile specification

[8] Specification of Bluetooth System, assigned numbers https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers

[9] IrOBEX specification, version 1.2, April, 1999

[10] VCard The Electronic Business Card, version 2.1, September 18th

[11] A MIME Content-Type for Directory Information, IETF, Network Working Group, RFC 2425, September 1998

[12] vCard MIME Directory Profile, IETF, Network Working Group, RFC 2426, September 1998

[13] Specification for Ir Mobile Communication (IrMC) version 1.1 with Errata, Infrared Data Association, October 24th, 2000.

[14] Specification of the Bluetooth System: Covered Core Package version: 1.2 or later

10 ANNEX A: Synopsis of the PBAP functions.

The usage of the various OBEX fields in the PBAP functions is summarized hereunder:

	PullPhoneBook	SetPhoneBook	PullvCardListing	PullvCardEntry
COMMAND	GET	SETPATH	GET	GET
REQUEST				
Connection ID	M	M	M	M
Name	M (phonebook name)	M	M (folder name)	M (object name)
Type	M ("x-bt/phonebook")	-	M ("x-bt/vcard-listing ")	M ("x-bt/vcard")
Application Parameters: Filter	O (all attributes by default)	-	-	O (all attributes by default)
Format	O (vCard 2.1 by default)	-	-	O (vCard 2.1 by default)
MaxListCount	M	-	M	
ListStartOffset	O (0 by default)	-	O (0 by default)	
Order	-	-	O (indexed by default)	
SearchValue	-	-	O	
SearchAttribute	-	-	O ("Name" by default)	
RESPONSE				
Body/EndOfBody	M	-	M	M
Application Parameters: PhonebookSize	C1	-	C1	-
NewMissedCalls	C2	-	C2	-

C1: Mandatory if the value of MaxListCount was 0

C2: Mandatory if the phonebook object is mch; Shall not be used otherwise