GENERIC PIM PROFILE (GPP)

Bluetooth® Profile Specification

- **Date** 2014-Sep-18
- **Revision** V1.0.0
- **Group Prepared By** Telephony and Car Working Group
- **Feedback Email** car-main@bluetooth.org

Abstract:

This Profile Specification is an abstract profile defining a set of generic requirements and functionality used to exchange PIM data objects between devices. It is especially tailored for use cases where a client device (e.g., a car-kit or another terminal device with IO-capabilities) accesses the PIM-object repository of a server device (e.g., a mobile phone or a Smartphone).

Revision History

Revision Number	Date	Comments
V1.0.0	9/18/2014	Adopted by the Bluetooth SIG BoD

Contributors

Name	Company
Veit Kötting (Editor)	Berner & Mattner
Joachim Mertz	Berner & Mattner
Rüdiger Mosig	Berner & Mattner
Yoav Yanai	Berner & Mattner
Burch Seymour	Continental Automotive Systems
Doron Elliott	Ford
Thomas Sibin	Mindtree
Kyle Penri-Williams	Parrot
Scott Walsh	Plantronics
Casper Bonde	Samsung

DISCLAIMER AND COPYRIGHT NOTICE

This disclaimer applies to all draft specifications and final specifications adopted by the Bluetooth SIG Board of Directors (both of which are hereinafter referred to herein as a Bluetooth “Specification”). Your use of this Specification in any way is subject to your compliance with all conditions of such use, and your acceptance of all disclaimers and limitations as to such use, contained in this Specification. Any user of this Specification is advised to seek appropriate legal, engineering or other professional advice regarding the use, interpretation or effect of this Specification on any matters discussed in this Specification.

Use of Bluetooth Specifications and any related intellectual property 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, including, but not limited to, the Membership Application, the Bluetooth Patent/Copyright License Agreement and the Bluetooth Trademark License Agreement (collectively, 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 use of any portion of a Bluetooth Specification may involve the use of intellectual property rights ("IPR"), including pending or issued patents, or copyrights or other rights. Bluetooth SIG has made no search or investigation for such rights and disclaims any undertaking or duty to do so. The legal rights and obligations of each Member are governed by the applicable Membership Agreements, 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 Agreements, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in (i) termination of the applicable Membership Agreements or Early Adopters Agreement and (ii) liability claims by Bluetooth SIG or any of its Members for patent, copyright and/or trademark infringement claims permitted by the applicable agreement or by applicable law.

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 wireless technology ("Bluetooth Products") may be subject to various regulatory controls under the laws and regulations applicable to products using wireless non licensed spectrum 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. To the extent not prohibited by law, in no event will Bluetooth SIG or its Members or their affiliates be liable for any damages, including without limitation, lost revenue, profits, data or programs, or business interruption, or for special, indirect, consequential, incidental or punitive damages, however caused and regardless of the theory of liability, arising out of or related to any furnishing, practicing, modifying, use or the performance or implementation of the contents of this Specification, even if Bluetooth SIG or its Members or their affiliates have been advised of the possibility of such damages. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST BLUETOOTH SIG AND ITS MEMBERS OR THEIR AFFILATES RELATED TO USE OF THE SPECIFICATION.

If this Specification is an intermediate draft, it is for comment only. No products should be designed based on it except solely to verify the prototyping specification at SIG sponsored IOP events and it does not represent any commitment to release or implement any portion of the intermediate draft, which may be withdrawn, modified, or replaced at any time in the adopted Specification.

Bluetooth SIG reserves the right to adopt any changes or alterations to the Specification 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*).

The term *Reserved for Future Use (RFU)* is used to indicate Bluetooth SIG assigned values that are reserved by the Bluetooth SIG and are not otherwise available for use by implementations.

Contents

1 Introduction

1.1 Scope

The Generic PIM Profile (GPP) specifies the generic requirements for protocols and profiles that shall be used by devices for exchange of PIM data objects. It is based on a Client-Server interaction model where the Client initiates the transactions.

This profile defines no specific application objects or data access functions but will be used as a generic framework for PIM profiles (e.g., for access to contact/phonebook data, message objects, social media data or calendar, tasks and notes objects). Therefore, GPP can be considered an abstract profile.

In particular, GPP covers common requirements concerning lower layer protocols and profiles, such as Bluetooth Security, GAP requirements, and Link Manager/Control, as well as common mechanisms and capabilities like general GOEP/OBEX functions (templates), general folder structure, and session initialization procedures.

The PIM application profiles based on GPP inherit the general GPP requirements and will add specific application objects, features, and functions, descriptive parts and further or refined requirements.

1.2 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 capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth qualification program.

1.3 Profile Dependencies

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).

A profile is dependent upon another profile if it re-uses parts of that profile by explicitly referencing it. The Bluetooth profile structure and the dependencies of GPP are depicted in Figure 1.1. A profile has dependencies on lower-level profile(s).

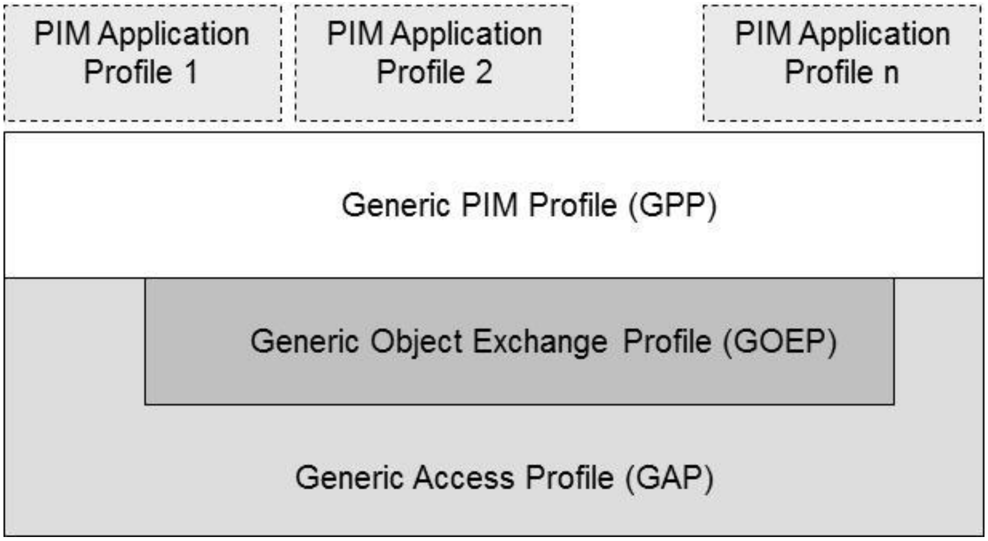
PIM Application Profile 1
PIM Application Profile 2
PIM Application Profile n
Generic PIM Profile (GPP)
Generic Object Exchange Profile (GOEP)
Generic Access Profile (GAP)

Figure 1.1: *Dependencies Bluetooth Profiles*

As indicated in Figure 1.1, the Generic PIM Profile is dependent upon the Generic Object Exchange Profile [3] and the Generic Access Profile [4]. In particular, GPP is based on GOEP v2.0 with OBEX over L2CAP.

The figure also depicts the dependencies of potential PIM application profiles that are based on GPP (e.g., for contacts, messages or calendar data). However, the definition of such profiles is not part of this specification.

The application profiles based on GPP may state additional requirements on the lower layer profiles or may require the usage of additional profiles. For instance, future versions of existing applications may also support OBEX over RFCOMM for the operation with legacy devices.

1.4 Bluetooth Specification Release Compatibility

This profile is compatible with the Bluetooth Core specification version 2.1 +EDR or later.

1.5 Symbols, Conventions, and Definitions

For the feature definition tables used 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.

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

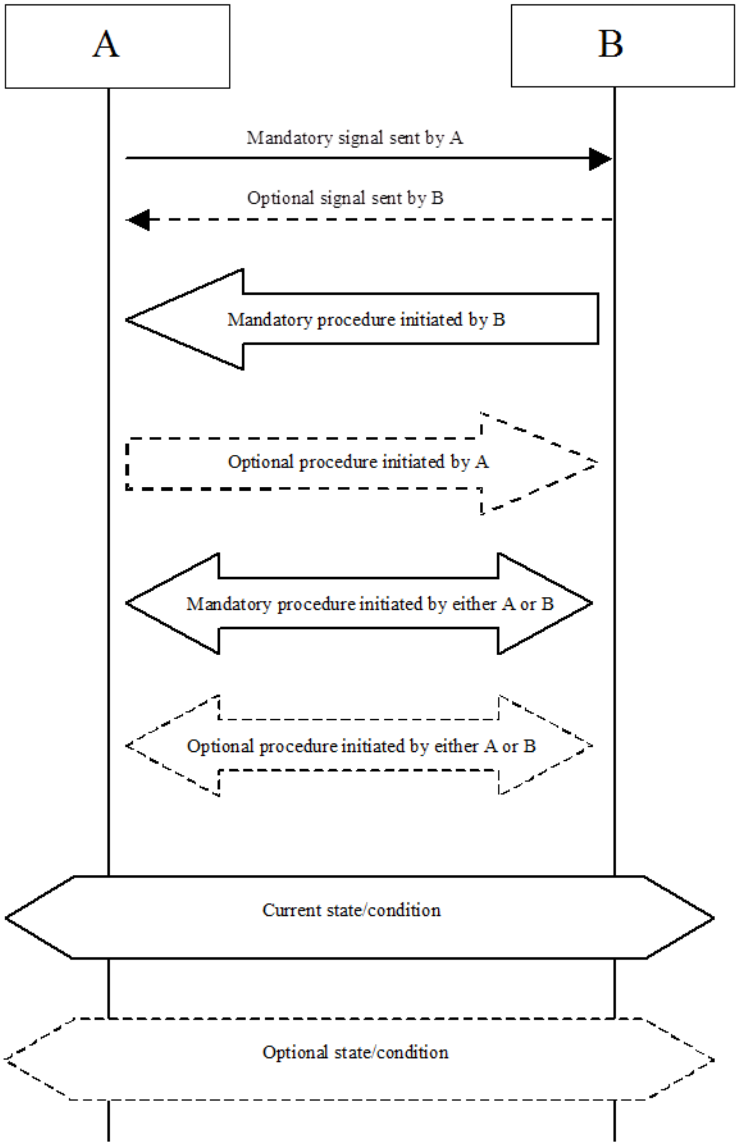
A
B
Mandatory signal sent by A
Optional signal sent by B
Mandatory procedure initiated by B
Optional procedure initiated by A
Mandatory procedure initiated by either A or B
Optional procedure initiated by either A or B
Current state/condition
Optional state/condition

Figure 1.2: *Conventions used in signaling diagrams*

2 Profile Overview

2.1 Protocol Stack

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

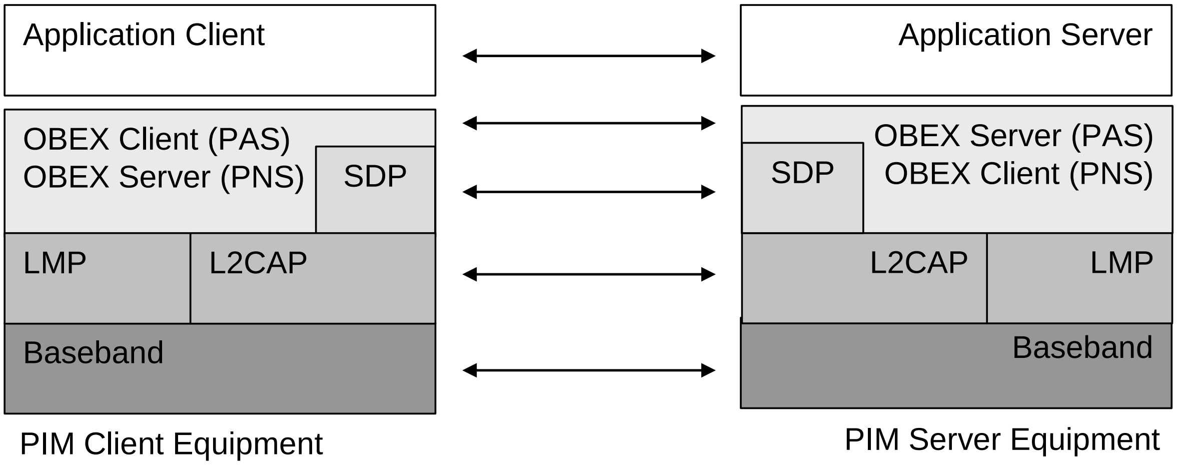

Figure 2.1: *Protocol model for transport of PIM data objects*

PAS = PIM Access Service, PNS = PIM Notification Service, see Section 4.

2.2 Configurations, Roles, and Modes

The following roles are defined for this profile:

- PIM Server Equipment (PIMSE) - is the device that provides the PIM object repository (i.e., has the ability to provide a client unit with PIM objects that are stored in this device and with notifications of changes in its repository). Furthermore, it provides features to upload and modify PIM entries in its repository. For example, a PIM device may be a mobile phone or a smartphone.

- PIM Client Equipment (PIMCE) - is the device that accesses the PIM objects repository engine of the PIMSE for downloading, browsing and displaying existing PIM objects, modifying such objects and also to upload objects to the PIMSE. For example, a PIMCE device may be a car head unit.

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

PIM objects may be any data objects related to personal information management (e.g., messages, contact data, calendar entries, blogs or chat entries).

Figure 2.2 below shows typical configurations of devices for which the PIM Access Profile is applicable. Nevertheless, any Bluetooth device with a PIM object repository can potentially act as a PIMSE and any Bluetooth device with suitable IO-capabilities can act as a PIMCE.

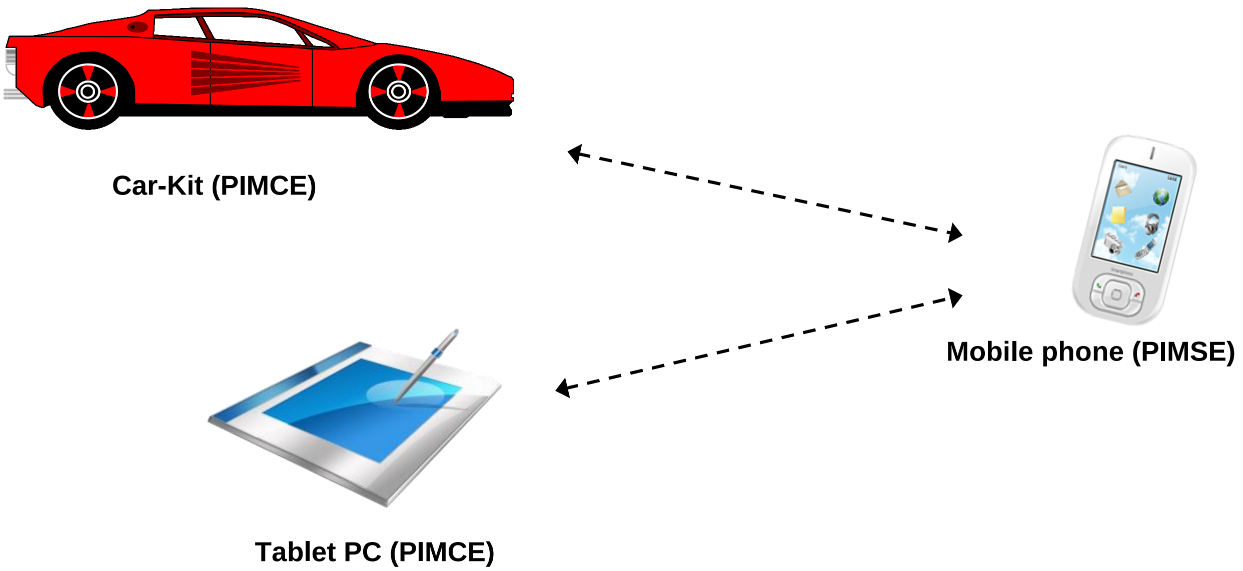

Figure 2.2: *Example of Generic PIM Profile roles*

PIM objects may be received from or sent to an external connection (e.g., a mobile network). The management of such external network connections is not covered by this profile specification. GPP only requires that any PIM object received from or sent to an external network shall be stored in the object repository of the PIMSE.

2.3 User Scenarios

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

- The PIMCE browses in the PIM objects repository of the PIMSE:
  In this scenario, the PIMCE can navigate through the PIMSE’s folder structure, can get listings of PIM objects and can get PIM objects that are locally stored in thePIMSE device. A typical configuration would be that of a Bluetooth car-kit or a PC browsing the content of a mobile phone's PIMSE repository.

- The PIMCE uploads PIM objects onto the PIMSE repository:
  In this scenario, PIM objects are created on the PIMCE device and uploaded to the PIMSE device for storage (e.g., a new calendar entry or an email for sending).

- The PIMCE deletes PIM objects from the PIMSE repository:
  In this scenario, the PIMCE device deletes selected objects on the PIMSE (e.g., removal of a calendar entry or a spam mail).

- The PIMSE notifies the PIMCE:
  In this scenario, the PIMSE notifies the PIMCE about a change in the PIM r (e.g., removal of a calendar entry, a status change of a calendar entry, reception of a new message or modification of a phonebook entry).

3 Application Layer

3.1 Overview

The definition of the specific application layer objects is provided by the PIM application specifications that are based on the Generic PIM Profile. Nevertheless, this specification defines hereunder, templates for some application objects and generic requirements for their use.

GPP considers the following object types that may be transferred:

- Literal objects:
  the actual PIM data objects stored in the repository of the PIMSE (e.g., a message, a phonebook entry, or a calendar entry).

- Listing objects:
  listings of literal objects with a number of listing entries, each with a limited but relevant amount of information about the related literal objects. The listings may be virtual and do not necessarily have to be identical with the contents of the physical PIMSE file system.

- Folder listing objects:
  listings with entries, describing the sub-folders of a folder. The folders structure may be virtual and does not necessarily have to be identical with the structure of the physical PIMSE file system.
  Event report objects:
  events are sent by the PIMSE to the PIMCE to report changes in the PIMSE's object repository.

The general requirements for these objects are described in the following sections. If needed, further object types may be defined by the PIM application profiles.

3.2 GPP Application Object Templates

The definition of the specific object formats used by the PIM application profiles shall be provided by the application profile specifications. For a number of typical objects templates, are given hereunder.

3.2.1 PIM Literal Objects

A PIM application profile shall specify its literal data objects in a text-based notation. The use of XML (eXtensible Markup Language) is recommended and a definition by an XSD (XML Schema Definition) or DTD (Document Type Definition) is required in this case. Alternatively, another textual representation may be chosen. A formal BNF (Backus-Naur form) definition is required in this case.

For both representations, a template is described hereunder:

```
<!-- DTD template for a PIM Literal Object-->

<!DOCTYPE //literal-PIM-object-name// [

<!--application specific elements -->
<!ELEMENT … >
…

<!ATTLIST //literal-PIM-object-name//

    version CDATA #FIXED "x.y">
    handle CDATA #REQUIRED

    <!--application specific attributes -->
    …

>
]>
```

BNF:

```
<literal-PIM-object>::= {
      "BEGIN:'literal-PIM-object-abbreviation'" <CRLF>
          <version-property>
          <handle-property>
          … application specific properties …

      "END:'literal-PIM-object-abbreviation'" <CRLF>
     }
     <version-property>::="VERSION:"
          <common-digit>*"."<common-digit>* <CRLF>
     <handle-property>::="HANDLE:"<hex-digit>* <CRLF>
     …
```

Attributes

- "version" is the version of the literal object as defined by the related application profile specification as a string "x.y" with (e.g., "1.0"). Note that the object version may be different from the version of the related specification.

- "handle" is the literal object handle in hexadecimal representation (see also Section 3.3.2)
- "literal-PIM-object-name" is the literal data object name; it shall be defined by the specific application specification (e.g., "MAP-msg-object")

3.2.2 Listing Object

A PIM application profile shall specify its listing objects by an XML, DTD, or XSD. The template DTD for the listing object is described below.

```
<!DTD for a PIM Listing Object-->

 <!DOCTYPE //listing-name// [

 <!ELEMENT //listing-name// ( //literal-object// )* >
 <!ATTLIST //listing-name// version CDATA #FIXED "x.y">

 <!ELEMENT //literal-object// …>
 <!ATTLIST //literal-object//
     handle CDATA #REQUIRED
     <!--application specific attributes -->

>
]>
```

Attributes

- "version" is the version of the listing object as defined by the related application profile specification as a string "x.y" with (e.g. "1.0"). Note that the object version may be different from the version of the related specification.
- "handle" is the literal object handle in hexadecimal representation (see also 3.3.2)
- "listing-object" is the listing object name; shall be defined by the specific application specification (e.g. "MAP-msg-listing")
- "literal-object" is the literal data object name; shall be defined by the specific application specification (e.g. "msg" for a MAP msg object)

The PIMSE may sort the entries of the listing by any listing entry attributes. The sorting capabilities are defined by the PIM application specification.

3.2.3 Folder-Listing Object

The folder-listing object is defined in 9.1 of [2] and shall be encoded in UTF-8. In the context of the usage for PIM application profiles, folder-listing objects shall not contain literal object entries but shall only contain (sub)folder entries located in the related folder level.

The following figure shows the generic structure of the repository as presented by the PIMSE:

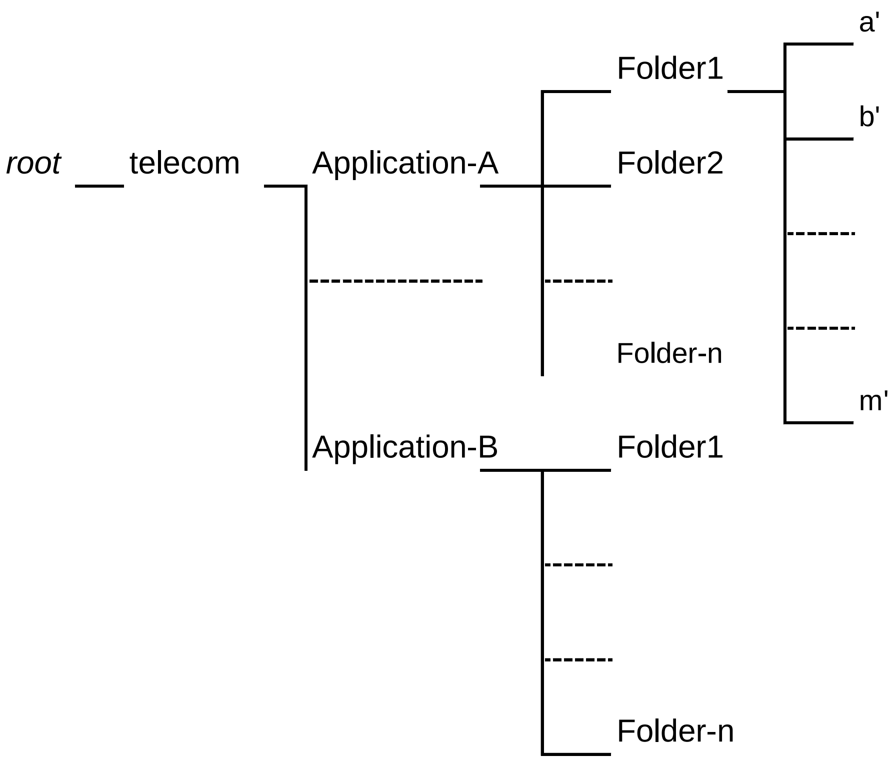

Figure 3.1: BPM virtual folder structure

The first folder layer following the root directory of the repository shall be named as 'telecom'.

The second layer designates the repositories of the particular PIM applications. Each PIM application supported by the PIMSE owns exactly one folder where this folder name is defined by the PIM application specification (e.g., 'msg' for the Message Access Profile). The PIM application may use an arbitrary number of subfolders as defined in the related PIM application profile specification.

3.2.4 Event-Report Object

A PIM application profile shall use an XML based definition of event-report objects. Accordingly, it shall be specified by an XSD or DTD. The event-report object shall be encoded in UTF-8. The event-report object is defined according to the following DTD:

```
<!DTD for the Event-Report object-->

<!DOCTYPE //event-report-name// [

<!ELEMENT //event-report-name// ( event ) >
<!ATTLIST //event-report-name//version CDATA #FIXED "x.y">

<!ELEMENT event EMPTY>
<!ATTLIST event
    type CDATA #REQUIRED
    handle CDATA #IMPLIED
    folder CDATA #IMPLIED
    <!--application specific attributes -->
>
]>
```

Attributes

- "type" is the type of the event, indicating the related change in the PIMSE repository; it shall be defined by the specific application specification (e.g., 'NewCalendarEntry' of the CTN profile)
- "version" is the version of the event-report object as defined by the related application profile specification as a string "x.y" (e.g., "1.0"). Note that the object version may be different from the version of the related specification.
- "handle" is the literal object handle in hexadecimal representation (see also Section 3.3.2)
- "folder" shall be the name of the folder related to the change in the PIMSE's repository, including the path.
- "event-report-name" is the event-report object name; it shall be defined by the specific application specification (e.g., "MAP-event-report")

3.3 GPP General Attributes and Conventions

3.3.1 Character-Set

If not defined otherwise by the specific PIM application profile the character set used for the attributes of the PIM application objects shall be UTF-8.

3.3.2 Handles and Sessions

Handle values:

When exchanging PIM objects, the PIMCE and PIMSE shall use unique identifiers, called handles, to identify individual objects. The PIMSE device shall assign a handle to each PIM object in its repository. The handle shall be a 128 bit unsigned integer whose value is defined by the PIMSE.

A handle shall be represented in capital hexadecimal notation with up to 32 digits, so only numeric 0-9 and upper case A-F characters shall be used; leading zero digits are optional, so a PIMCE shall accept both formats. For example the two strings, "000000000000000000000012345678AB" and"12345678AB", specify an equivalent handle value.

The handle values may be globally unique in the application profiles based on GPP, but this is not guaranteed. As a minimum requirement, handles shall be locally unique within the context of the related service (i.e. for each service represented by a PIMSE SDP record representing).

- a Bluetooth device and
- a Bluetooth application or profile and
- an instance of this application

This means that the handle shall unambiguously identify one and only one PIM object for this device and specific application instance.

Accordingly, a PIMSE device may present objects by identical handle values for different applications. For example, messages from a message application, and calendar events from a separate calendar application on the same PIMSE device, may have the same handle value. Also, different PIMSE devices may use same handle values. For example, messages from the message application on device A and messages on device B may have some identical handle values.

If a PIMCE is connected to multiple PIMSEs, applications, or application instances, the PIMCE is responsible for keeping separate lists of the handles for each PIMSE device and for each application instance provided by these devices.

Handle persistency:

It is recommended that a PIM application based on GPP uses persistent handles(i.e., the handle value of a literal PIM object is not changed during the overall lifetime of the object on the PIMSE device). This means a handle can be reused by the PIMCE after reconnection.

However, as a minimum requirement, a handle shall be valid for the duration of a GPP session. When a GPP session ends (e.g., initiated by the user or caused by a link loss, the handle may no longer valid). Thus, in this case, a PIMCE shall not reuse a handle obtained in a previous GPP session (see also definition of session below).

The kind of persistency shall be stated clearly in the particular application profile specification. If persistency is optional in an application profile, the profile should allow support for it to be advertised in the SDP record's SupportedFeatures attribute.

Sessions:

A GPP session starts with the establishment of a PIM Access Service connection. The term 'GPP session' shall be defined as the time during which at least one PIM Access Service connection or PIM Notification Service connection is ongoing (see also OBEX definitions Section 6.2).

3.3.3 PIMSE Instances

PIMSE devices may present one or several PAS instances to the PIMCE, each providing the complete PIMSE server functionality. For each PAS Instance, there shall be exactly one PAS-Server SDP record. Each PAS instance shall have an ID unique for the corresponding application profile. A PIMCE can access to each PAS instance by a dedicated OBEX connection. The following figure gives an example of a potential configuration:

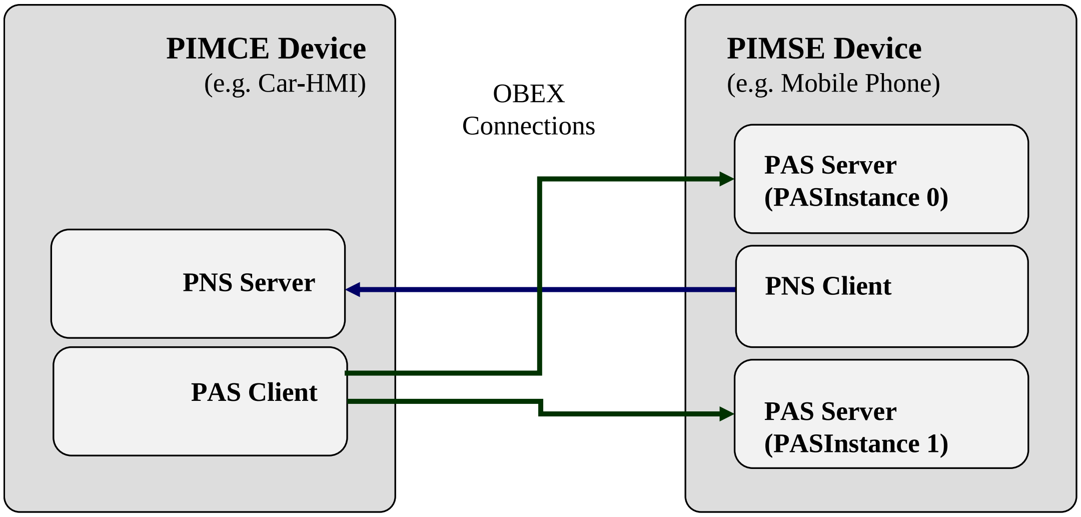

Figure 3.2: *Example for two PAS Instances and its connections*

The following requirements have to be considered:

- PIMSE device

  - Each PIM application profile based on GPP shall provide at least one PAS instance and may provide multiple PAS instances.
  - The SDP record attribute 'ServiceName' (see Section 7.1) should enable an easy recognition of the related repository on the PIMSE device by the user. Nevertheless, as the SDP records may be accessible without pairing, the 'ServiceName' should not expose any personal information (e.g., a complete phone number or email address, with respect to privacy protection).
  - The PIMSE device shall enable simultaneous and independent OBEX connections to all provided PAS instances.
  - If notification is supported by an application profile based on GPP, there shall be only one PNS connection to handle all notification of this application profile, independent from the number of instances of this profile.
- PIMCE device
  - A PIMCE may connect via OBEX to one, several, or all PAS Instances offered by a PIMSE device.

4 Generic PIM Profile Features

4.1 Overview

As described in Section 2.1, GPP is based on OBEX. For further information about the OBEX/GOEP requirements see also Section 6.2. Within the scope of GPP, the following OBEX services are defined:

- The PIM Access Service (PAS) is an OBEX service by which the PIMCE acts as an OBEX Client and connects to a PIMSE that acts as OBEX Server.
- The PIM Notification Service (PNS) is an OBEX service by which the PIMSE acts as OBEX Client and connects to the PIMCE that acts as an OBEX Server.

GPP includes the features in the table below. For a device to comply with this specification, it shall observe the following implementation requirements table:

Feature	Support by the PIMCE	Support by the PIMSE
Connect PIM Access Service	M*	M**
Disconnect PIM Access Service	M*	M**
Connect PIM Notification Service	O**	M*
Disconnect PIM Notification Service	C1**	M*

Table 4.1: *GPP features*

* ability to request or initiate, ** ability to respond or react
C1: Feature is mandatory if feature 'Connect PIM Notification Service' is supported by device

4.2 Initialization Sequences

The OBEX connection setup in GPP shall always be triggered by the PIMCE device. However, the connection initialization sequence depends on the features used during a GPP session:

The initialization sequence that applies to applications that use only the PIMSE's PIM Access Service is described in Section 4.2.1.

For applications that use both the PIM Access Service of the PIMSE and the PIM Notification Service of the PIMCE, the initialization sequence that applies is described in Section 4.2.2.

4.2.1 Initialization Sequence for a GPP Session That Uses Only the PIM Access Service

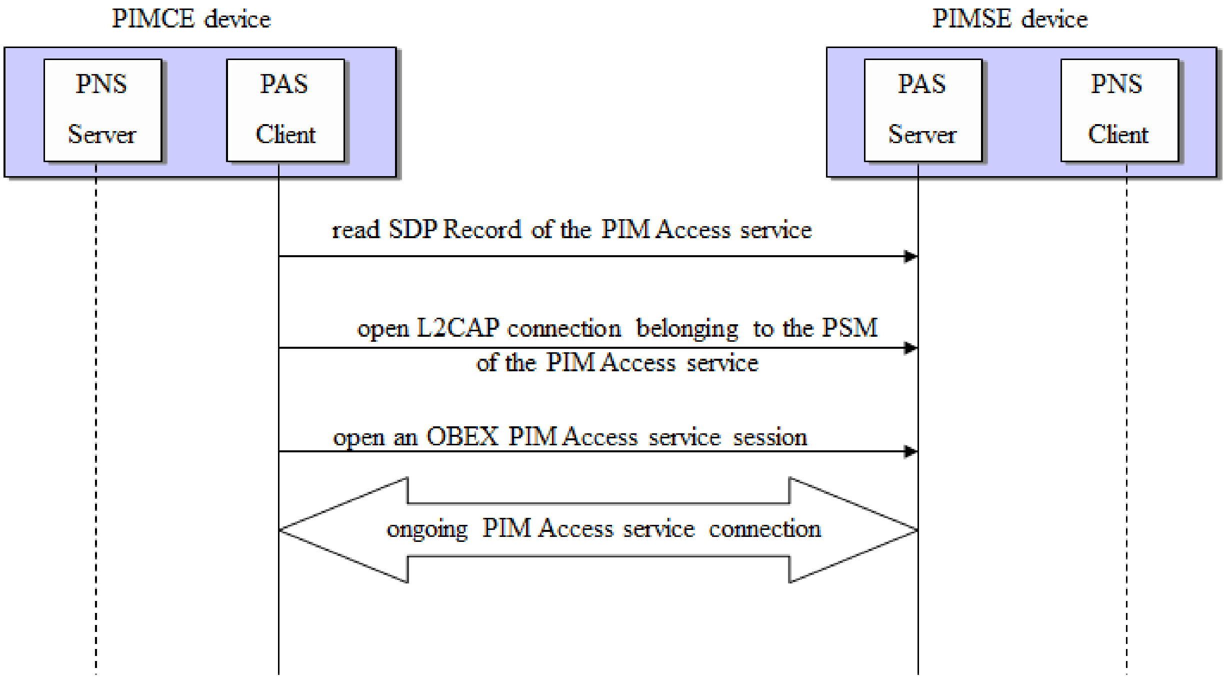
PIMCE device
PNS Server
PAS Client
PIMSE device
PAS Server
PNS Client
read SDP Record of the PIM Access service
open L2CAP connection belonging to the PSM of the PIM Access service
open an OBEX PIM Access service session
ongoing PIM Access service connection

Figure 4.1: *Establishment of GPP session (usage of PIM Access Service only)*

The establishment of a PIM Access Service connection is done in accordance with [4], where PIMCE acts as client and the PIMSE acts as server.

The UUID value to be used for the OBEX Target header shall be defined in the application profile specification.

4.2.2 Initialization Sequence for a GPP Session That Uses Both the PIM Access Service and the PIM Notification Service

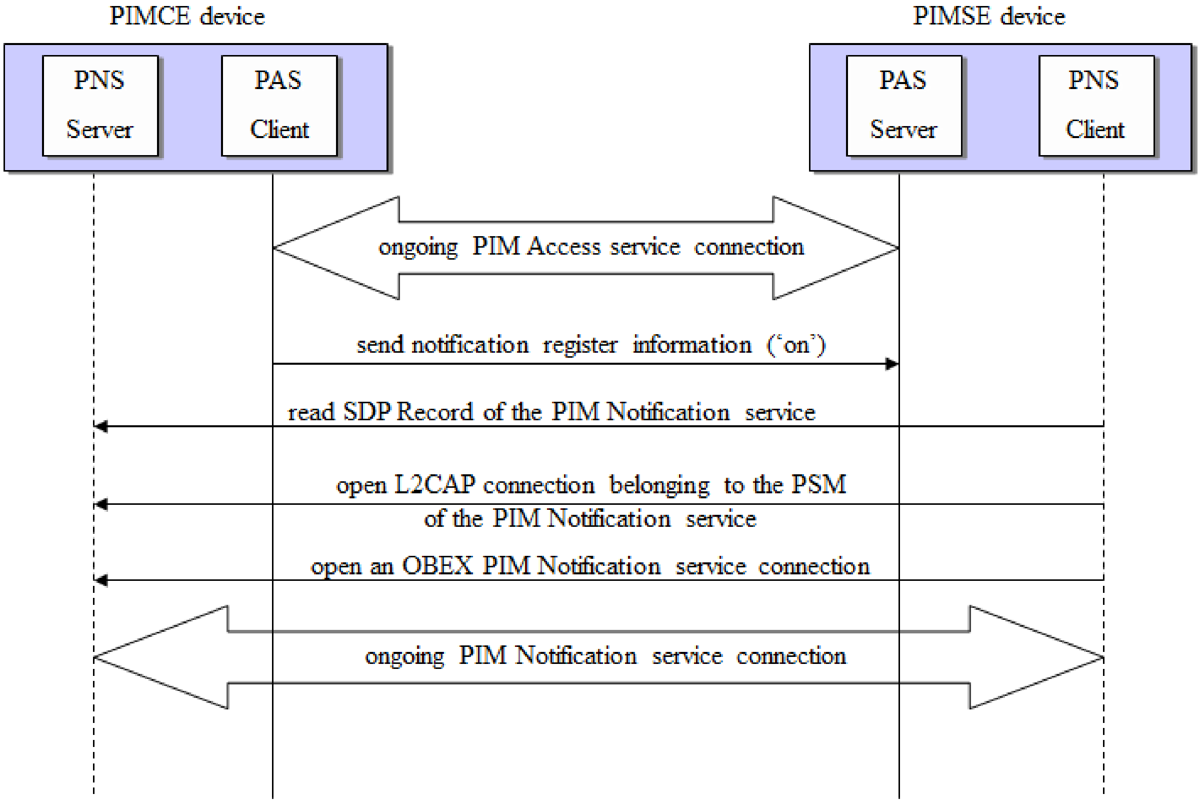
PIMCE device
PNS Server
PAS Client
PIMSE device
PAS Server
PNS Client
ongoing PIM Access service connection
send notification register information ('on')
read SDP Record of the PIM Notification service
open L2CAP connection belonging to the PSM of the PIM Notification service
open an OBEX PIM Notification service connection
ongoing PIM Notification service connection

Figure 4.2: *Establishment of GPP session (usage of PIM Access Service and PIM Notification Service)*

Each application profile session between two devices shall have at most one PIM Notification Service connection which shall be used for the notifications of all PAS connections for this application. Application profiles shall not share a PIM Notification Service connection with other application profiles, see figure with example below.

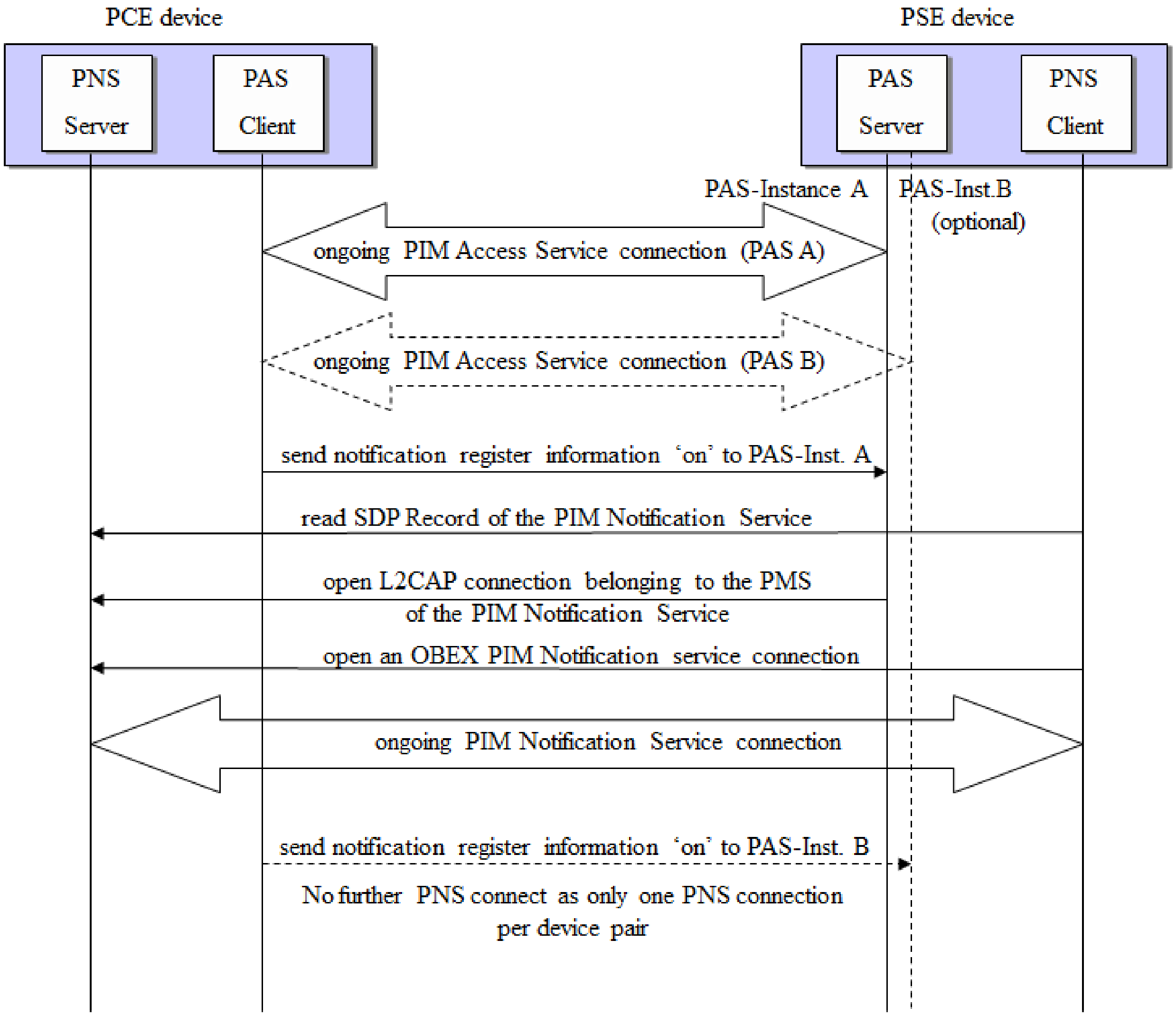
PCE device
PSE device
PNS Server
PAS Client
PAS Server
PNS Client
PAS-Instance A
PAS-Inst.B (optional)
ongoing PIM Access Service connection (PAS A)
ongoing PIM Access Service connection (PAS B)
send notification register information 'on' to PAS-Inst. A
read SDP Record of the PIM Notification Service
open L2CAP connection belonging to the PMS of the PIM Notification Service
open an OBEX PIM Notification service connection
ongoing PIM Notification Service connection
send notification register information 'on' to PAS-Inst. B
No further PNS connect as only one PNS connection per device pair

Figure 4.3: *Example of establishment of a GPP session with several PAS connections*

The establishment of a PIM Notification Service connection is done in accordance with [3] with the PIMCE as OBEX Server and the PIMSE as OBEX Client.

The UUID value to be used for the OBEX Target header shall be defined in the application profile specification.

The establishment of a PIM Notification connection requires the previous establishment of a PIM Access Service connection as described in Section 4.2.1.

4.2.3 Terminating a PIM Access or PIM Notification Service Connection

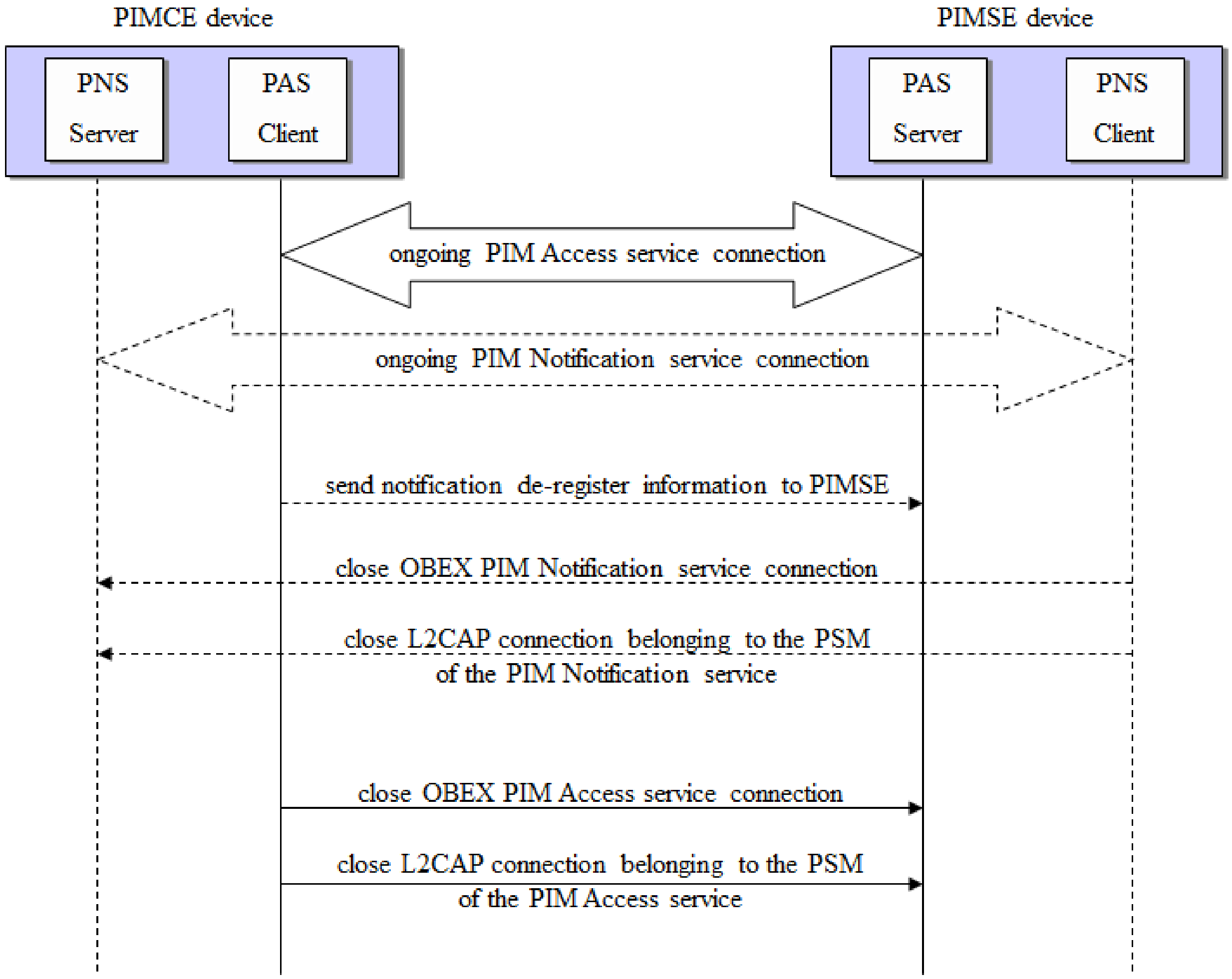
PIMCE device
PIMSE device
PNS Server
PAS Client
PAS Server
PNS Client
ongoing PIM Access service connection
ongoing PIM Notification service connection
send notification de-register information to PIMSE
close OBEX PIM Notification service connection
close L2CAP connection belonging to the PSM of the PIM Notification service
close OBEX PIM Access service connection
close L2CAP connection belonging to the PSM of the PIM Access service

Figure 4.4: *Termination of a GPP session (usage of PIM Access and optional PIM Notification Service)*

The termination of a PIM Access or a PIM Notification Service connection is done in accordance with [2]. The PNS connection of a given GPP-based application shall be closed if :

- all registered PAS connections have been de-registered

  or

- if all the application's PAS connections have been closed

5 Generic PIM Profile Functions

5.1 Overview

The definitions of the specific functions are up to the applications based on GPP.

However, general requirements are stated here as well as templates for some typical functions. In particular, this includes OBEX fields and headers that are common for all application profiles based on GPP.

Note that only OBEX headers are listed for the GPP function templates hereunder that are determined by the profiles based on GPP. General OBEX headers which are just part of an OBEX protocol without any further requirements by GPP (e.g., PacketLength) are not listed but have to be present in the specific function requests/responses according to [3].

5.2 SendEvent Function

This function is used by the PIMSE to send a notification event to the PIMCE in case of a change in the PIMSE's object repository(i.e., an addition, a removal or a shift of an object in the repository or a modification of an object).

Each change within the repository shall result in a sending of a single event by the PIMSE device.

The common fields and headers of the function are listed below. The application profile may add further headers.

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	Opcode	PUT (0x02 or 0x82)	M
Field	Packet Length	Varies	M
Header	Connection ID	Varies	M
Header	Single Response Mode	0x01	M
Header	Type	Defined by Application	M
Header	Application Parameters		
	- InstanceID	Varies	M
Header	Body/End of Body	Event Report Object of the application	M

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	Response Code	0x90 or 0xA0 or Error Code	M

Field	Packet Length	Varies	M
Header	Single Response Mode	0x01	C1
Header	Single Response Mode Param	0x01	C2

C1: The Single Response Mode header is mandatory (M) if:

- an SRM header has been received in the previous PUT request AND

- the response code is success (0x90 or 0xA0).

otherwise excluded (X).

C2: The Single Response Mode Param header is optional (O) if Single Response Mode is used, otherwise excluded (X).

5.2.1 Connection ID

The connection ID header shall be used to indicate the connection ID, received during the connection establishment, in order to signal the recipient of the request which OBEX connection this request belongs to.

5.2.2 Type

The type header shall be used by the related application profile to indicate the type of object to be transmitted. Accordingly, the value has to be defined by the application profile.

5.2.3 Application parameters

For further details see Section 6.2.5.

5.2.3.1 InstanceID

This header shall be used by the PIMSE to indicate the corresponding PIMSE-Instance (see Section 3.3.3). As only one PNS service connection per application can be established from the PIMSE device to the MCE, this parameter is required by the PIMCE to determine the PIMSE Instance that should receive this event. The PIMCE can retrieve the corresponding InstanceID from the PIMSE's SDP record (see Section 7.1.1 ‘InstanceID’ parameter).

5.2.4 Body/EndOfBody

This header shall contain the event-report object that is sent by the PIMSE device.

5.3 SetNotificationRegistration Function

This function is used by the PIMCE device to register for notifications of changes in the PIMSE's object repository for a specific application.

The common fields and headers of the function are listed below. The application profile may add further headers.

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	1. Opcode	2. PUT (0x02 or 0x82)	3. M
4. Field	5. Packet Length	6. Varies	7. M
8. Header	9. Connection ID	10. Varies	11. M
12. Header	Type	Defined by Application	13. M
14. Header	15. Application Parameters		
	- NotificationStatus	On/Off	M
16. Header	17. Body/End of Body	18. Filler-byte 0x30	19. M

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	20. Response Code	21. 0x90 or 0xA0 or Error Code	22. M
23. Field	24. Packet Length	25. Varies	26. M

5.3.1 Connection ID

See Section 5.2.1.

5.3.2 Type

The type header shall be used by the related application profile to indicate the type of object to be transmitted. Accordingly, the value has to be defined by the application profile.

5.3.3 Application Parameters

For further details see Section 6.2.5.

5.3.3.1 NotificationStatus

The PIMCE shall indicate the request for being notified about changes in the object repository for the corresponding application profile. The header shall have either of the values:

- "Off", meaning no notification required or
- "On", meaning the notification service (PNS) session of the corresponding application shall be established

5.3.4 Body/EndOfBody

To avoid a PUT with an empty body leading to a 'delete' these headers contain a filler byte. The value of this byte shall be 0x30 (="0").

5.4 GetObjectListing Function

This function is used by the PIMCE to retrieve a literal object listing from the PIMSE.

The common fields and headers of the function are listed below. The application profile may add further headers. In particular this function may require filtering of the objects.

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	27. Opcode	28. GET (0x03 or 0x83)	29. M
30. Field	31. Packet Length	32. Varies	33. M
34. Header	35. Connection ID	36. Varies	37. M
38. Header	39. Single Response Mode	40. 0x01	41. M
42. Header	43. Single Response Mode Param	44. 0x01	45. O
46. Header	47. Type	48. Defined by Application	49. M
50. Header	51. Name	52. Name of the Folder	53. M
54. Header	55. Application Parameters		
	- MaxListCount	Varies	O
	- ListStartOffset	Varies	O

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	56. Response Code	57. 0x90 or 0xA0 or Error Code	58. M
59. Field	Packet Length	60. Varies	61. M
62. Header	63. Single Response Mode	64. 0x01	65. C1
66. Header	67. Single Response Mode Param	68. 0x01	69. C2
70. Header	71. Application Parameters		
	- ListingSize	Varies	C3
72. Header	73. Body/End of Body	74. Object Listing object	75. C3

C1: The Single Response Mode header is mandatory (M) if:
- an SRM header has been received in the previous GET request AND
- the response code is success (0x90 or 0xA0).
otherwise excluded (X).

C2: The Single Response Mode Param header is optional if Single Response Mode is used, otherwise excluded (X).

C3: These parameters shall be present if the request has been successful

5.4.1 Connection ID

See Section 5.2.1.

5.4.2 Name

This property shall be used to indicate the folder from which the Object-Listing object is to be retrieved.

5.4.3 Type

The type header shall be used by the related application profile to indicate the type of object to be transmitted. Accordingly, the value has to be defined by the application profile.

5.4.4 Application Parameters

For further details see Section 6.2.5.

5.4.4.1 MaxListCount

This header may be used to indicate the maximum number of literal objects listed in the Object-Listing object. The number of objects in the list shall not exceed the limit specified by this parameter. The maximum number of entries shall be 1024 if this header is not specified. The Object-Listing object shall contain MaxListCount entries if enough entries exist, starting from ListStartOffset. Lists are zero-origined(i.e., the first entry in a list is entry zero). For example, a request with 'ListStartOffset' = 0 and 'MaxListCount'=100 delivers the first 100 literal objects. The listing order of the objects is application specific. If 'MaxListCount'=0 is specified in the request, the PIMSE shall respond with the header "MessagesListingSize" only and shall not deliver an Object-Listing object.

5.4.4.2 ListStartOffset

This header may be used to indicate the offset of the first entry of the returned Object-Listing object. For example, if ListStartOffset is 5, then the first 5 objects are not delivered. The default offset shall be 0 if this header is not specified.

5.4.4.3 ListingSize

If the request has been successful, this application parameter shall be used in the response to report the number of accessible literal objects in the corresponding folder that are selected by the filter parameters as described above. If MaxListCount = 0, the PIMSE shall ignore the request-parameter "ListStartOffset". In this case the response shall not contain the Body header.

5.4.5 Body/EndOfBody

If the request has been successful, these headers shall contain the Object-Listing object that is returned by the PIMSE device. If MaxListCount = 0 in the request this parameter shall not be present in the response.

5.5 GetObject function

This function is used by the PIMCE to download a literal data object from the PIMSE's repository.

The common fields and headers of the function are listed below. The application profile may add further headers.

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	76. Opcode	77. GET (0x03 or 0x83)	78. M
79. Field	80. Packet Length	81. Varies	82. M
83. Header	84. Connection ID	85. Varies	86. M
87. Header	88. Single Response Mode	89. 0x01	90. M
91. Header	92. Single Response Mode Param	93. 0x01	94. O
95. Header	96. Type	97. Defined by Application	98. M
99. Header	100. Name	101. Object handle	102. M

The response is formatted as follows

Field/Header	Name	Value	Status
Field	103. Response Code	104. 0x90 or 0xA0 or Error Code	M
105. Field	106. Packet Length	107. Varies	108. M
109. Header	110. Single Response Mode	111. 0x01	112. C1
113. Header	114. Single Response Mode Param	115. 0x01	116. C2
117. Header	118. Body/End of Body	119. Literal object of application	120. C3

C1: The Single Response Mode header is mandatory (M) if:

- an SRM header has been received in the previous GET request AND
- the response code is success (0x90 or 0xA0).

otherwise excluded (X).

C2: The Single Response Mode Param header is optional if Single Response Mode is used, otherwise excluded (X).

C3: These parameters shall be present if the request has been successful

5.5.1 Connection ID

See Section 5.2.1.

5.5.2 Name

The Name header shall be used to indicate the handle of the literal object to be retrieved. The handle shall be represented by a null-terminated Unicode text string with up to 32 hexadecimal digits (the handle is 128 bits but leading zeros may be omitted).

5.5.3 Type

The type header shall be used by the related application profile to indicate the type of object to be transmitted. Accordingly, the value has to be defined by the application profile.

5.5.4 Application Parameters

No common application parameters are defined by GPP.

5.5.5 Body/EndOfBody

If the request has been successful, this header shall contain the body of the literal object that is returned by the PIMSE device.

5.6 PushObject Function

This function is used by the PIMCE to push or upload a literal data object to the PIMSE's repository.

The common fields and headers of the function are listed below. The application profile may add further headers. In particular this function may require filtering of the objects.

Field/Header	Name	Value	Status
Field	121. Opcode	122. PUT (0x02 or 0x82)	123. M
124. Field	125. Packet Length	126. Varies	127. M
128. Header	129. Connection ID	130. Varies	131. M
132. Header	133. Single Response Mode	134. 0x01	135. M
136. Header	137. Type	138. Defined by Application	139. M
140. Header	141. Name	142. Name of the Folder	143. M
144. Header	145. Body/End of Body	146. Application object	147. M

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	148. Response Code	149. 0x90 or 0xA0 or Error Code	150. M
151. Field	152. Packet Length	153. Varies	154. M
155. Header	156. Single Response Mode	157. 0x01	158. C1
159. Header	160. Single Response Mode Param	161. 0x01	162. C2
163. Header	164. Name	165. Handle of the Object	166. C3

C1: The Single Response Mode header is mandatory (M) if:

- an SRM header has been received in the previous PUT request AND

- the response code is success (0x90 or 0xA0).

otherwise excluded (X).

C2: The Single Response Mode Param header is optional if Single Response Mode is used, otherwise excluded (X).

C3: These parameter shall be present if the request has been successful.

5.6.1 Connection ID

See Section 5.2.1.

5.6.2 Name

5.6.2.1 In Request

This property shall be used to indicate the folder to which the literal object is to be pushed. The property shall be empty in case the desired listing is that of the current folder or shall be the name of a child folder otherwise. Thus, the value shall not include any path information.

5.6.2.2 In Response

The Name header shall be used to contain the handle that was assigned by the PIMSE device to the literal object that was pushed by the PIMCE device. The handle shall be represented by a null-terminated Unicode text string with up to 32 hexadecimal digits.

5.6.3 Type

The type header shall be used by the related application profile to indicate the type of object to be transmitted. Accordingly, the value has to be defined by the application profile.

5.6.4 Application Parameters

No common application parameters are defined by GPP.

5.6.5 Body/EndOfBody

These headers shall contain the body of the literal object that is delivered to the PIMSE device.

5.7 GetInstanceInformation Function

This function may be used by the PIMCE to retrieve information about the application instances provided by the PIMSE (e.g., a user-readable name). The request is formatted as follows:

The common fields and headers of the function are listed below. The application profile may add further headers.

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	167. Opcode	168. GET (0x03 or 0x83)	169. M
170. Field	171. Packet Length	172. Varies	173. M
174. Header	175. Connection ID	176. Varies	177. M
178. Header	179. Single Response Mode	180. 0x01	181. M

182. Header	183. Single Response Mode Param	184. 0x01	185. O
186. Header	187. Type	188. Defined by Application	189. M
190. Header	191. Application Parameters - InstanceID	Varies	M

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	192. Response Code	193. 0x90 or 0xA0 or Error Code	194. M
195. Field	196. Packet Length	197. Varies	198. M
199. Header	200. Single Response Mode	201. 0x01	202. C1
203. Header	Single Response Mode Param	204. 0x01	205. C2
206. Header	207. Body/End of Body	208. Instance description of application	209. C3

C1: The Single Response Mode header is mandatory (M) if:
- an SRM header has been received in the previous GET request AND
- the response code is success (0x90 or 0xA0)
otherwise excluded (X).

C2: The Single Response Mode Param header is optional if Single Response Mode is used, otherwise excluded (X).

C3: These parameter shall be present if the request has been successful.

5.7.1 Connection ID

See Section 5.2.1.

5.7.2 Type

The type header is defined by the application that uses the function.

5.7.3 Application Parameters

5.7.3.1 Instance ID

The identifier of the PIMSE-instance for which the information is requested.

5.7.4 Body/EndOfBody

The Body includes a string with the requested user-readable information of the application instance. It shall be represented by a null-terminated UTF-8 text string of at most 200 characters (including the null termination character).

5.8 SyncInstance Function

This function allows the PIMCE to initiate a synchronization of the corresponding application instance with an external server. For example, this may be an email server or a contact synchronization.

The request is formatted as follows:

Field/Header	Name	Value	Status
Field	Opcode	PUT (0x02 or 0x82)	M
Field	Packet Length	Varies	M
Header	Connection ID	Varies	M
Header	Type	Defined by Application	M
Header	EndOfBody	Filler-byte "0x30"	M

The response is formatted as follows:

Field/Header	Name	Value	Status
Field	210. Response Code	211. 0x90 or 0xA0 or Error Code	212. M
213. Field	214. Packet Length	215. Varies	216. M

If the PIMSE does not allow an external sync of the related instance, it shall answer with a 'Not implemented' error response. If the sync is only temporary not available, (e.g., temporary loss of the network connection) it shall answer with a ‘Service Unavailable’ error response.

5.8.1 Connection ID

See Section 5.2.1.

5.8.2 Type

The type header is defined by the application that uses the function.

5.8.3 Application Parameters

No common application parameters are defined by GPP.

5.8.4 EndOfBody

To avoid PUT with empty Body leading to a 'delete' this header shall contain a filler byte. The value of this byte shall be 0x30.

6 Profile Dependencies

6.1 Profile F

6.1.1 Session initialization

The PIMCE device shall use the services of the PIMSE device only after successfully creating a secure connection. This includes exchanging of security initialization messages, bonding, and enabling encryption.

Either the PIMSE or PIMCE may initiate bonding. At a minimum, the PIMSE shall support Inquiry and Paging and the PIMCE shall support Inquiry Scan and Page Scan in order to initiate bonding. Either device can initiate the link establishment.

Only the PIMCE can start a GPP session on OBEX level (OBEX services PAS and PNS, see also Section 4.1).

6.1.2 Bluetooth Security

The two devices shall create a secure connection using the GAP authentication procedure as described in the Generic Access Profile [4]. This procedure shall include the Secure Simple Pairing procedures and will include creation of link keys.

The Generic PIM Profile mandates the use of several Bluetooth security features:

> **Bonding:** The PIMCE and PIMSE shall be bonded before setting up a PIM Access Profile connection. Security Mode 4 shall be used. While any of the four association models of Secure Simple Pairing may be used [1], it is recommended to use authenticated link keys (i.e., using one of the association models 'Numeric Comparison', 'Out Of Band', or 'Passkey Entry').
>
> **Encryption:** The link between PIMCE and PIMSE shall be encrypted.

Furthermore, the following issues are mandated for devices complying with the Generic PIM Profile:

> **Link keys:** Combination keys shall be used for Generic PIM Profile connections.
>
> **Encryption key length:** The length of the encryption key should be at least 64 bits. For increased security, use of the maximum length allowed given regional regulation is encouraged.

6.2 OBEX

6.2.1 OBEX Services

This profile is based on GOEPv2 [3] and makes use of the IrOBEX specification [2]. Within the scope of GPP, two OBEX services are defined: The PIM Access service (PAS) and the PIM Notification Service (PNS), see also Section 4.1.

The GPP features are using the above defined OBEX services in the following way:

Feature	OBEX Service
Connect PIM Access Service	PIM Access Service
Disconnect PIM Access Service	PIM Access Service
Connect PIM Notification Service	PIM Notification Service
Disconnect PIM Notification Service	PIM Notification Service

***Table 6.1:** OBEX Services*

6.2.2 OBEX Operations

The OBEX operations used, as well as their mandatory or optional nature, depend on the application profile and shall be defined in the related specification.

6.2.3 OBEX Headers

The list of OBEX headers used, as well as their mandatory or optional nature depend on the application profile shall be defined in the related specification.

6.2.4 Service Target UUIDs

The OBEX Target header UUID values for the OBEX services PAS and PNS shall be defined by the particular application profile specifications.

6.2.5 Application Parameter Headers

The tag IDs used in the Application Parameters header are listed below.

Value	Tag ID	Length	Possible Values
MaxListCount	0x41	2 bytes	0x0000 to 0xFFFF
ListStartOffset	0x42	2 bytes	0x0000 to 0xFFFF

NotificationStatus	0x43	1 byte	Bit mask (*):
			xxxxxxx0 = “Off”
			xxxxxxx1 = “On”

InstanceID	0x44	1 byte	0…255
ListingSize	0x46	2 bytes	0x0000 to 0xFFFF

***Table 6.2:** Tag IDs*

(*) the bits marked by 'x" shall be set to zero; all other values are reserved for future use

All of the Application Parameter header values use big-endian byte ordering.

6.2.6 OBEX Headers in Multiple-Packet Responses

In the case of multi-packet responses, there is a need to specify which packet contains the headers to be returned to the PIMCE or to the PIMSE. Although the IrOBEX specification does not impose any restrictions in this area, the following rule shall be used in the Generic PIM Profile to encourage interoperability:

In the case of a multi-packet PUT response (i.e., the object being transported is large enough to require several packets), the headers, except SRM and SRMP, shall be placed in the last packet, or in the last packets if one packet is not sufficient for the headers. All intermediate response packets shall contain only the Continue response code or, when necessary, one of the error codes. Note that if the OBEX Partial Content response code is used, it shall be issued in the last response packet, after the object has been completely received.

In the case of multi-packet Get response (i.e., the object being transported is large enough to require several packets), all the headers other than the body header shall be placed in the first packet. If the first packet has enough room to include a portion of the object body, then the first packet shall end with the body header that carries this portion of object. Otherwise, the object shall be transferred in subsequent packets.

6.2.7 OBEX Error Codes

The only mandatory error codes for the PIMSE are:

- Bad Request: indicates that the request could not be correctly interpreted or handled.
- Not implemented: indicates that the requested function is not supported.

However, the PIMSE can use the following error codes to provide to the PIMCE a more detailed error report:

Error Code	Client (interprets the Error Codes)	Server (informs of Errors)	Meaning in the PIM Access Profile
Bad Request	M*	M	Function not recognized of ill-formatted.
Not implemented	M*	M	Function recognized but not supported.
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 and well formatted, but that the object to be handled is protected and access is not authorized (either temporarily or permanently).
Precondition Failed	M*	O	The function was recognized and well formatted but there is a problem with one of the request's parameter values.

Not Found	M*	O	The function was recognized and well formatted and all the parameters are proper, but a requested object could not be found.
Not Acceptable	M*	O	The request is recognized and well formatted 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*	O	The function was recognized and well formatted and is normally executable, but a system condition prevents it from being performed.
Forbidden	M*	O	Function recognized and correctly formatted but refused by server, for any reason.

Table 6.3: *Error Codes*

* Indicates that the Client shall recognize this response code as an error code.

On the PIMCE side, the entire response code list above shall be recognized as error codes; how to handle these error codes is left to the implementer's discretion. Support for response codes indicated above as optional is recommended, because they are more informative and provide the PIMCE with a better indication of the nature of an error; this permits better error reporting.

When multi-packet responses are used, response codes should 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.

6.2.8 OBEX Authentication

OBEX authentication shall not be used within GPP and profiles based on it to avoid redundancy with the authentication on link level. However, the application profiles may add this feature if and only if support for legacy devices is required (i.e., GOEP v1.0 implementations).

6.3 Link Control (LC) Interoperability Requirements

In the table below, changes to the support status as listed in the Serial Port Profile [1]. The specification allows both PIMCE and PIMSE to support Inquiry and Inquiry scan. It is mandatory for PIMSE to support Inquiry and for the PIMCE to support Inquiry scan. It is optional for the PIMSE to support Inquiry Scan and for the PIMCE to support Inquiry though this support is strongly recommended.

	Capability	Support in PIMCE	Support in PIMSE

1	Inquiry	O	M
2	Inquiry scan	M	O

Table 6.4: *LC capabilities*

6.3.1 Class of Device/Service Field

This specification places no requirement on the Class of Device/Service Field. It is expected that many types of devices will implement GPP.

6.4 Generic Access Profile

This profile requires compliance to the Generic Access Profile [4].

This section defines the support requirements for the capabilities as defined in the Generic Access Profile [4]. All “Mandatory”, “Optional”, “Excluded” and “Conditional” properties are inherited from GAP. This section lists only those properties that differ from the requirements in GAP.

6.4.1 Modes

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

	Procedure	Support in PIMCE	Support in PIMSE
	Discoverability modes		
1	General discoverable mode	C.1	O
2	Limited discoverable mode	C.1	O
	Pairing modes		
3	Pairable mode	M	M

Table 6.5: *GAP Modes*

C.1: At least one of these modes shall be supported

6.4.2 Security Aspects

There is no change to the requirements as stated in the General Access Profile [4].

6.4.3 Idle Mode Procedures

The following table shows the differences to in the support status for the GAP Idle mode procedures within this profile:

	Procedure	Support in PIMCE	Support in PIMSE
1	General inquiry	O	M
2	Limited Inquire	O	O

Table 6.6: *Idle mode procedures*

7 Service Discovery

7.1 SDP Interoperability Requirements

The following service records are defined for the Generic PIM Profile. There shall be one service record per PIMCE device (PIM Notification Service, server role) and on the PIMSE device one for each PAS instance (PIM Access Service, server role, see also Section 3.3.3).

If a PIMCE requires connection to several PAS Instances it shall establish separate PAS connections to each of these PAS Instances so it shall use dedicated OBEX channels for each PAS connection. The PIMCE can differentiate the particular PAS Instances by their SDP Service record attributes ServiceName and InstanceID. Both values shall be unique for all SDP records of a PIMSE device. The value range of the InstanceID shall be 0..255.

7.1.1 SDP Record Template for the PIM Access Service on the PIMSE Device

The Generic PIM Profile (GPP) does not own SDP records. The following table is a template for the application profiles to use if they have a PAS.

Item	Definition	Type	Value	Status	Default
ServiceClassID List				M	
ServiceClass #0		UUID	Defined by the application profile	M	
Protocol Descriptor List				M	
Protocol #0		UUID	L2CAP	M	
Protocol #1		UUID	OBEX	M	
ServiceName	Displayable text name	String	Defined by the application profile	O	
Bluetooth Profile Descriptor List				M	
Profile #0	Supported Profiles	UUID	Defined by the application profile	M	
Param #0	Profile Version	Uint16	Defined by the application profile	M	
InstanceID		Uint8		O	
SupportedFeatures		Uint16	Defined by the application profile	O	

Profiles based on this specification may customize this template by adding new optional or mandatory items to these records.

7.1.2 SDP Record Template for the PIM Notification Service on the PIMCE Device

The Generic PIM Profile (GPP) does not own SDP records. The following table is a template for the application profiles to use if they have a PNS.

Item	Definition	Type	Value	Status	Default
ServiceClassID List				M	
ServiceClass #0		UUID	Defined by the application profile	M	
Protocol Descriptor List				M	
Protocol #0		UUID	L2CAP	M	
Protocol #1		UUID	OBEX	M	
ServiceName	Displayable text name	String	Defined by the application profile	O	
Bluetooth Profile Descriptor List				M	
Profile #0	Supported profiles	UUID	Defined by the application profile	M	
Param #0	Profile version	Uint16	Defined by the application profile	M	

Profiles based on this specification may customize this template by adding new optional or mandatory items to these records.

8 Acronyms and Abbreviations

Abbreviation or Acronym	Meaning
BNF	Backus-Naur Form
DTD	XML Document Type Definition
FRD	Feature Requirements Document
GAP	Generic Access Profile
GPP	Generic PIM Profile
GOEP	Generic Object Exchange Profile
L2CAP	Logical Link Control and Adaptaion Protocol
LMP	Link Manager Protocol
MAP	Message Access Profile
MRD	Market Requirements Document
MSC	Message Sequence Chart
PAS	PIM Access Service
PIM	Personal Information Management
PIMCE	PIM Client Equipment
PIMSE	PIM Server Equipment
PNS	PIM Notification Service
PSM	Protocol Service Multiplexer
SDP	Service Discovery Protocol
WG	Working Group
XML	eXtensible Markup Language
XSD	XML Schema Definition

Table 8.1: *Abbreviations and Acronyms*

9 References

[1] Bluetooth Core Specification, Version 2.1 or later

[2] IrDA Object Exchange Protocol OBEX™ with Published Errata, Version 1.4, July 2007, iAnywhere Inc, Extended systems Inc. and Microsoft Corporation

[3] Generic Object Exchange Profile v2.0 , 26 August 2010, Bluetooth SIG

[4] Generic Access Profile Specification, latest version applies, Bluetooth SIG