Part K:10

GENERIC OBJECT EXCHANGE PROFILE

This profile defines the requirements for Bluetooth devices necessary for the support of the object exchange usage models. The requirements are expressed by defining the features and procedures that are required for interoperability between Bluetooth devices in the object exchange usage models.

CONTENTS

FOREWORD

The purpose of this document is to work as a generic profile document for all application profiles using the OBEX protocol.

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 Generic Object Exchange profile defines the protocols and procedures that shall be used by the applications providing the usage models which need the object exchange capabilities. The usage model can be, for example, Synchronization, File Transfer, or Object Push model. The most common devices using these usage models can be notebook PCs, PDAs, smart phones, and mobile phones.

1.2 BLUETOOTH PROFILE STRUCTURE

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

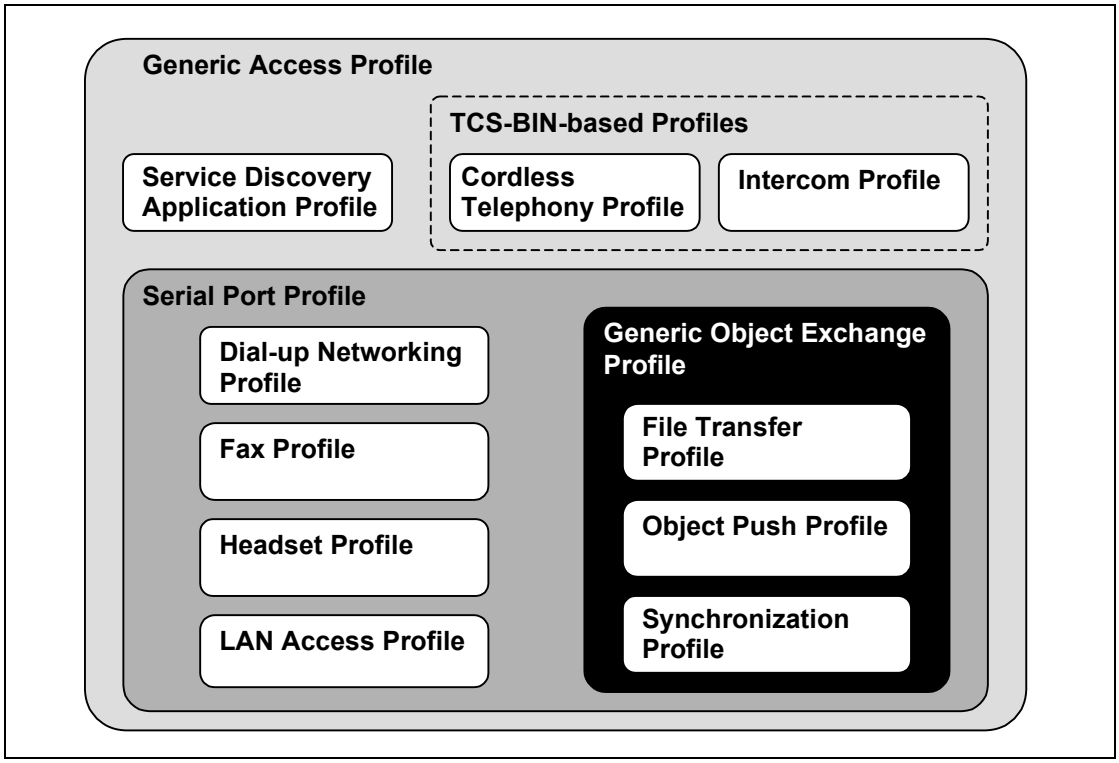

Figure 1.1: Bluetooth Profiles

1.3 BLUETOOTH OBEX-RELATED SPECIFICATIONS

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

1. Bluetooth IrDA Interoperability Specification [1]

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

2. Bluetooth Generic Object Exchange Profile Specification (This specification)

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

3. Bluetooth Synchronization Profile Specification [2]

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

4. Bluetooth File Transfer Profile Specification [3]

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

5. Bluetooth Object Push Profile Specification [4]

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

1.4 SYMBOLS AND CONVENTIONS

1.4.1 Requirement status symbols

In this document, the following symbols are used:

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

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

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

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

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

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

1.4.2 Signaling diagram conventions

The following arrows are used in diagrams describing procedures:

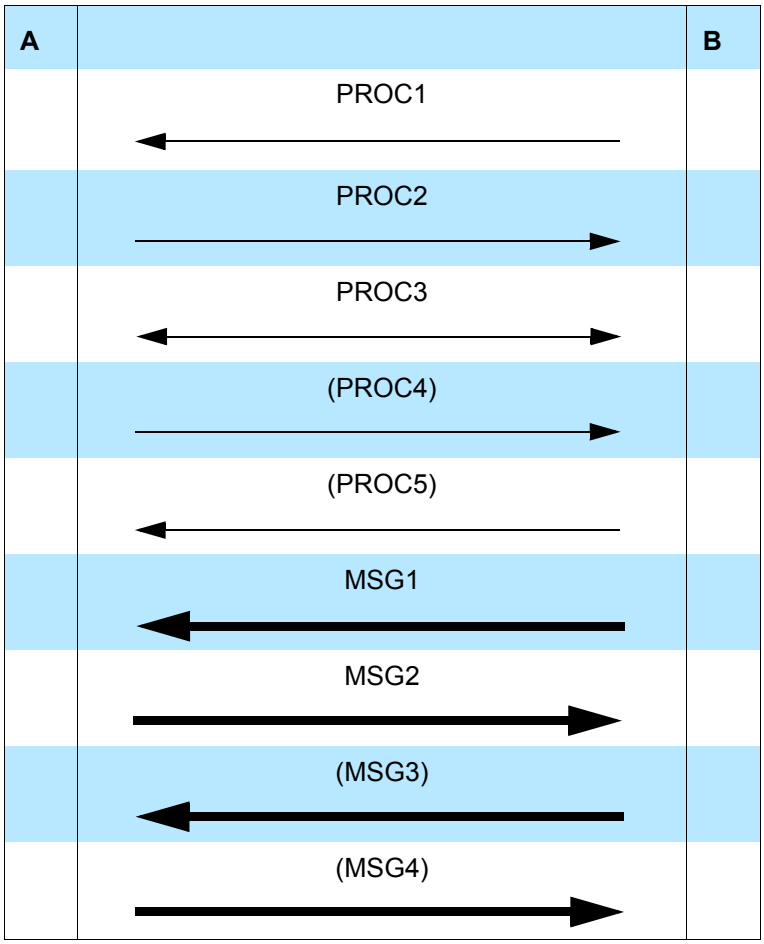

Table 1.1: Arrows used in signaling diagrams

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

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

2 PROFILE OVERVIEW

2.1 PROFILE STACK

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

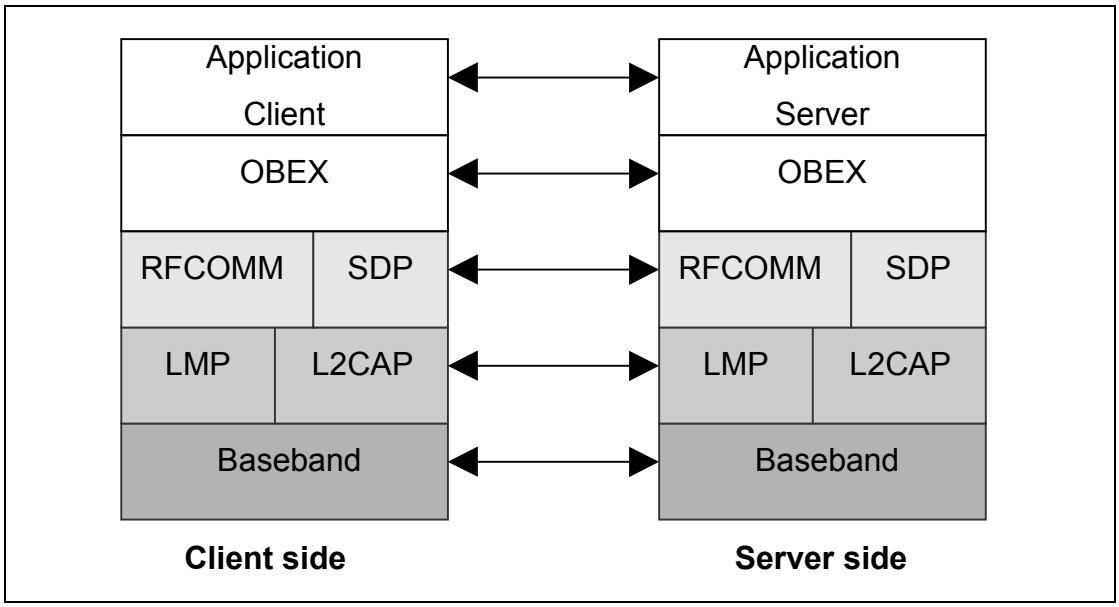

Figure 2.1: Protocol model

The Baseband [5], LMP [6] and L2CAP [7] are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM [8] is the Bluetooth adaptation of GSM TS 07.10 [9]. SDP is the Bluetooth Service Discovery Protocol [10]. OBEX [1] is the Bluetooth adaptation of IrOBEX [11].

The Application Client layer shown in Figure 2.1 is the entity sending and retrieving data object from the Server using the OBEX operations. The application Server is the data storage to and from which the data object can be sent or retrieved.

2.2 CONFIGURATIONS AND ROLES

The following roles are defined for this profile:

Server – This is the device that provides an object exchange server to and from which data objects can be pushed and pulled, respectively.

Client – This is the device that can push or/and pull data object(s) to and from the Server.

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are the following:

- Usage of a Server by a Client to push data object(s)
- Usage of a Server by a Client to pull data object(s)

The following restrictions apply to this profile:

a) For the device containing the Server, it is assumed that the user may have to put it into the discoverable and connectable modes when the inquiry and link establishment procedures, respectively, are processed in the Client (see Generic Access Profile).

b) The profile only supports point-to-point configurations. As a result, the Server is assumed to offer services only for one Client at a time. However, the implementation may offer a possibility for multiple Clients at a time but this is not a requirement.

2.4 PROFILE FUNDAMENTALS

The profile fundamentals, with which all application profiles must comply, are the following:

1. Before a Server is used with a Client for the first time, a bonding procedure including the pairing may be performed (see Section 7.3.1). This procedure must be supported, but its usage is dependent on the application profiles. The bonding typically involves manually activating bonding support and entering a Bluetooth PIN code (see Section 7.3.1) on the keyboards of the Client and Server devices. This procedure may have to be repeated under certain circumstances; for example, if a common link key (as a bonding result) is removed on the device involved in the object exchange.
2. In addition to the link level bonding, an OBEX initialization procedure may be performed (see Section 5.3) before the Client can use the Server for the first time. The application profiles using GOEP must specify whether this procedure must be supported to provide the required security level.
3. Security can be provided by authenticating the other party upon connection establishment, and by encrypting all user data on the link level. The authentication and encryption must be supported by the devices; but whether they are used depends on the application profile using GOEP.
4. Link and channel establishments must be done according to the procedures defined in GAP (see Section 7.1-7.2 in [14]). Link and channel establishment procedures in addition to the procedures in GAP must not defined by the application profiles using GOEP.
5. There are no fixed master/slave roles.
6. This profile does not require any lower power mode to be used.

3 USER INTERFACE ASPECTS

User interface aspects are not defined in this profile.They are instead defined in the application profiles, where necessary.

4 APPLICATION LAYER

This section describes the service capabilities which can be utilized by the application profiles using GOEP.

4.1 FEATURE OVERVIEW

Table 4.1 shows the features which the Generic Object Exchange profile provides for the application profiles. The usage of other features (e.g. setting the current directory) must be defined by the applications profiles needing them.

Feature no.	Feature
1	Establishing an Object Exchange session
2	Pushing a data object
3	Pulling a data object

Table 4.1: Features provided by GOEP

4.2 ESTABLISHING AN OBJECT EXCHANGE SESSION

This feature is used to establish the object exchange session between the Client and Server. Before a session is established, payload data cannot be exchanged between the Client and the Server. The usage of the OBEX operations for establishing an OBEX session is described in Section 5.4.

4.3 PUSHING A DATA OBJECT

If data needs to be transferred from the Client to the Server, then this feature is used. The usage of the OBEX operations for pushing the data object(s) is described in Section 5.5.

4.4 PULLING A DATA OBJECT

If data need to be transferred from the Server to the Client, then this feature is used. The usage of the OBEX operations for pulling the data object(s) is described in Section 5.6.

5 OBEX INTEROPERABILITY REQUIREMENTS

5.1 OBEX OPERATIONS USED

Table 5.1 shows the OBEX operations which are specified by the OBEX protocol. The application profiles using GOEP must specify which operations must be supported to provide the functionality defined in the application profiles.

Operation no.	OBEX Operation
1	Connect
2	Disconnect
3	Put
4	Get
5	Abort
6	SetPath

Table 5.1: OBEX Operations

The IrOBEX specification does not define how long a client should wait for a response to an OBEX request. However, implementations which do not provide a user interface for canceling an OBEX operation should wait a reasonable period between a request and response before automatically canceling the operation. A reasonable time period is 30 seconds or more.

5.2 OBEX HEADERS

Table 5.2 shows the specified OBEX headers.

Header no.	OBEX Headers
1	Count
2	Name
3	Type
4	Length
5	Time
6	Description
7	Target
8	HTTP

Table 5.2: OBEX Headers

Header no.	OBEX Headers
9	Body
10	End of Body
11	Who
12	Connection ID
13	Authenticate Challenge
14	Authenticate Response
15	Application Parameters
16	Object Class

Table 5.2: OBEX Headers

Applications profiles dedicated to specific usage models must specify which of these headers must be supported.

5.3 INITIALIZATION OF OBEX

If the OBEX authentication is supported and used by the Server and the Client, the initialization for this authentication (see also Section 5.4.2) must be done before the first OBEX connection can be established. The initialization can be done at any time before the first OBEX connection. The initialization of the OBEX authentication requires user intervention on both the Client device and the Server device.

Authentication is done using an OBEX password, which may be the same as a Bluetooth PIN code on the link level. Even if the user uses the same code for link authentication and OBEX authentication, the user must enter these codes separately. After entering the OBEX password in both the Client and Server, the OBEX password is stored in the Client and the Server, and it can be used in the future for authenticating the Client and the Server. When an OBEX connection is established, the devices must authenticate each other if the OBEX authentication is enabled.

5.4 ESTABLISHMENT OF OBEX SESSION

For the Object Exchange, the OBEX connection can be made with or without OBEX authentication. In the next two subsections, both of these cases are explained. All application profiles using GOEP must support an OBEX session without authentication.

5.4.1 OBEX Session without Authentication

Figure 5.1 depicts how an OBEX session is established using the CONNECT operation.

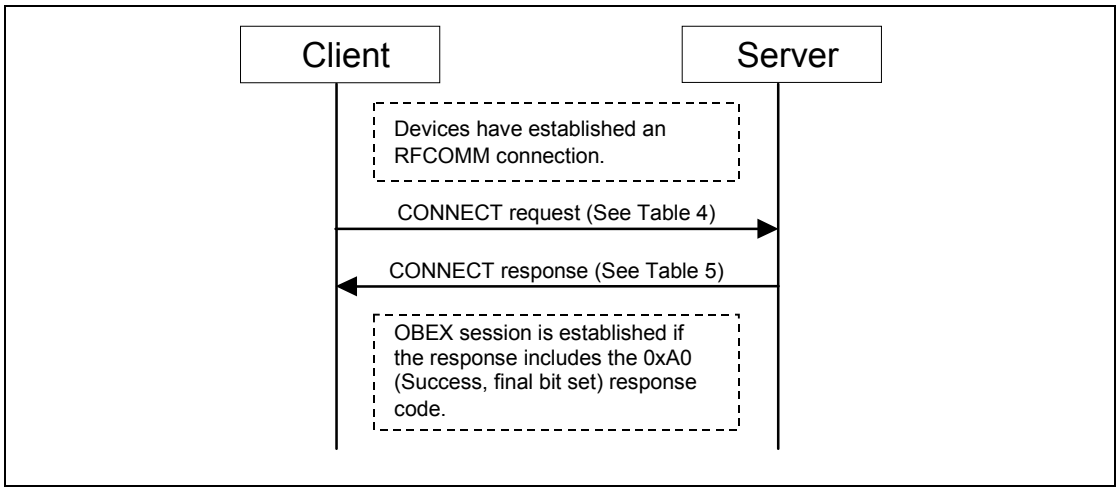

Figure 5.1: Establishment of OBEX Session without Authentication

The CONNECT request indicates a need for connection and may also indicate which service is used. The fields in the CONNECT request are listed below:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for CONNECT	0x80	M	-
Field	Connect Packet Length	Varies	M	-
Field	OBEX Version Number	Varies	M	-
Field	Flags	Varies	M	-
Field	Max OBEX Packet Length	Varies	M	-
Header	Target	Varies	C1	Used to indicate the specific Service.

Table 5.3: Fields and Headers in CONNECT Request

C1: The use of the Target header is mandatory for some application profiles. The application profiles define explicitly whether they use it or not. For the Target header, the example value could be ‘IRMC-SYNC’ to indicate the IrMC synchronization service. The target header is placed after the Maximum OBEX Packet Length field in the CONNECT request.

The response to the CONNECT request includes the fields listed below:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for CONNECT request	Varies	M	0xA0 for success
Field	Connect Response Packet Length	Varies	M	-
Field	OBEX Version Number	Varies	M	-
Field	Flags	Varies	M	-
Field	Max OBEX Packet Length	Varies	M	-
Header	ConnectionID	Varies	C2	The header value specifies the current connection to the specific service.
Header	Who	Varies	C2	The header value matches the Target header value.

Table 5.4: Fields and Headers in CONNECT Response

C2: The Who and Connection ID headers must be used if the Target header is used in the Connect request. These headers are placed after the Maximum OBEX Packet Length field in the response to the CONNECT request.

5.4.2 OBEX Session with Authentication

The OBEX authentication scheme is based on the HTTP scheme but does not have all the features and options. In GOEP, OBEX authentication is used to authenticate the Client and the Server. Figure 5.2 depicts establishment of an OBEX session with authentication.

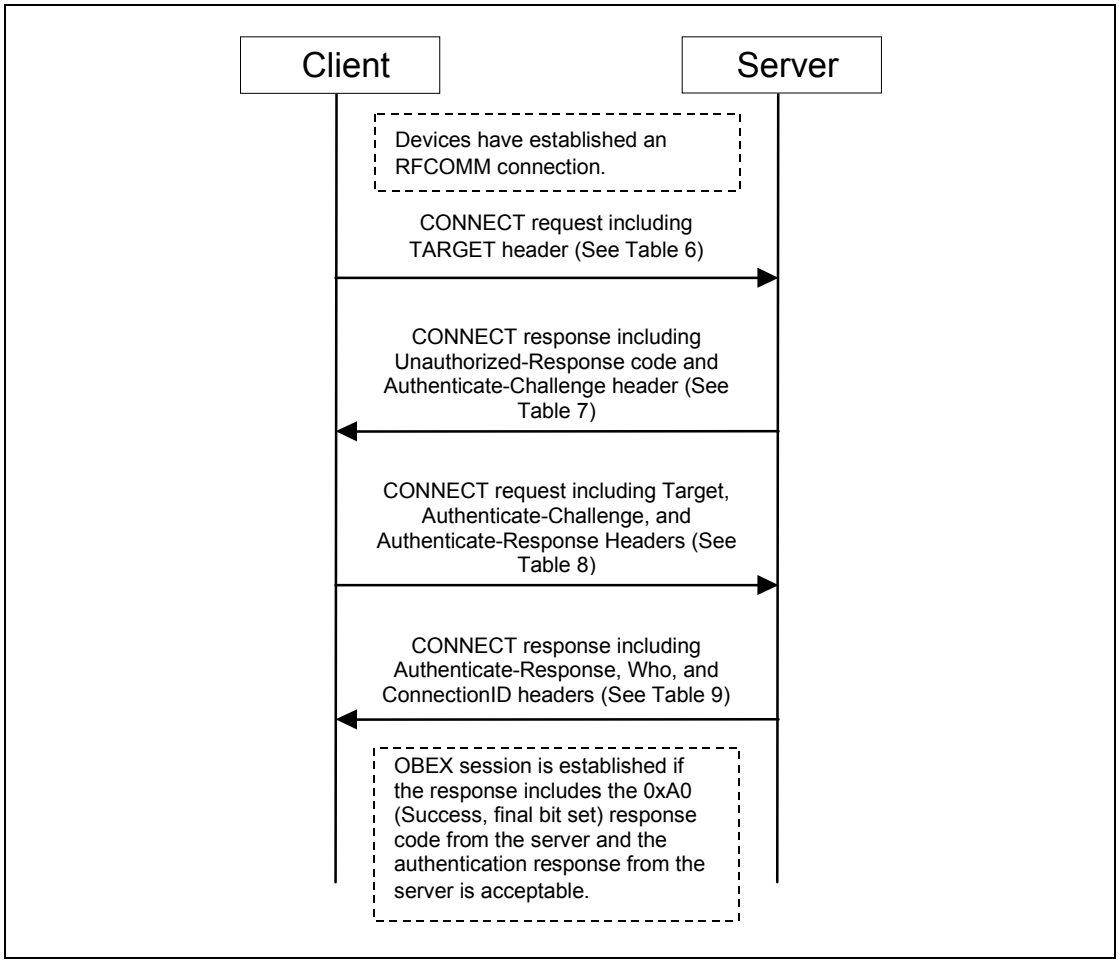

Figure 5.2: Establishment of OBEX Session with Authentication

The first CONNECT request indicates a need for connection and which service is used. The fields and the header in the CONNECT request are listed below:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for CONNECT	0x80	M	-
Field	Connect Packet Length	Varies	M	-
Field	OBEX Version Number	Varies	M	-

Table 5.5: Fields and Headers in CONNECT Request when Authentication Used

Field/ Header	Name	Value	Status	Explanation
Field	Flags	Varies	M	-
Field	Max OBEX Packet Length	Varies	M	-
Header	Target	Varies	C1	Used to indicate the specific Service

Table 5.5: Fields and Headers in CONNECT Request when Authentication Used

C1: The usage of the Target header is dependent on the application profile utilizing GOEP. The example value for the Target header can be ‘IRMC-SYNC’ to indicate the IrMC synchronization service.

The first response to the CONNECT request from the Server, which authenticates the Client, includes the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for CONNECT request	Varies	M	0x41 for Unauthorized, because OBEX authentication is used.
Field	Connect Response Packet Length	Varies	M	-
Field	OBEX Version Number	Varies	M	-
Field	Flags	Varies	M	-
Field	Max OBEX Packet Length	Varies	M	-
Header	Authenticate Challenge	Varies	M	Carries the digest-challenge string.

Table 5.6: Fields and Headers in First CONNECT Response when Authenticating

The second CONNECT request has the following fields and headers in this order:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for CONNECT	0x80	M	-
Field	Connect Packet Length	Varies	M	-
Field	OBEX Version Number	Varies	M	-
Field	Flags	Varies	M	-
Field	Max OBEX Packet Length	Varies	M	-

Table 5.7: Fields and Headers in Second CONNECT Request when Authentication Used

Field/ Header	Name	Value	Status	Explanation
Header	Target	Varies	C1	-
Header	Authenticate Challenge	Varies	M	Carries the digest-challenge string.
Header	Authenticate Response	Varies	M	Carries the digest-response string. This is the response to the challenge from the Server.

Table 5.7: Fields and Headers in Second CONNECT Request when Authentication Used

C1: see Table 5.5

The second response to the CONNECT request has the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for CONNECT request	Varies	M	0xA0 for success
Field	Connect Response Packet Length	Varies	M	-
Field	OBEX Version Number	Varies	M	-
Field	Flags	Varies	M	-
Field	Max OBEX Packet Length	Varies	M	-
Header	ConnectionID	Varies	M	The header value specifies the current connection to the specific service.
Header	Who	Varies	M	The header value matches the Target header value.
Header	Authenticate Response	Varies	M	Carries the digest-response string. This is the response to the challenge from the Client.

Table 5.8: Fields and Headers in Second CONNECT Response when Authenticating

If the response code from the Server is successful, and the Client accepts the authentication response from the Server, the session is established and authenticated.

5.5 PUSHING DATA TO SERVER

The data object(s) is pushed to the Server using the PUT operation of the OBEX protocol. The data can be sent in one or more OBEX packets.
The PUT request must include the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for PUT	0x02 or 0x82	M	-
Field	Packet Length	Varies	M	-
Header	ConnectionID	Varies	C1	The header value specifies the current connection to the specific service.
Header	Name	Varies	M	The header value is the name of a single object, object store, or log information.
Header	Body/End of Body	Varies	M	End of Body identifies the last chunk of the object body.

Table 5.9: Fields and Headers in PUT Request

C1: The ConnectionID header is mandatory if the Target header is used when establishing the OBEX session.

Other headers, which can be optionally used, are specified in [11].

The response packet for the PUT request has the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for PUT	0x90 or 0xAO	M	0x90 for continue or 0xA0 for success
Field	Packet Length	Varies	M	-

Table 5.10: Fields and Headers in PUT Response

Other headers, which can be optionally used, are specified in [11].

5.6 PULLING DATA FROM SERVER

The data object(s) is pulled from the Server using the GET operation of the OBEX protocol. The data can be sent in one or more OBEX packets. The first GET request includes the following fields and headers.

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for GET	0x03	M	-
Field	Packet Length	Varies	M	-
Header	ConnectionID	Varies	C1	The header value specifies the current connection to the specific service.
Header	Type	Varies	C2	Indicates the type of the object to be pulled.
Header	Name	Varies	C2	The header value is the name of a single object, object store, or log information.

Table 5.11: Fields and Headers in GET Request

C1: The ConnectionID header is mandatory if the Target header is used when establishing the OBEX session.

C2: Either the Type header or the Name header must be included in the GET request when it is sent to the server.

Other headers, which can be optionally used, are specified in [11].

The response packet for the GET request has the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Response code for Get	0x90 or 0xA0	M	0x90 or 0xA0 if the packet is the last the object
Field	Packet Length	Varies	M	-
Header	Name	Varies	O	The header value is the name of a single object, object store, or log information.
Header	Body/End of Body	Varies	M	End of Body identifies the last chunk of the object body.

Table 5.12: Fields and Headers in GET Response

Other headers, which can be optionally used, are specified in [11].

5.7 DISCONNECTION

see Chapter 2.2.2 in [1].

6 SERIAL PORT PROFILE INTEROPERABILITY REQUIREMENTS

This profile requires compliance to the protocol requirements of the Serial Port Profile (SeP) [12]. For the purposes of reading the SeP [12], the Server shall always be considered to be Device B and the Client shall always be considered to be Device A.

The following text, together with the associated sub-clauses, defines the requirements with regards to this profile – in addition to the requirements defined in [9].

6.1 RFCOMM INTEROPERABILITY REQUIREMENTS

For the RFCOMM layer, no additions to the requirements stated in Section 4 of Serial Port Profile apply.

6.2 L2CAP INTEROPERABILITY REQUIREMENTS

For the L2CAP layer, no additions to the requirements stated in Section 5 of Serial Port Profile apply.

6.3 SDP INTEROPERABILITY REQUIREMENTS

These requirements are defined by the application profiles. Thus, none of the requirements defined in the SeP profile (Section 6 in [12]) apply to this profile.

6.4 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

For the LM layer, no additions to the requirements stated in Section 7 of Serial Port Profile apply.

6.5 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

In the table below, LC capabilities differing from the capabilities required by the SeP profile (Section 8 in [12]) are listed.

	Capabilities	Support in baseband	Support in Server	Support in Client
5.	Packet types			
L	HV1 packet	M	X	X

Table 6.1: Baseband/LC capabilities

	Capabilities	Support in baseband	Support in Server	Support in Client
M	HV2 packet	O	X	X
N	HV3 packet	O	X	X
O	DV packet	M	X	X
7.	Voice codec			
A	A-law	O	X	X
B	μ-law	O	X	X
C	CVSD	O	X	X

Table 6.1: Baseband/LC capabilities

6.5.1 Inquiry and Inquiry Scan

For this profile, the Limited discoverable mode (see Section 7.1) should be used; but, if the Server device for some reason (e.g. lack of a sufficient user interface) wants to be visible at all times, the General discoverable mode (see Section 7.1) can be used instead. The client device must support the General inquiry procedure (see Section 7.3), and should also support the Limited inquiry procedure.

If the Limited inquiry procedure is supported, it should be used primarily. When this procedure is initiated in the Client, the client must perform this procedure for at least $T_{GAP}(100)$ (see Section 6.2.4 in GAP [14]). After the execution of the Limited inquiry procedure, the device may fall back to perform the General inquiry procedure. The device must support this fall-back functionality if the Limited inquiry procedure is supported. The fall-back procedure may or may not require user intervention. When general inquiry is initiated by the Client after limited inquiry, it shall be in this General limited procedure state for at least $T_{GAP}(100)$ (see Section 6.2.4 in GAP [14]).

For the inquiry, the returned CoD in the FHS packet must indicate that Object Transfer service is supported (see [13]). The major and minor device classes depend on the device supporting this profile. Therefore, usage of them is not defined in this profile.

The Limited Inquiry, Device Discovery and Name Discovery procedures are described in Section 6.2-6.4 in the Generic Access profile [14].

7 GENERIC ACCESS PROFILE INTEROPERABILITY REQUIREMENTS

This profile requires compliance to the Generic Access Profile. This section defines the support requirements with regards to procedures and capabilities defined in GAP.

7.1 MODES

Table 7.1 shows the support status for Modes within this profile.

	Procedure	Support in Client	Support in Server
1	Discoverability modes		
	Non-discoverable mode	N/A	M
	Limited discoverable mode	N/A	C1
	General discoverable mode	N/A	C1
2	Connectability modes		
	Non-connectable mode	N/A	O
	Connectable mode	N/A	M
3	Pairing modes		
	Non-pairable mode	N/A	M
	Pairable mode	N/A	M

Table 7.1: Modes

C1: The Limited discoverable mode should be used, but if the Server device for some reason (e.g. lack of a sufficient user interface) wants to be visible at all times, the General discoverable mode can be used instead.

7.2 SECURITY ASPECTS

Table 7.2 shows the support status for Security aspects within this profile.

	Procedure	Support in Client	Support in Server
1	Authentication	M	M
2	Security modes		

Table 7.2: Security aspects

	Procedure	Support in Client	Support in Server
	Security modes 1	M	M
	Security modes 2	C1	C1
	Security modes 3	C1	C1

Table 7.2: Security aspects

C1: Support for at least one of the security modes 2 and 3 is mandatory.

7.3 IDLE MODE PROCEDURES

Table 7.3 shows the support status for Idle mode procedures within this profile.

	Procedure	Support in Client	Support in Server
1	General inquiry	M	N/A
2	Limited inquiry	O	N/A
3	Name discovery	M	N/A
4	Device discovery	M	N/A
5	Bonding	M (Note 1)	M (Note 1)
Note 1: see section 7.3.1			

Table 7.3: Idle mode procedures

7.3.1 Bonding

It is mandatory for the Client and Server to support bonding. Bonding may be required before permitting communication between a Client and a Server. During bonding, the Client and Server are paired, which means that the Client and Server establish a security association (a common link key). This requires that an identical Bluetooth PIN code be entered on both the Client and Server devices.

The usage of bonding is optional for both Client and Server. The bonding procedures are defined in Section 6.5 in GAP [14].

8 REFERENCES

8.1 NORMATIVE REFERENCES

[1] Bluetooth Special Interest Group, IrDA Interoperability

[2] Bluetooth Special Interest Group, Synchronization Profile

[3] Bluetooth Special Interest Group, File Transfer Profile

[4] Bluetooth Special Interest Group, Object Push Profile

[5] Bluetooth Special Interest Group, Baseband Specification

[6] Bluetooth Special Interest Group, LMP Specification

[7] Bluetooth Special Interest Group, L2CAP Specification

[8] Bluetooth Special Interest Group, RFCOMM with TS 07.10", Specification of the Bluetooth System

[9] ETSI, TS 07.10, Version 6.3.0

[10] Bluetooth Special Interest Group, SDP Specification

[11] Infrared Data Association, IrDA Object Exchange Protocol (IrOBEX) with Published Errata, Version 1.2, April 1999

[12] Bluetooth Special Interest Group, Serial Port Profile

[13] Internet Assigned Numbers Authority, IANA Protocol/Number Assignments Directory (http://www.iana.org/numbers.html), May 1999.

[14] Bluetooth Special Interest Group, Generic Access Profile