Specification of the Bluetooth System

Wireless connections made easy

Profiles

Bluetooth™

v1.0 B
December 1st 1999

MEDIATEK-1044
7,483,686

BLUETOOTH DOC	Date / Day-Month-Year 01 Dec 99	N.B.	Document No. **1.C.47/1.0 B**
Responsible	e-mail address		Status

Bluetooth™

Profiles of the Bluetooth System

Version 1.0B

Revision History

The Revision History is shown in Appendix I on page 413

Contributors

The persons who contributed to this specification are listed in Appendix II on page 421.

Web Site

This specification can also be found on the Bluetooth web site:
http://www.bluetooth.com

Disclaimer and copyright notice

THIS SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Copyright © 1999

Telefonaktiebolaget LM Ericsson,
International Business Machines Corporation,
Intel Corporation,
Nokia Corporation,
Toshiba Corporation .

*Third-party brands and names are the property of their respective owners.

MASTER TABLE OF CONTENTS

For the Core Specification, *see Volume 1*

Part K:1

GENERIC ACCESS PROFILE

Contents 15

Foreword 19
1 Introduction 20
2 Profile Overview 22
3 User Interface Aspects 25
4 Modes 29
5 Security Aspects 33
6 Idle Mode Procedures 37
7 Establishment Procedures 45
8 Definitions 52
9 Annex A (Normative): Timers and Constants 56
10 Annex B (Informative): Information Flows of Related Procedures 57
11 References 60

Part K:2

SERVICE DISCOVERY APPLICATION PROFILE

Contents 63

Foreword 65
1 Introduction 66
2 Profile Overview 68
3 User Interface Aspects 72
4 Application Layer 73
5 Service Discovery 79
6 L2CAP 82
7 Link Manager 86
8 Link Control 88
9 References 91
10 Definitions 92
11 Appendix A (Informative): Service Primitives and the Bluetooth PDUs 93

Bluetooth

Part K:3

CORDLESS TELEPHONY PROFILE

Contents **97**

1 Introduction 100
2 Profile Overview 103
3 Application Layer 108
4 TCS-BIN Procedures 110
5 Service Discovery Procedures 120
6 L2CAP Procedures 121
7 LMP Procedures Overview 122
8 LC Features 124
9 General Access Profile Interoperability Requirements 126
10 Annex A (Informative): Signalling Flows 128
11 Timers and Counters 135
12 References 136
13 List of Figures 137
14 List of Tables 138

Part K:4

INTERCOM PROFILE

Contents **141**

1 Introduction 143
2 Profile Overview 145
3 Application Layer 148
4 TCS Binary 149
5 SDP Interoperability Requirements 153
6 L2CAP Interoperability Requirements 154
7 Link Manager (LM) Interoperability Requirements 155
8 Link Control (LC) Interoperability Requirements 156
9 Generic Access Profile 158
10 Annex A (Informative): Signalling flows 159
11 Timers and Counters 161
12 List of Figures 162
13 List of Tables 163

Bluetooth

Part K:5

SERIAL PORT PROFILE

Contents 167

Foreword 169

1 Introduction 170

2 Profile Overview 171

3 Application Layer.....174

4 RFCOMM Interoperability Requirements 177

5 L2CAP Interoperability Requirements.....179

6 SDP Interoperability Requirements.....181

7 Link Manager (LM) Interoperability Requirements 183

8 Link Control (LC) Interoperability Requirements 184

9 References.....186

10 List of Figures.....187

11 List of Tables 188

Part K:6

HEADSET PROFILE

Contents 191

1 Introduction 193

2 Profile Overview 196

3 Application Layer.....200

4 Headset Control Interoperability Requirements 201

5 Serial Port Profile 210

6 Generic Access Profile.....214

7 References.....215

8 List of Figures.....216

9 List of Tables 217

Bluetooth

Part K:7

DIAL-UP NETWORKING PROFILE

Contents 221

1 Introduction 223
2 Profile Overview 226
3 Application Layer 230
4 Dialling and Control Interoperability Requirements 231
5 Serial Port Profile Interoperability Requirements 235
6 Generic Access Profile Interoperability Requirements 238
7 References 240
8 List of Figures 241
9 List of Tables 242

Part K:8

FAX PROFILE

Contents 245

1 Introduction 246
2 Profile Overview 249
3 Application Layer 253
4 Dialling and Control Interoperability Requirements 254
5 Serial Port Profile 256
6 Generic Access Profile Interoperability Requirements 259
7 References 261
8 List of Figures 262
9 List of Tables 263

Bluetooth

Part K:9

LAN ACCESS PROFILE

Contents 267

1 Introduction 269
2 Profile Overview 271
3 User Interface Aspects 275
4 Application Layer 278
5 PPP 281
6 RFCOMM 284
7 Service Discovery 285
8 L2CAP 287
9 Link Manager 288
10 Link Control 290
11 Management Entity Procedures 291
12 APPENDIX A (Normative): Timers and counters 293
13 APPENDIX B (Normative): Microsoft Windows 294
14 APPENDIX C (Informative): Internet Protocol (IP) 295
15 List of Figures 297
16 List of Tables 298
17 References 299

Part K:10

GENERIC OBJECT EXCHANGE PROFILE

Contents 303

Foreword 305
1 Introduction 306
2 Profile Overview 310
3 User Interface Aspects 312
4 Application Layer 313
5 OBEX Interoperability Requirements 314
6 Serial Port Profile Interoperability Requirements 324
7 Generic Access Profile Interoperability Requirements 326
8 References 328

Bluetooth

Part K:11

OBJECT PUSH PROFILE

Contents 331

Foreword 333
1 Introduction 334
2 Profile Overview 338
3 User Interface Aspects 340
4 Application Layer 344
5 OBEX 348
6 Service Discovery 351
7 References 353

Part K:12

FILE TRANSFER PROFILE

Contents 357

Foreword 359
1 Introduction 360
2 Profile Overview 364
3 User Interface Aspects 367
4 Application Layer 370
5 OBEX 374
6 Service Discovery 383
7 References 385

Part K:13

SYNCHRONIZATION PROFILE

Contents 389

Foreword 391
1 Introduction 392
2 Profile Overview 396
3 User Interface Aspects 399
4 Application Layer 402
5 IrMC Synchronization Requirements 404
6 OBEX 406
7 Service Discovery 408
8 References 411

Bluetooth™

Appendix I

REVISION HISTORY 413

Appendix II

CONTRIBUTORS 421

Appendix III

ACRONYMS AND ABBREVIATIONS 429

INDEX 435

Part K:1

GENERIC ACCESS PROFILE

This profile defines the generic procedures related to discovery of Bluetooth devices (idle mode procedures) and link management aspects of connecting to Bluetooth devices (connecting mode procedures). It also defines procedures related to use of different security levels. In addition, this profile includes common format requirements for parameters accessible on the user interface level.

CONTENTS

1 Introduction 20
1.1 Scope 20
1.2 Symbols and conventions 20
1.2.1 Requirement status symbols 20
1.2.2 Signalling diagram conventions 21
1.2.3 Notation for timers and counters 21
2 Profile overview 22
2.1 Profile stack 22
2.2 Configurations and roles 22
2.3 User requirements and scenarios 23
2.4 Profile fundamentals 23
2.5 Conformance 24
3 User interface aspects 25
3.1 The user interface level 25
3.2 Representation of Bluetooth parameters 25
3.2.1 Bluetooth device address (BD_ADDR) 25
3.2.1.1 Definition 25
3.2.1.2 Term on user interface level 25
3.2.1.3 Representation 25
3.2.2 Bluetooth device name (the user-friendly name) 25
3.2.2.1 Definition 25
3.2.2.2 Term on user interface level 26
3.2.2.3 Representation 26
3.2.3 Bluetooth passkey (Bluetooth PIN) 26
3.2.3.1 Definition 26
3.2.3.2 Terms at user interface level 26
3.2.3.3 Representation 26
3.2.4 Class of Device 27
3.2.4.1 Definition 27
3.2.4.2 Term on user interface level 27
3.2.4.3 Representation 27
3.3 Pairing 28

4 Modes 29
4.1 Discoverability modes 29
4.1.1 Non-discoverable mode 30
4.1.1.1 Definition 30
4.1.1.2 Term on UI-level 30
4.1.2 Limited discoverable mode 30
4.1.2.1 Definition 30
4.1.2.2 Conditions 31
4.1.2.3 Term on UI-level 31
4.1.3 General discoverable mode 31
4.1.3.1 Definition 31
4.1.3.2 Conditions 31
4.1.3.3 Term on UI-level 31
4.2 Connectability modes 31
4.2.1 Non-connectable mode 31
4.2.1.1 Definition 31
4.2.1.2 Term on UI-level 32
4.2.2 Connectable mode 32
4.2.2.1 Definition 32
4.2.2.2 Term on UI-level 32
4.3 Pairing modes 32
4.3.1 Non-pairable mode 32
4.3.1.1 Definition 32
4.3.1.2 Term on UI-level 32
4.3.2 Pairable mode 32
4.3.2.1 Definition 32
4.3.2.2 Term on UI-level 32
5 Security aspects 33
5.1 Authentication 33
5.1.1 Purpose 33
5.1.2 Term on UI level 33
5.1.3 Procedure 34
5.1.4 Conditions 34
5.2 Security modes 34
5.2.1 Security mode 1 (non-secure) 36
5.2.2 Security mode 2 (service level enforced security) 36
5.2.3 Security modes 3 (link level enforced security) 36

6 Idle mode procedures 37

6.1 General inquiry 37

6.1.1 Purpose 37

6.1.2 Term on UI level 37

6.1.3 Description 38

6.1.4 Conditions 38

6.2 Limited inquiry 38

6.2.1 Purpose 38

6.2.2 Term on UI level 39

6.2.3 Description 39

6.2.4 Conditions 39

6.3 Name discovery 40

6.3.1 Purpose 40

6.3.2 Term on UI level 40

6.3.3 Description 40

6.3.3.1 Name request 40

6.3.3.2 Name discovery 40

6.3.4 Conditions 41

6.4 Device discovery 41

6.4.1 Purpose 41

6.4.2 Term on UI level 41

6.4.3 Description 42

6.4.4 Conditions 42

6.5 Bonding 42

6.5.1 Purpose 42

6.5.2 Term on UI level 42

6.5.3 Description 43

6.5.3.1 General bonding 43

6.5.3.2 Dedicated bonding 44

6.5.4 Conditions 44

7 Establishment procedures 45
7.1 Link establishment 45
7.1.1 Purpose 45
7.1.2 Term on UI level 45
7.1.3 Description 46
7.1.3.1 B in security mode 1 or 2 46
7.1.3.2 B in security mode 3 47
7.1.4 Conditions 47
7.2 Channel establishment 48
7.2.1 Purpose 48
7.2.2 Term on UI level 48
7.2.3 Description 48
7.2.3.1 B in security mode 2 49
7.2.3.2 B in security mode 1 or 3 49
7.2.4 Conditions 49
7.3 Connection establishment 50
7.3.1 Purpose 50
7.3.2 Term on UI level 50
7.3.3 Description 50
7.3.3.1 B in security mode 2 50
7.3.3.2 B in security mode 1 or 3 51
7.3.4 Conditions 51
7.4 Establishment of additional connection 51
8 Definitions 52
8.1 General definitions 52
8.2 Connection-related definitions 52
8.3 Device-related definitions 53
8.4 Procedure-related definitions 54
8.5 Security-related definitions 54
9 Annex A (Normative): Timers and constants 56
10 Annex B (Informative): Information flows of related procedures 57
10.1 lmp-authentication 57
10.2 lmp-pairing 58
10.3 Service discovery 58
11 References 60

FOREWORD

Interoperability between devices from different manufacturers is provided for a specific service and use case, 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 use case(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 purpose of the Generic Access Profile is:

To introduce definitions, recommendations and common requirements related to modes and access procedures that are to be used by transport and application profiles.

To describe how devices are to behave in standby and connecting states in order to guarantee that links and channels always can be established between Bluetooth devices, and that multi-profile operation is possible. Special focus is put on discovery, link establishment and security procedures.

To state requirements on user interface aspects, mainly coding schemes and names of procedures and parameters, that are needed to guarantee a satisfactory user experience.

1.2 SYMBOLS AND CONVENTIONS

1.2.1 Requirement status symbols

In this document (especially in the profile requirements tables), 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.

In this specification, the word *shall* is used for mandatory requirements, the word *should* is used to express recommendations and the word *may* is used for options.

1.2.2 Signalling diagram conventions

The following arrows are used in diagrams describing procedures
:

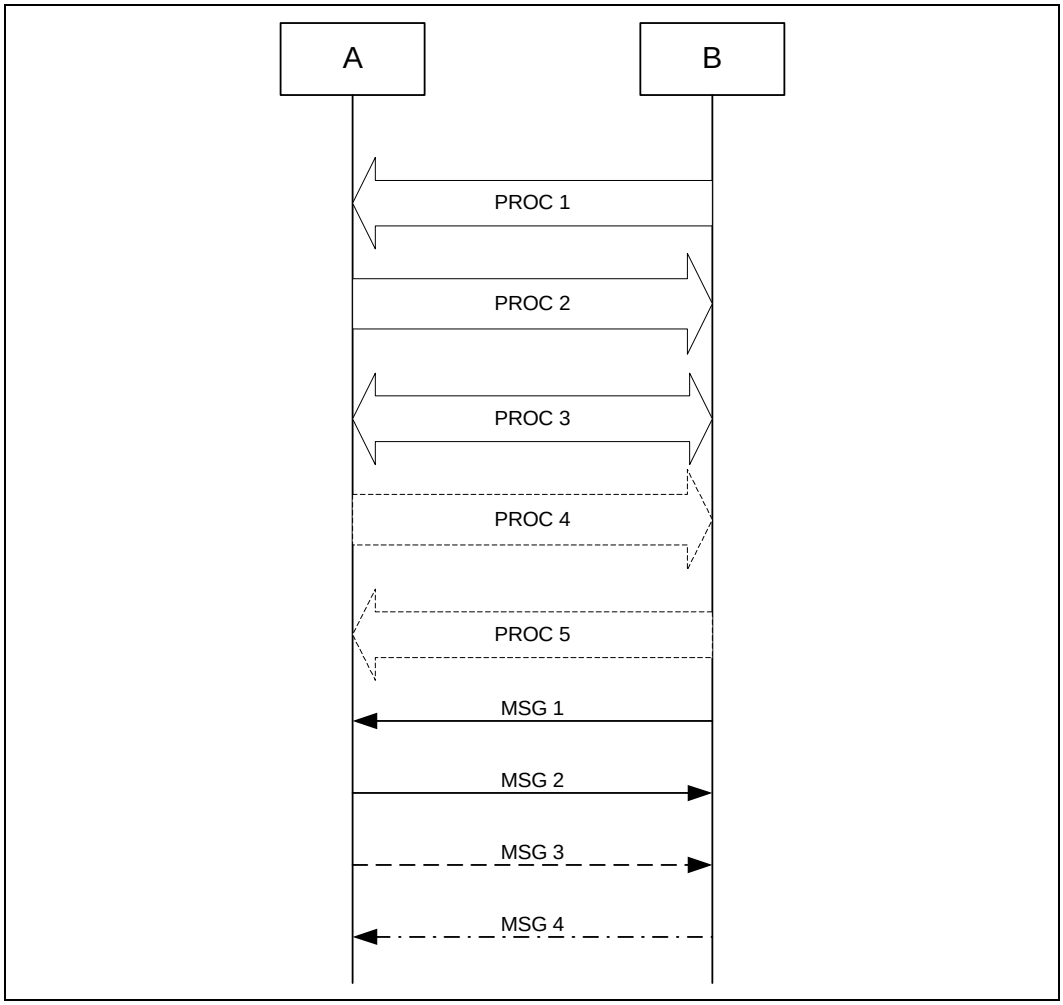


Figure 1.1: Arrows used in signalling diagrams

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 or B). Dashed arrows denote optional steps. 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 a conditional message from B to A.

1.2.3 Notation for timers and counters

Timers are introduced specific to this profile. To distinguish them from timers used in the Bluetooth protocol specifications and other profiles, these timers are named in the following format: 'T_{GAP}*(nnn)*'.

2 PROFILE OVERVIEW

2.1 PROFILE STACK

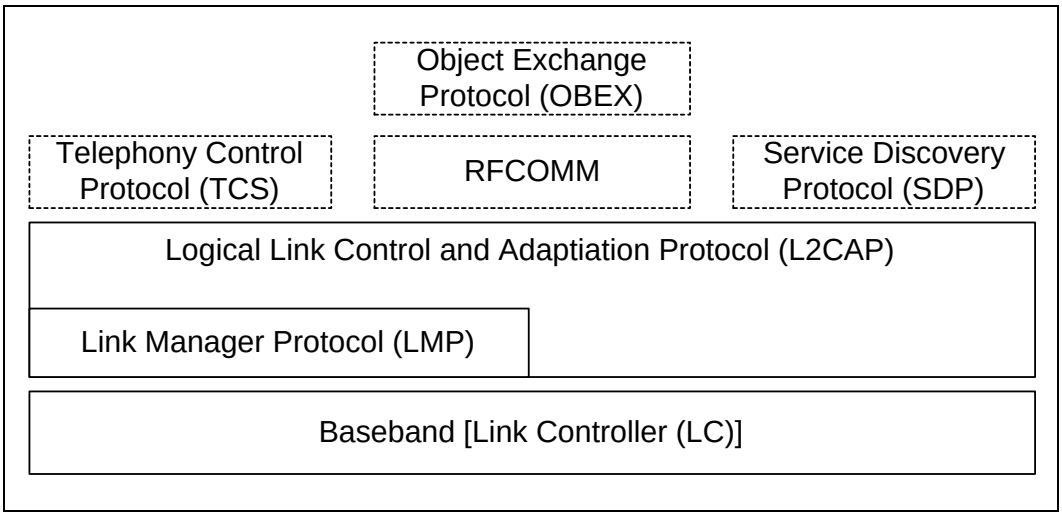


Figure 2.1: Profile stack covered by this profile.

The main purpose of this profile is to describe the use of the lower layers of the Bluetooth protocol stack (LC and LMP). To describe security related alternatives, also higher layers (L2CAP, RFCOMM and OBEX) are included.

2.2 CONFIGURATIONS AND ROLES

For the descriptions in this profile of the roles that the two devices involved in a Bluetooth communication can take, the generic notation of the A-party (the *paging device* in case of link establishment, or *initiator* in case of another procedure on an established link) and the B-party (*paged device* or *acceptor*) is used. The A-party is the one that, for a given procedure, initiates the establishment of the physical link or initiates a transaction on an existing link.

This profile handles the procedures between two devices related to discovery and connecting (link and connection establishment) for the case where none of the two devices has any link established as well as the case where (at least) one device has a link established (possibly to a third device) before starting the described procedure.

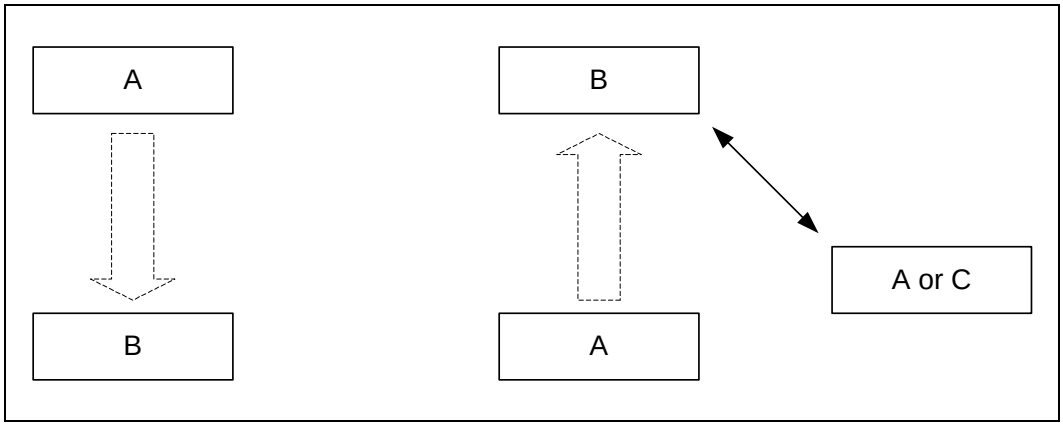


Figure 2.2: This profile covers procedures initiated by one device (A) towards another device (B) that may or may not have an existing Bluetooth link active.

The initiator and the acceptor generally operate the generic procedures according to this profile or another profile referring to this profile. If the acceptor operates according to several profiles simultaneously, this profile describes generic mechanisms for how to handle this.

2.3 USER REQUIREMENTS AND SCENARIOS

The Bluetooth user should in principle be able to connect a Bluetooth device to any other Bluetooth device. Even if the two connected devices don't share any common application, it should be possible for the user to find this out using basic Bluetooth capabilities. When the two devices do share the same application but are from different manufacturers, the ability to connect them should not be blocked just because manufacturers choose to call basic Bluetooth capabilities by different names on the user interface level or implement basic procedures to be executed in different orders.

2.4 PROFILE FUNDAMENTALS

This profile states the requirements on names, values and coding schemes used for names of parameters and procedures experienced on the user interface level.

This profile defines modes of operation that are not service- or profile-specific, but that are generic and can be used by profiles referring to this profile, and by devices implementing multiple profiles.

This profile defines the general procedures that can be used for discovering identities, names and basic capabilities of other Bluetooth devices that are in a mode where they can be discoverable. Only procedures where no channel or connection establishment is used are specified.

This profile defines the general procedure for how to create bonds (i.e. dedicated exchange of link keys) between Bluetooth devices.

This profile describes the general procedures that can be used for establishing connections to other Bluetooth devices that are in mode that allows them to accept connections and service requests.

2.5 CONFORMANCE

Bluetooth devices that do not conform to any other Bluetooth profile shall conform to this profile to ensure basic interoperability and co-existence.

Bluetooth devices that conform to another Bluetooth profile may use adaptations of the generic procedures as specified by that other profile. They shall, however, be compatible with devices compliant to this profile at least on the level of the supported generic procedures.

If conformance to this profile is claimed, all capabilities indicated 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 certification program.

3 USER INTERFACE ASPECTS

3.1 THE USER INTERFACE LEVEL

In the context of this specification, the user interface level refers to places (such as displays, dialog boxes, manuals, packaging, advertising, etc.) where users of Bluetooth devices encounters names, values and numerical representation of Bluetooth terminology and parameters.

This profile specifies the generic terms that should be used on the user interface level. These terms should be translated into languages supported by the Bluetooth device according to tables provided by the Bluetooth SIG.

3.2 REPRESENTATION OF BLUETOOTH PARAMETERS

3.2.1 Bluetooth device address (BD_ADDR)

3.2.1.1 Definition

BD_ADDR is the unique address of a Bluetooth device as defined in [1]. It is received from a remote device during the device discovery procedure.

3.2.1.2 Term on user interface level

When the Bluetooth address is referred to on UI level, the term ’Bluetooth Device Address’ should be used.

3.2.1.3 Representation

On BB level the BD_ADDR is represented as 48 bits [1].

On the UI level the Bluetooth address shall be represented as 12 hexadecimal characters, possibly divided into sub-parts separated by’:’.
(E.g., ’000C3E3A4B69’ or ’00:0C:3E:3A:4B:69’.) At UI level, any number shall have the MSB -> LSB (from left to right) ’natural’ ordering (e.g., the number ’16’ shall be shown as ’0x10’).

3.2.2 Bluetooth device name (the user-friendly name)

3.2.2.1 Definition

The Bluetooth device name is the user-friendly name that a Bluetooth device presents itself with. It is a character string returned in LMP_name_res as response to a LMP_name_req.

3.2.2.2 Term on user interface level

When the Bluetooth device name is referred to on UI level, the term 'Bluetooth Device Name' should be used.

3.2.2.3 Representation

The Bluetooth device name can be up to 248 bytes maximum according to [2]. It shall be coded according to Unicode UTF-8 (i.e. name entered on UI level may be down to 82 characters if UCS-2 is used).

A device can not expect that a general remote device is able to handle more than the first 40 characters of the Bluetooth device name. If a remote device has limited display capabilities, it may use only the first 20 characters.

3.2.3 Bluetooth passkey (Bluetooth PIN)

3.2.3.1 Definition

The Bluetooth PIN is used to authenticate two Bluetooth devices (that have not previously exchanged link keys) to each other and create a trusted relationship between them. The PIN is used in the pairing procedure (see Section 10.2) to generate the initial link key that is used for further authentication.

The PIN may be entered on UI level but may also be stored in the device; e.g. in the case of a device without sufficient MMI for entering and displaying digits.

3.2.3.2 Terms at user interface level

When the Bluetooth PIN is referred to on UI level, the term 'Bluetooth Passkey' should be used.

3.2.3.3 Representation

The Bluetooth PIN has different representations on different level. PIN_{BB} is used on baseband level, and PIN_{UI} is used on user interface level.

PIN_{BB} is the PIN used by [1] for calculating the initialization key during the pairing procedure. PIN_{UI} is the character representation of the PIN that is entered on UI level. The transformation between PIN_{BB} and PIN_{UI} shall be according to Unicode UTF-8.

According to [1], PIN_{BB} can be 128 bits (16 bytes). When PIN is entered on UI level (PIN_{UI}), it is to be coded into PIN_{BB} according to Unicode UTF-8 (i.e. if a

device supports entry of characters outside the Unicode range 0x00 - 0x7F, the maximum number of characters in the PIN_{UI} may be less than 16).

Examples:

User-entered code	Corresponding PIN_{BB}[0..length-1] (value as a sequence of octets in hexadecimal notation)
’0123’	length = 4, value = 0x30 0x31 0x32 0x33
’Ärlich’	length = 7, value = 0xC3 0x84 0x72 0x6C 0x69 0x63 0x68

All Bluetooth devices that support the bonding procedure and support PIN handling on UI level shall support UI level handling of PINs consisting of decimal digits. In addition, devices may support UI level handling of PINs consisting of general characters.

If a device has a fixed PIN (i.e. PIN is stored in the device and cannot be entered on UI level during pairing), the PIN shall be defined using decimal digits. A device that is expected to pair with a remote device that has restricted UI capabilities should ensure that the PIN can be entered on UI level as decimal digits.

3.2.4 Class of Device

3.2.4.1 Definition

Class of device is a parameter received during the device discovery procedure, indicating the type of device and which types of service that are supported.

3.2.4.2 Term on user interface level

The information within the Class of Device parameter should be referred to as ’Bluetooth Device Class’ (i.e. the major and minor device class fields) and ’Bluetooth Service Type’ (i.e. the service class field). The terms for the defined Bluetooth Device Types and Bluetooth Service Types are defined in [11].

When using a mix of information found in the Bluetooth Device Class and the Bluetooth Service Type, the term ’Bluetooth Device Type’ should be used.

3.2.4.3 Representation

The Class of device is a bit field and is defined in [11]. The UI-level representation of the information in the Class of device is implementation specific.

3.3 PAIRING

Two procedures are defined that make use of the pairing procedure defined on LMP level (LMP-pairing, see Section 10.2). Either the user initiates the bonding procedure and enters the passkey with the explicit purpose of creating a bond (and maybe also a secure relationship) between two Bluetooth devices, or the user is requested to enter the passkey during the establishment procedure since the devices did not share a common link key beforehand. In the first case, the user is said to perform ’bonding (with entering of passkey)’ and in the second case the user is said to ’authenticate using the passkey’.

4 MODES

	Procedure	Ref.	Support
1	Discoverability modes	4.1	
	Non-discoverable mode		C1
	Limited discoverable mode		C2
	General discoverable mode		C2
2	Connectability modes	4.1.3.3	
	Non-connectable mode		O
	Connectable mode		M
3	Pairing modes	4.2.2.2	
	Non-pairable mode		O
	Pairable mode		C3
C1: If limited discoverable mode is supported, non-discoverable mode is mandatory, otherwise optional.			
C2: A Bluetooth device shall support at least one discoverable mode (limited or/and general).			
C3: If the bonding procedure is supported, support for pairable mode is mandatory, otherwise optional.			

Table 4.1: Conformance requirements related to modes defined in this section

4.1 DISCOVERABILITY MODES

With respect to inquiry, a Bluetooth device shall be either in non-discoverable mode or in a discoverable mode. (The device shall be in one, and only one, discoverability mode at a time.) The two discoverable modes defined here are called limited discoverable mode and general discoverable mode. Inquiry is defined in [1].

When a Bluetooth device is in non-discoverable mode it does not respond to inquiry.

A Bluetooth device is said to be made discoverable, or set into a discoverable mode, when it is in limited discoverable mode or in general discoverable mode. Even when a Bluetooth device is made discoverable it may be unable to respond to inquiry due to other baseband activity [1]. A Bluetooth device that does not respond to inquiry for any of these two reasons is called a silent device.

After being made discoverable, the Bluetooth device shall be discoverable for at least $T_{GAP}(103)$.

4.1.1 Non-discoverable mode

4.1.1.1 Definition

When a Bluetooth device is in non-discoverable mode, it shall never enter the INQUIRY_RESPONSE state.

4.1.1.2 Term on UI-level

Bluetooth device is ’non-discoverable’ or in ’non-discoverable mode’.

4.1.2 Limited discoverable mode

4.1.2.1 Definition

The limited discoverable mode should be used by devices that need to be discoverable only for a limited period of time, during temporary conditions or for a specific event. The purpose is to respond to a device that makes a limited inquiry (inquiry using the LIAC).

A Bluetooth device should not be in limited discoverable mode for more than $T_{GAP}(104)$. The scanning for the limited inquiry access code can be done either in parallel or in sequence with the scanning of the general inquiry access code. When in limited discoverable mode, one of the following options shall be used.

4.1.2.1.1 Parallel scanning

When a Bluetooth device is in limited discoverable mode, it shall enter the INQUIRY_SCAN state at least once in $T_{GAP}(102)$ and scan for the GIAC and the LIAC for at least $T_{GAP}(101)$.

4.1.2.1.2 Sequential scanning

When a Bluetooth device is in limited discoverable mode, it shall enter the INQUIRY_SCAN state at least once in $T_{GAP}(102)$ and scan for the GIAC for at least $T_{GAP}(101)$ and enter the INQUIRY_SCAN state more often than once in $T_{GAP}(102)$ and scan for the LIAC for at least $T_{GAP}(101)$.

If an inquiry message is received when in limited discoverable mode, the entry into the INQUIRY_RESPONSE state takes precedence over the next entries into INQUIRY_SCAN state until the inquiry response is completed.

4.1.2.2 Conditions

When a device is in limited discoverable mode it shall set bit no 13 in the Major Service Class part of the Class of Device/Service field [11].

4.1.2.3 Term on UI-level

Bluetooth device is ’discoverable’ or in ’discoverable mode’.

4.1.3 General discoverable mode

4.1.3.1 Definition

The general discoverable mode shall be used by devices that need to be discoverable continuously or for no specific condition. The purpose is to respond to a device that makes a general inquiry (inquiry using the GIAC).

4.1.3.2 Conditions

When a Bluetooth device is in general discoverable mode, it shall enter the INQUIRY_SCAN state more often than once in $T_{GAP}(102)$ and scan for the GIAC for at least $T_{GAP}(101)$.

A device in general discoverable mode shall not respond to a LIAC inquiry.

4.1.3.3 Term on UI-level

Bluetooth device is ’discoverable’ or in ’discoverable mode’.

4.2 CONNECTABILITY MODES

With respect to paging, a Bluetooth device shall be either in non-connectable mode or in connectable mode. Paging is defined in [1].

When a Bluetooth device is in non-connectable mode it does not respond to paging. When a Bluetooth device is in connectable mode it responds to paging.

4.2.1 Non-connectable mode

4.2.1.1 Definition

When a Bluetooth device is in non-connectable mode it shall never enter the PAGE_SCAN state.

4.2.1.2 Term on UI-level

Bluetooth device is ’non-connectable’ or in ’non-connectable mode’.

4.2.2 Connectable mode

4.2.2.1 Definition

When a Bluetooth device is in connectable mode it shall periodically enter the PAGE_SCAN state.

4.2.2.2 Term on UI-level

Bluetooth device is ’connectable’ or in ’connectable mode’.

4.3 PAIRING MODES

With respect to pairing, a Bluetooth device shall be either in non-pairable mode or in pairable mode. In pairable mode the Bluetooth device accepts paring – i.e. creation of bonds – initiated by the remote device, and in non-pairable mode it does not. Pairing is defined in [1] and [2].

4.3.1 Non-pairable mode

4.3.1.1 Definition

When a Bluetooth device is in non-pairable mode it shall respond to a received LMP_in_rand with LMP_not_accepted with the reason *pairing not allowed*.

4.3.1.2 Term on UI-level

Bluetooth device is ’non-bondable’ or in ’non-bondable mode’ or “does not accept bonding”.

4.3.2 Pairable mode

4.3.2.1 Definition

When a Bluetooth device is in pairable mode it shall respond to a received LMP_in_rand with LMP_accepted (or with LMP_in_rand if it has a fixed PIN).

4.3.2.2 Term on UI-level

Bluetooth device is ’bondable’ or in ’bondable mode’ or “accepts bonding”.

5 SECURITY ASPECTS

	Procedure	Ref.	Support
1	Authentication	5.1	C1
2	Security modes	5.2	
	Security mode 1		O
	Security mode 2		C2
	Security mode 3		C2
C1: If security mode 1 is the only security mode that is supported, support for authentication is optional, otherwise mandatory. (Note: support for LMP-authentication and LMP-pairing is mandatory according [2] independent of which security mode that is used.)			
C2: If security mode 1 is not the only security mode that is supported, then support for at least one of security mode 2 or security mode 3 is mandatory.			

Table 5.1: Conformance requirements related to the generic authentication procedure and the security modes defined in this section

5.1 AUTHENTICATION

5.1.1 Purpose

The generic authentication procedure describes how the LMP-authentication and LMP-pairing procedures are used when authentication is initiated by one Bluetooth device towards another, depending on if a link key exists or not and if pairing is allowed or not.

5.1.2 Term on UI level

’Bluetooth authentication’.

5.1.3 Procedure

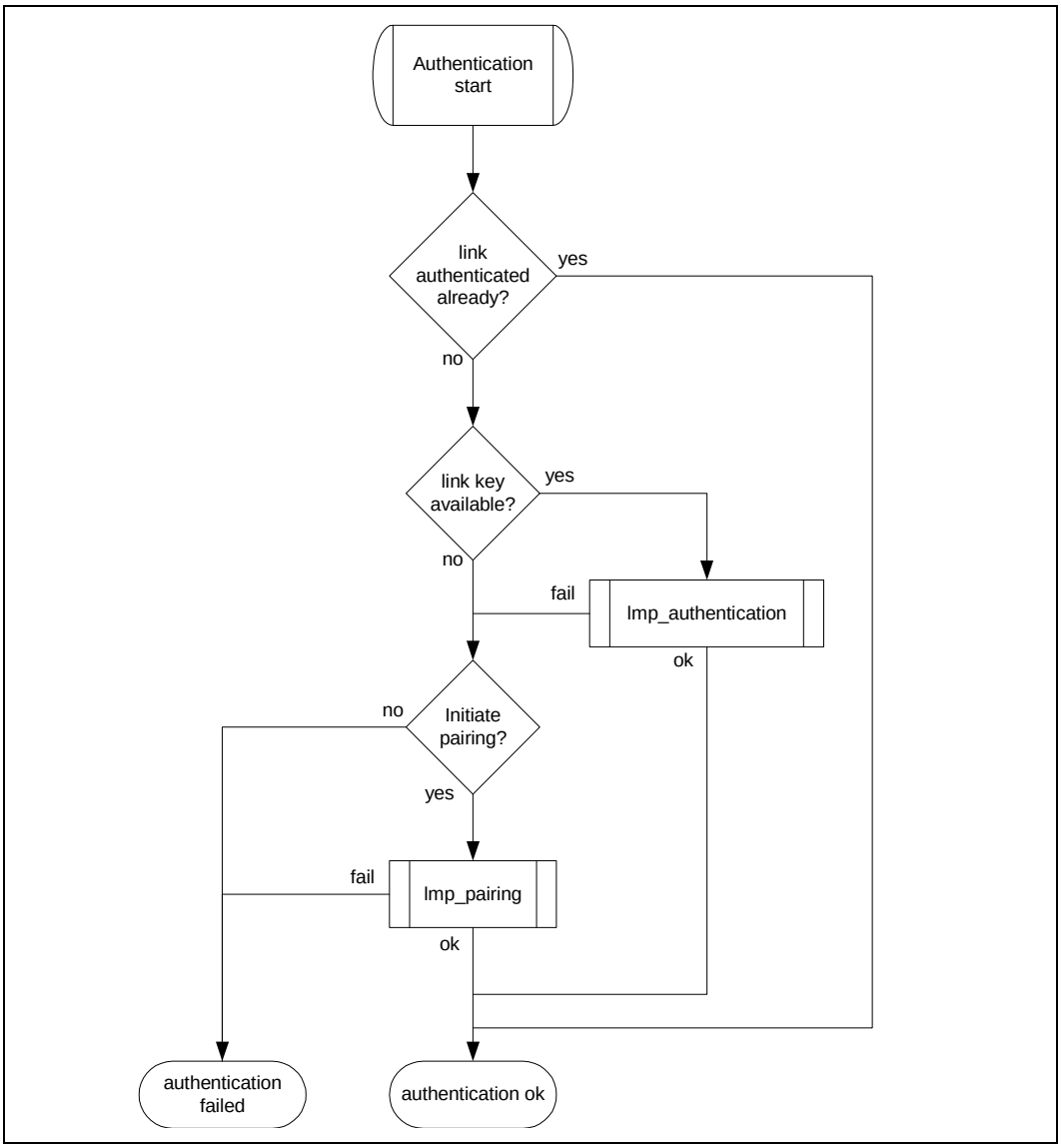


Figure 5.1: Definition of the generic authentication procedure.

5.1.4 Conditions

The device that initiates authentication has to be in security mode 2 or in security mode 3.

5.2 SECURITY MODES

The following flow chart describes where in the channel establishment procedures initiation of authentication takes place, depending on which security mode the Bluetooth device is in.

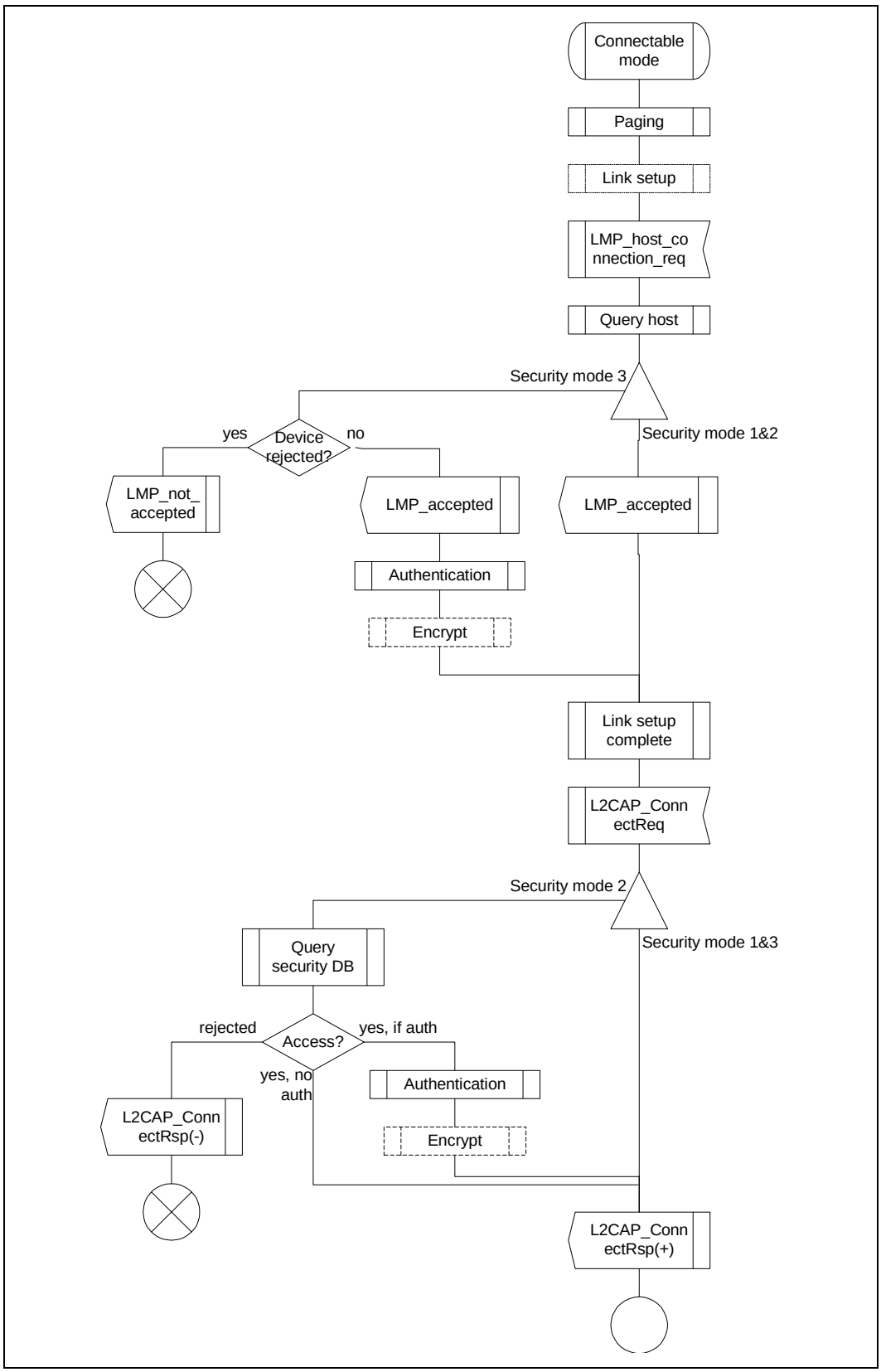


Figure 5.2: Illustration of channel establishment using different security modes.

When authentication is initiated towards a Bluetooth device, it shall act according to [2] and the current pairing mode, independent of which security mode it is in.

5.2.1 Security mode 1 (non-secure)

When a Bluetooth device is in security mode 1 it shall never initiate any security procedure (i.e., it shall never send LMP_au_rand, LMP_in_rand or LMP_encryption_mode_req).

5.2.2 Security mode 2 (service level enforced security)

When a Bluetooth device is in security mode 2 it shall not initiate any security procedure before a channel establishment request (L2CAP_ConnectReq) has been received or a channel establishment procedure has been initiated by itself. (The behavior of a device in security mode 2 is further described in [10].) Whether a security procedure is initiated or not depends on the security requirements of the requested channel or service.

A Bluetooth device in security mode 2 should classify the security requirements of its services using at least the following attributes:

- Authorization required;
- Authentication required;
- Encryption required.

Note: Security mode 1 can be considered (at least from a remote device point of view) as a special case of security mode 2 where no service has registered any security requirements.

5.2.3 Security modes 3 (link level enforced security)

When a Bluetooth device is in security mode 3 it shall initiate security procedures before it sends LMP_link_setup_complete. (The behavior of a device in security mode 3 is as described in [2].)

A Bluetooth device in security mode 3 may reject the host connection request (respond with LMP_not_accepted to the LMP_host_connection_req) based on settings in the host (e.g. only communication with pre-paired devices allowed).

6 IDLE MODE PROCEDURES

The inquiry and discovery procedures described here are applicable only to the device that initiates them (A). The requirements on the behavior of B is according to the modes specified in Section 4 and to [2].

	Procedure	Ref.	Support
1	General inquiry	6.1	C1
2	Limited inquiry	6.2	C1
3	Name discovery	6.3	O
4	Device discovery	6.4	O
5	Bonding	6.5	O
C1: If initiation of bonding is supported, support for at least one inquiry procedure is mandatory, otherwise optional. (Note: support for LMP-pairing is mandatory [2].)			

6.1 GENERAL INQUIRY

6.1.1 Purpose

The purpose of the general inquiry procedure is to provide the initiator with the Bluetooth device address, clock, Class of Device and used page scan mode of general discoverable devices (i.e. devices that are in range with regard to the initiator and are set to scan for inquiry messages with the General Inquiry Access Code). Also devices in limited discoverable mode will be discovered using general inquiry.

The general inquiry should be used by devices that need to discover devices that are made discoverable continuously or for no specific condition.

6.1.2 Term on UI level

’Bluetooth Device Inquiry’.

6.1.3 Description

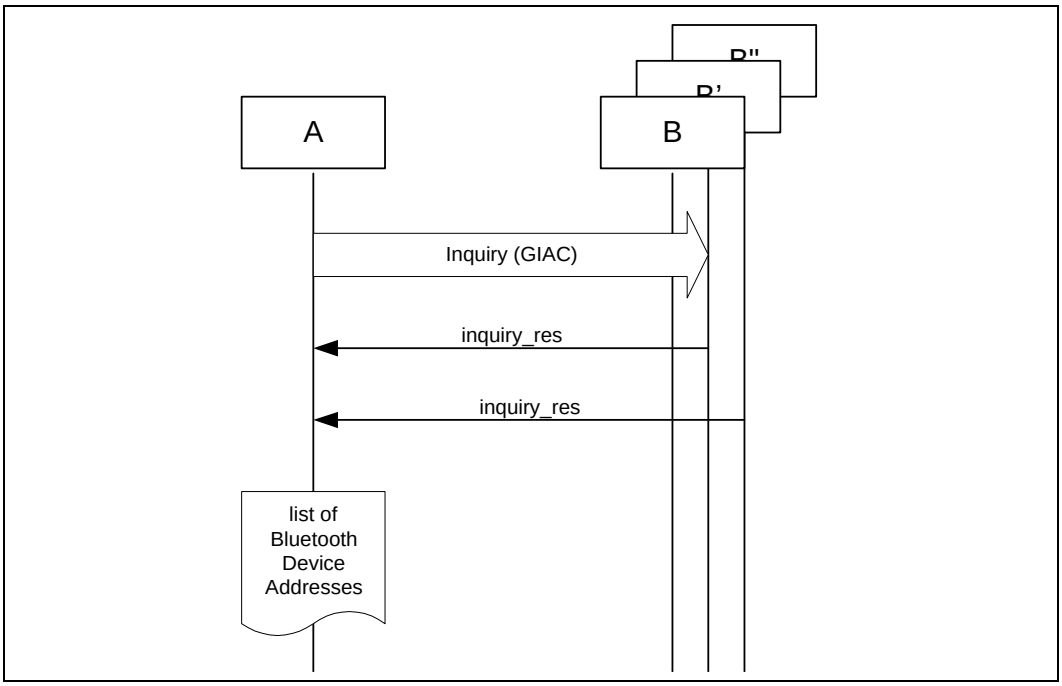


Figure 6.1: General inquiry ,where B is a device in non-discoverable mode, B´ is a device in limited discoverable mode and B" is a device in general discoverable mode. (Note that all discoverable devices are discovered using general inquiry, independent of which discoverable mode they are in.)

6.1.4 Conditions

When general inquiry is initiated by a Bluetooth device, it shall be in the INQUIRY state for at least $T_{GAP}(100)$ and perform inquiry using the GIAC.

In order to receive inquiry response, the remote devices in range have to be made discoverable (limited or general).

6.2 LIMITED INQUIRY

6.2.1 Purpose

The purpose of the limited inquiry procedure is to provide the initiator with the Bluetooth device address, clock, Class of Device and used page scan mode of limited discoverable devices. The latter devices are devices that are in range with regard to the initiator, and may be set to scan for inquiry messages with the Limited Inquiry Access Code, in addition to scanning for inquiry messages with the General Inquiry Access Code.

The limited inquiry should be used by devices that need to discover devices that are made discoverable only for a limited period of time, during temporary conditions or for a specific event. Since it is not guaranteed that the

discoverable device scans for the LIAC, the initiating device may choose any inquiry procedure (general or limited). Even if the remote device that is to be discovered is expected to be made limited discoverable (e.g. when a dedicated bonding is to be performed), the limited inquiry should be done in sequence with a general inquiry in such a way that both inquiries are completed within the time the remote device is limited discoverable, i.e. at least T_{GAP}(103).

6.2.2 Term on UI level

'Bluetooth Device Inquiry'.

6.2.3 Description

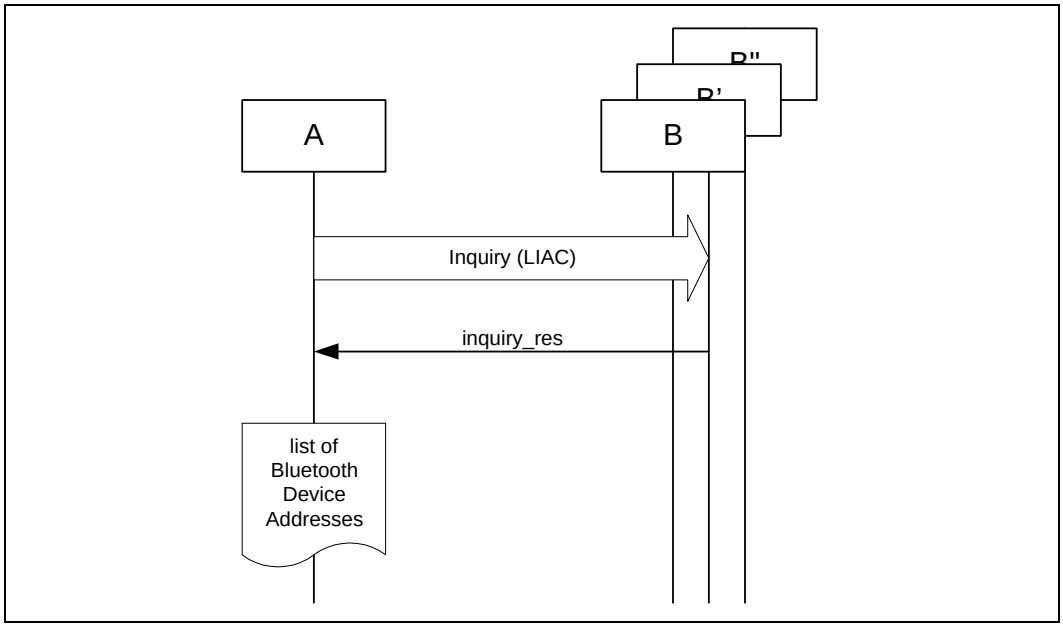


Figure 6.2: Limited inquiry where B is a device in non-discoverable mode, B' is a device in limited discoverable mode and B" is a device in general discoverable mode. (Note that only limited discoverable devices can be discovered using limited inquiry.)

6.2.4 Conditions

When limited inquiry is initiated by a Bluetooth device, it shall be in the INQUIRY state for at least T_{GAP}(100) and perform inquiry using the LIAC.

In order to receive inquiry response, the remote devices in range has to be made limited discoverable.

6.3 NAME DISCOVERY

6.3.1 Purpose

The purpose of name discovery is to provide the initiator with the Bluetooth Device Name of connectable devices (i.e. devices in range that will respond to paging).

6.3.2 Term on UI level

'Bluetooth Device Name Discovery'.

6.3.3 Description

6.3.3.1 Name request

Name request is the procedure for retrieving the Bluetooth Device Name from a connectable Bluetooth device. It is not necessary to perform the full link establishment procedure (see Section 7.1) in order to just to get the name of another device.

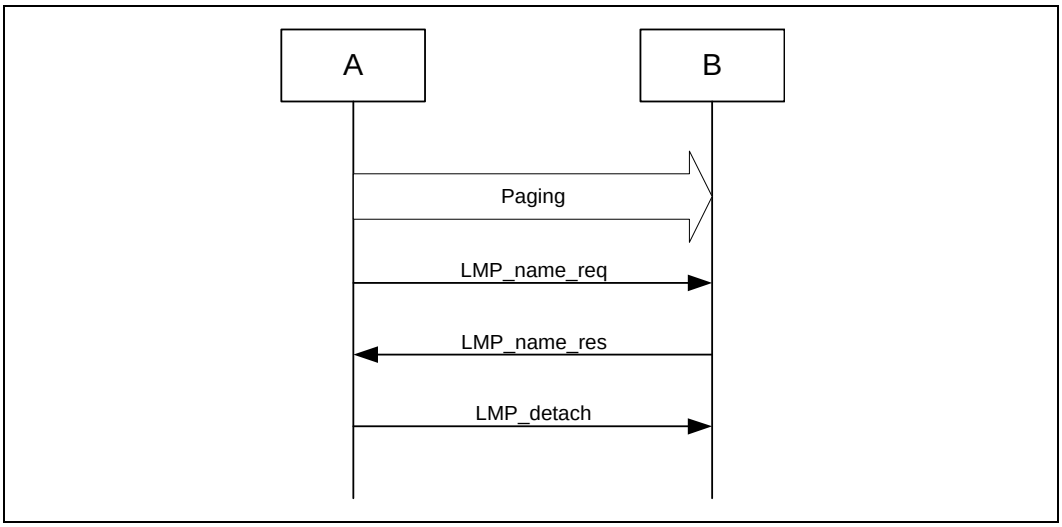


Figure 6.3: Name request procedure.

6.3.3.2 Name discovery

Name discovery is the procedure for retrieving the Bluetooth Device Name from connectable Bluetooth devices by performing name request towards known devices (i.e. Bluetooth devices for which the Bluetooth Device Addresses are available).

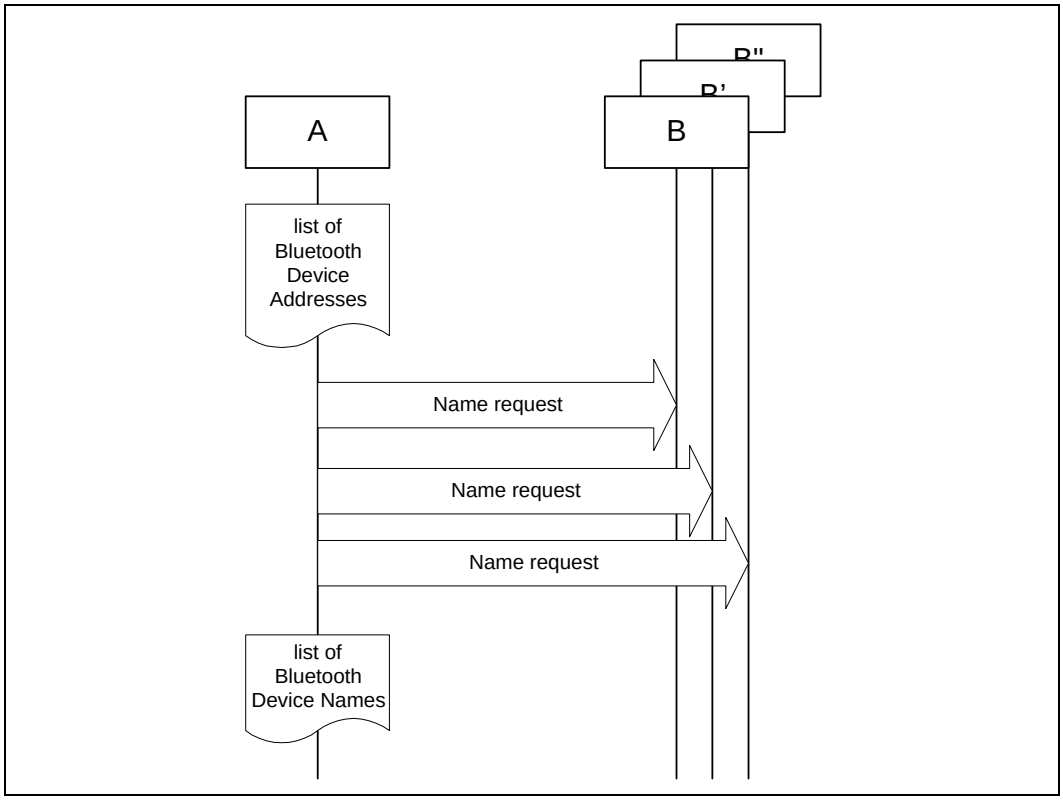


Figure 6.4: Name discovery procedure.

6.3.4 Conditions

In the name request procedure, the initiator will use the Device Access Code of the remote device as retrieved immediately beforehand – normally through an inquiry procedure.

6.4 DEVICE DISCOVERY

6.4.1 Purpose

The purpose of device discovery is to provide the initiator with the Bluetooth Address, clock, Class of Device, used page scan mode and Bluetooth device name of discoverable devices.

6.4.2 Term on UI level

’Bluetooth Device Discovery’.

6.4.3 Description

During the device discovery procedure, first an inquiry (either general or limited) is performed, and then name discovery is done towards some or all of the devices that responded to the inquiry.

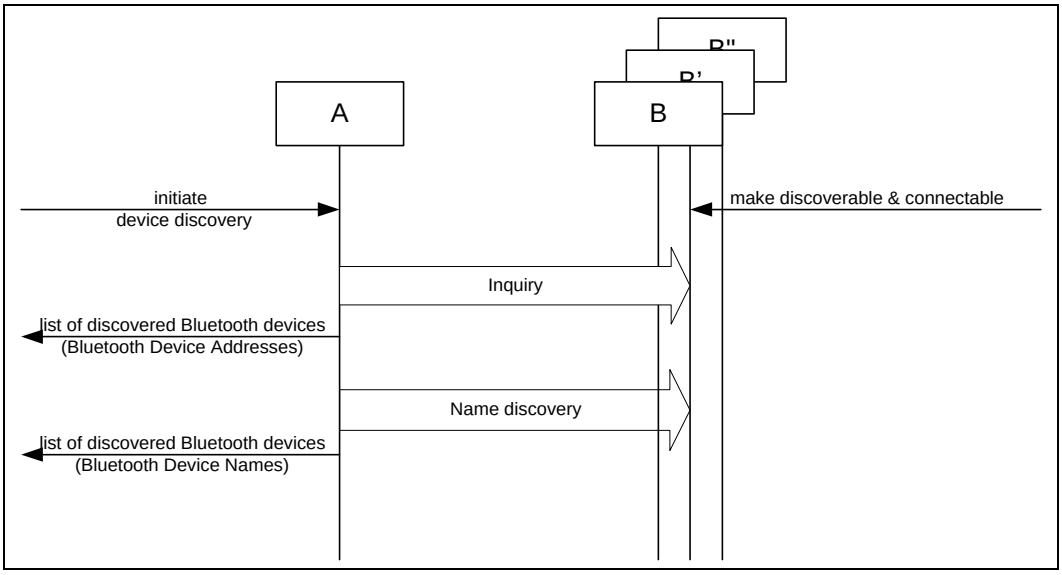


Figure 6.5: Device discovery procedure.

6.4.4 Conditions

Conditions for both inquiry (general or limited) and name discovery must be fulfilled (i.e. devices discovered during device discovery must be both discoverable and connectable).

6.5 BONDING

6.5.1 Purpose

The purpose of bonding is to create a relation between two Bluetooth devices based on a common link key (a bond). The link key is created and exchanged (pairing) during the bonding procedure and is expected to be stored by both Bluetooth devices, to be used for future authentication.

In addition to pairing, the bonding procedure can involve higher layer initialization procedures.

6.5.2 Term on UI level

’Bluetooth Bonding’

6.5.3 Description

Two aspects of the bonding procedure are described here. Dedicated bonding is what is done when the two devices are explicitly set to perform only a creation and exchange of a common link key.

General bonding is included to indicate that the framework for the dedicated bonding procedure is the same as found in the normal channel and connection establishment procedures. This means that pairing may be performed successfully if A has initiated bonding while B is in its normal connectable and security modes.

The main difference with bonding, as compared to a pairing done during link or channel establishment, is that for bonding it is the paging device (A) that must initiate the authentication.

6.5.3.1 General bonding

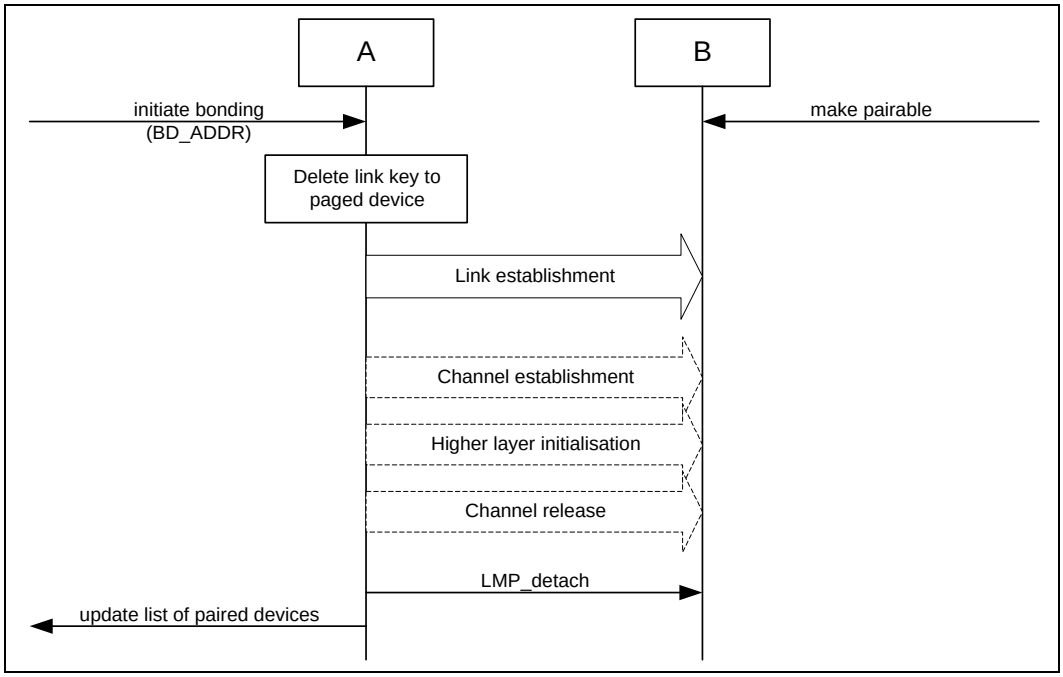


Figure 6.6: General description of bonding as being the link establishment procedure executed under specific conditions on both devices, followed by an optional higher layer initalization process.

6.5.3.2 Dedicated bonding

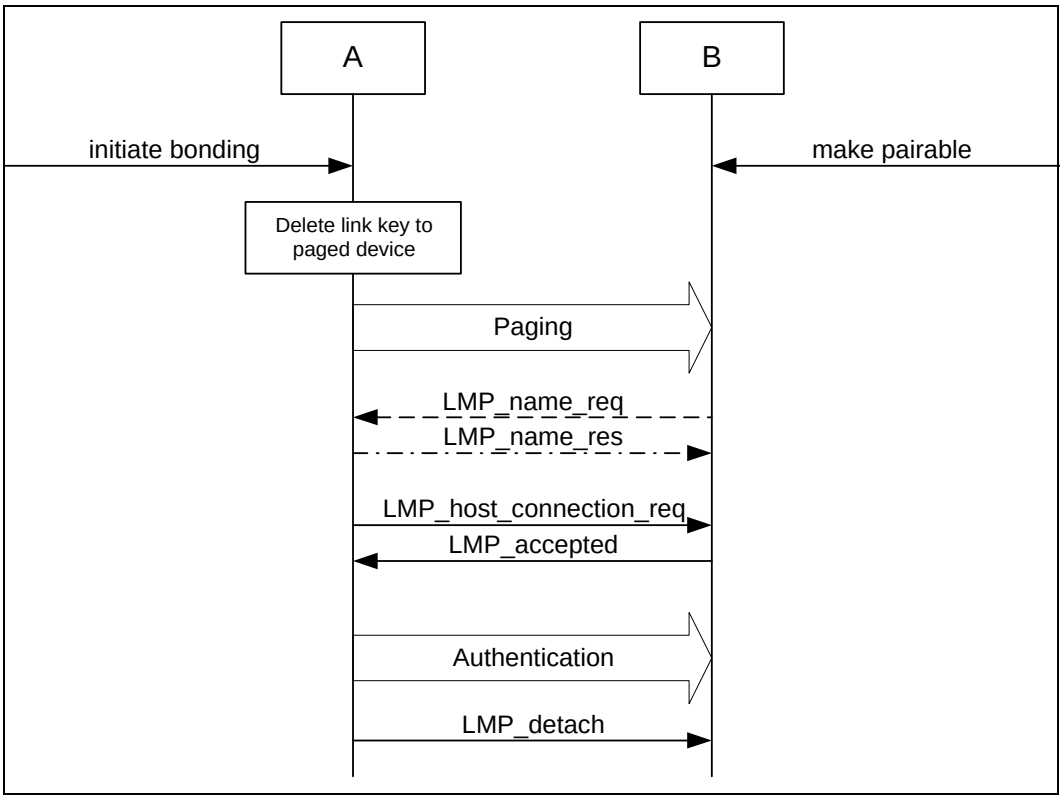


Figure 6.7: Bonding as performed when the purpose of the procedure is only to create and exchange a link key between two Bluetooth devices.

6.5.4 Conditions

Before bonding can be initiated, the initiating device (A) must know the Device Access Code of the device to pair with. This is normally done by first performing device discovery. A Bluetooth Device that can initiate bonding (A) should use limited inquiry, and a Bluetooth Device that accepts bonding (B) should support the limited discoverable mode.

Bonding is in principle the same as link establishment with the conditions:

- The paged device (B) shall be set into pairable mode. The paging device (A) is assumed to allow pairing since it has initiated the bonding procedure.
- The paging device (the initiator of the bonding procedure, A) shall initiate authentication.
- Before initiating the authentication part of the bonding procedure, the paging device should delete any link key corresponding to a previous bonding with the paged device.
- If the paging device does not intend to initiate any higher layer initialization during bonding, it need not send LMP_host_request before initiating authentication.

7 ESTABLISHMENT PROCEDURES

	Procedure	Ref.	Support in A	Support in B
1	Link establishment	7.1	M	M
2	Channel establishment	7.2	O	M
3	Connection establishment	7.3	O	O

Table 7.1: Establishment procedures

The establishment procedures defined here do not include any discovery part. Before establishment procedures are initiated, the information provided during device discovery (in the FHS packet of the inquiry response or in the response to a name request) has to be available in the initiating device. This information is:

- The Bluetooth Device Address (BD_ADDR) from which the Device Access Code is generated;
- The system clock of the remote device;
- The page scan mode used by the remote device.

Additional information provided during device discovery that is useful for making the decision to initiate an establishment procedure is:

- The Class of device;
- The Device name.

7.1 LINK ESTABLISHMENT

7.1.1 Purpose

The purpose of the link establishment procedure is to establish a physical link (of ACL type) between two Bluetooth devices using procedures from [1] and [2].

7.1.2 Term on UI level

’Bluetooth link establishment’

7.1.3 Description

In this sub-section, the paging device (A) is in security mode 3. The paging device cannot during link establishment distinguish if the paged device (B) is in security mode 1 or 2.

7.1.3.1 B in security mode 1 or 2

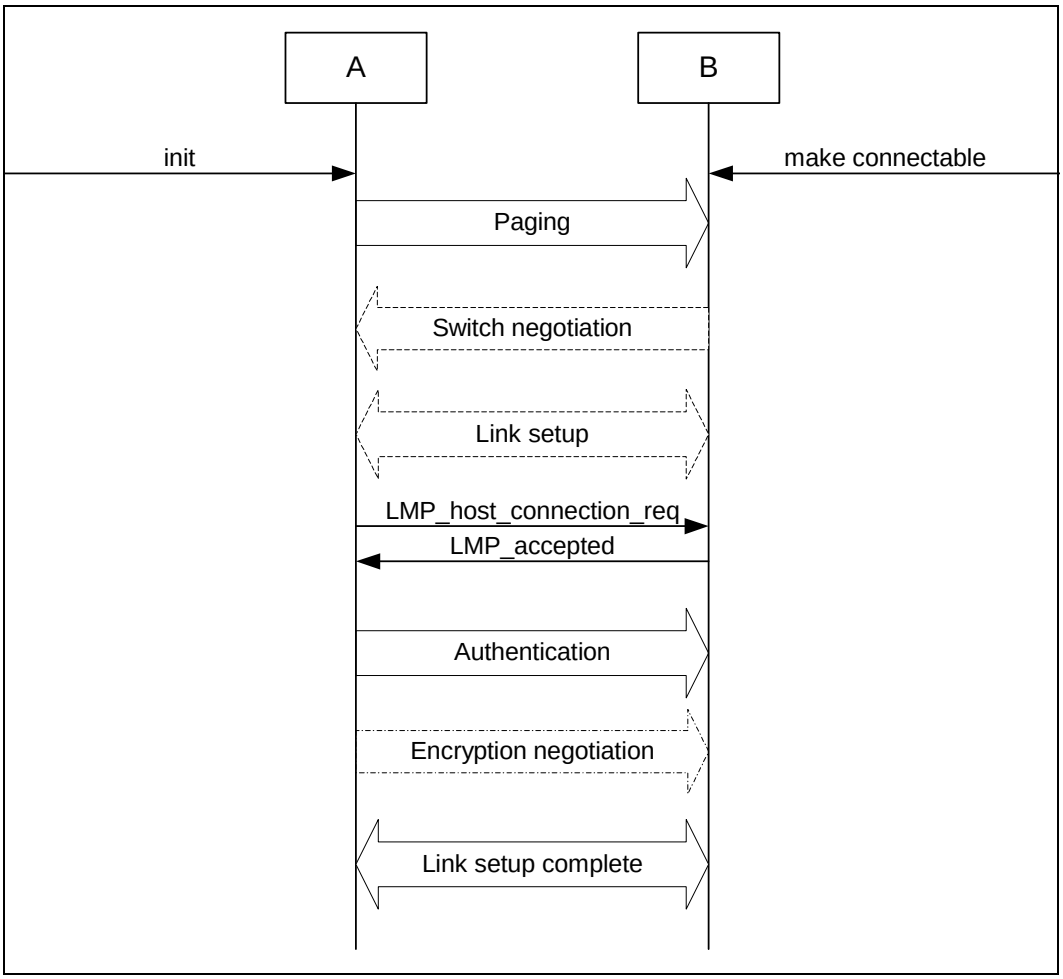


Figure 7.1: Link establishment procedure when the paging device (A) is in security mode 3 and the paged device (B) is in security mode 1 or 2.

7.1.3.2 B in security mode 3

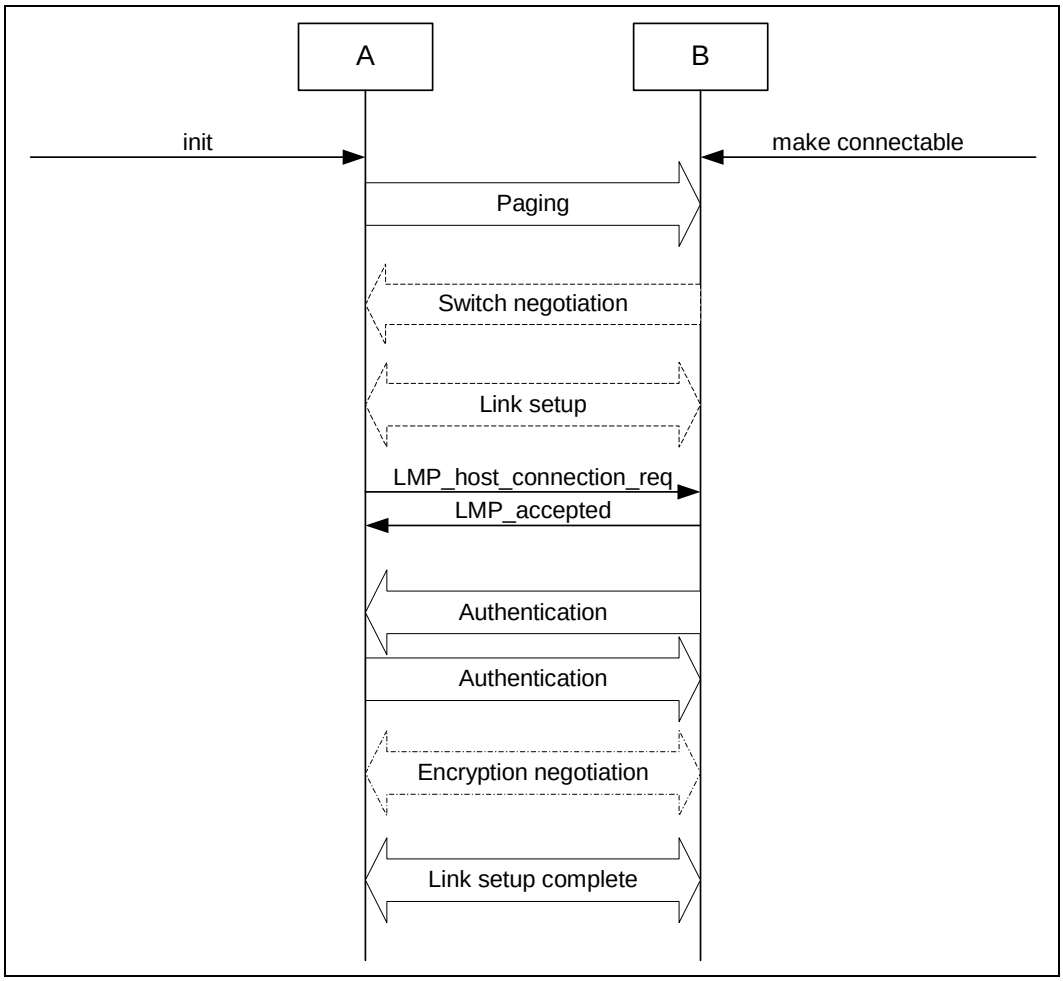


Figure 7.2: Link establishment procedure when both the paging device (A) and the paged device (B) are in security mode 3.

7.1.4 Conditions

The paging procedure shall be according to [1] and the paging device should use the Device access code and page mode received through a previous inquiry. When paging is completed, a physical link between the two Bluetooth devices is established.

If role switching is needed (normally it is the paged device that has an interest in changing the master/slave roles) it should be done as early as possible after the physical link is established. If the paging device does not accept the switch, the paged device has to consider whether to keep the physical link or not.

Both devices may perform link setup (using LMP procedures that require no interaction with the host on the remote side). Optional LMP features can be used after having confirmed (using LMP_feature_req) that the other device supports the feature.

When the paging device needs to go beyond the link setup phase, it issues a request to be connected to the host of the remote device. If the paged device is in security mode 3, this is the trigger for initiating authentication.

The paging device shall send LMP_host_connection_req during link establishment (i.e. before channel establishment) and may initiate authentication only after having sent LMP_host_connection_request.

After an authentication has been performed, any of the devices can initiate encryption.

Further link configuration may take place after the LMP_host_connection_req. When both devices are satisfied, they send LMP_setup_complete.

Link establishment is completed when both devices have sent LMP_setup_complete.

7.2 CHANNEL ESTABLISHMENT

7.2.1 Purpose

The purpose of the channel establishment procedure is to establish a Bluetooth channel (a logical link) between two Bluetooth devices using [3].

7.2.2 Term on UI level

’Bluetooth channel establishment’.

7.2.3 Description

In this sub-section, the initiator (A) is in security mode 3. During channel establishment, the initiator cannot distinguish if the acceptor (B) is in security mode 1 or 3.

7.2.3.1 B in security mode 2

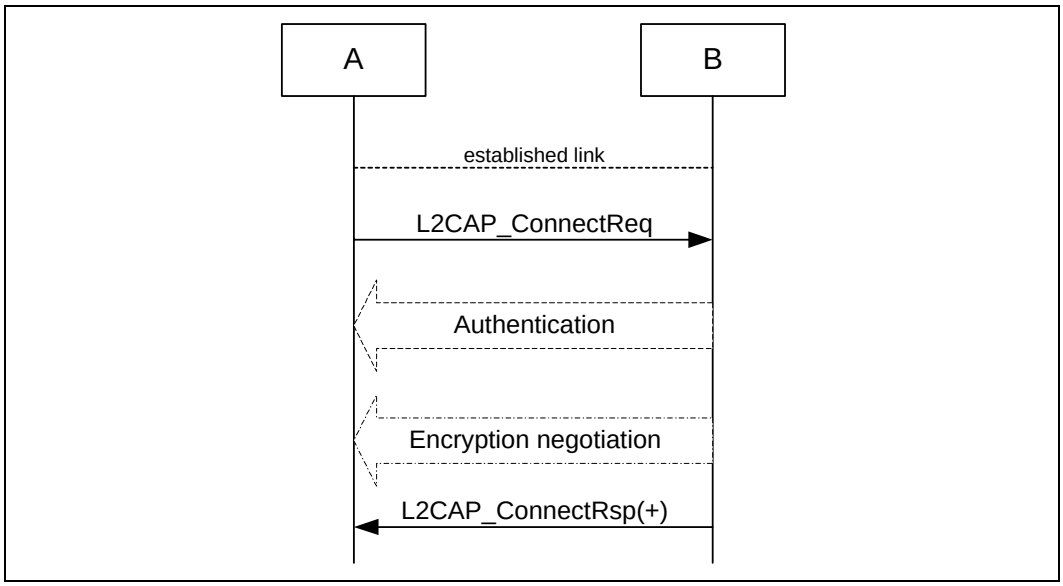


Figure 7.3: Channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2.

7.2.3.2 B in security mode 1 or 3

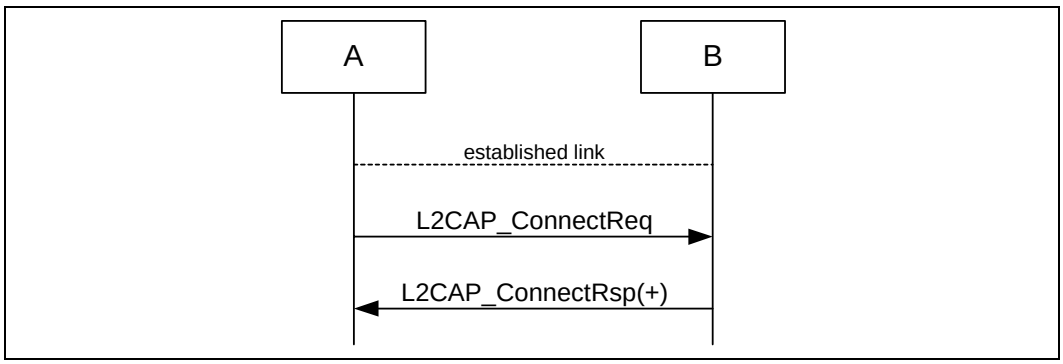


Figure 7.4: Channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3.

7.2.4 Conditions

Channel establishment starts after link establishment is completed when the initiator sends a channel establishment request (L2CAP_ConnectReq).

Depending on security mode, security procedures may take place after the channel establishment has been initiated.

Channel establishment is completed when the acceptor responds to the channel establishment request (with a positive L2CAP_ConnectRsp).

7.3 CONNECTION ESTABLISHMENT

7.3.1 Purpose

The purpose of the connection establishment procedure is to establish a connection between applications on two Bluetooth devices.

7.3.2 Term on UI level

'Bluetooth connection establishment'

7.3.3 Description

In this sub-section, the initiator (A) is in security mode 3. During connection establishment, the initiator cannot distinguish if the acceptor (B) is in security mode 1 or 3.

7.3.3.1 B in security mode 2

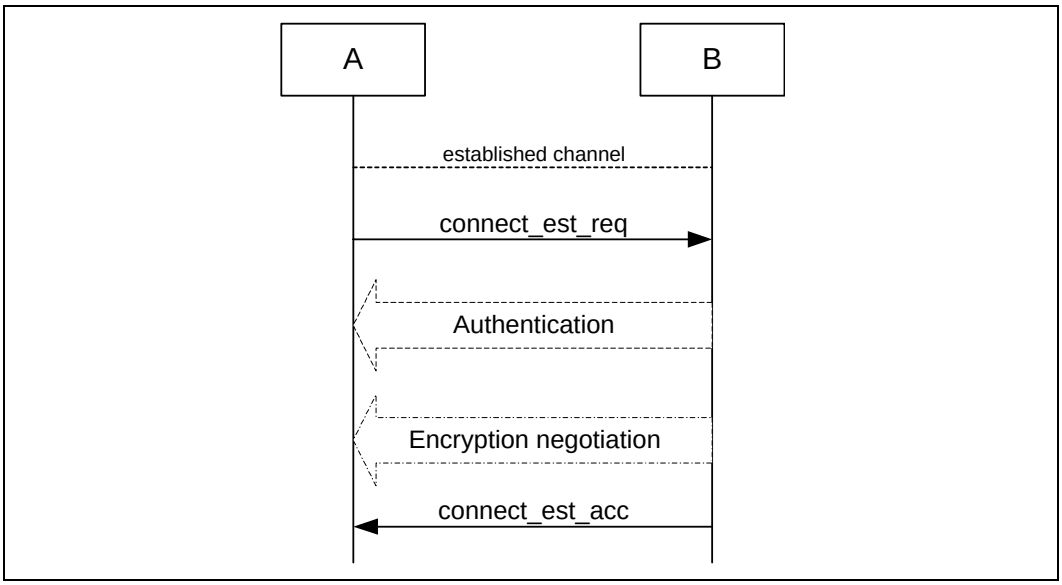


Figure 7.5: Connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2.

7.3.3.2 B in security mode 1 or 3

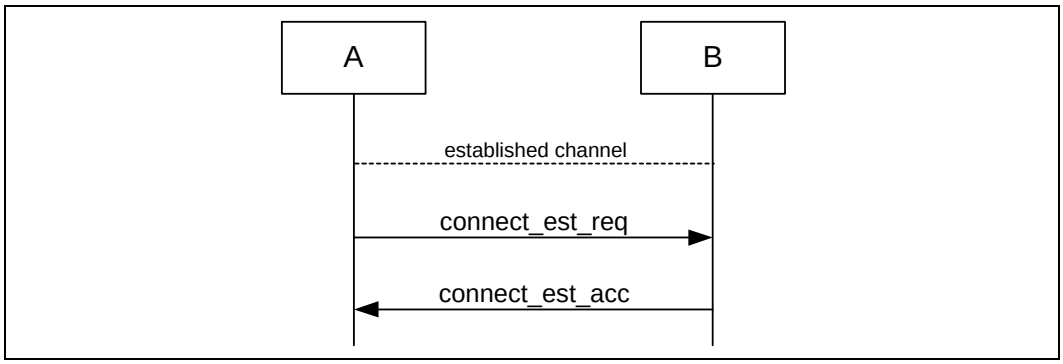


Figure 7.6: Connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3.

7.3.4 Conditions

Connection establishment starts after channel establishment is completed, when the initiator sends a connection establishment request ('connect_est_req' is application protocol-dependent). This request may be a TCS SETUP message [5] in the case of a Bluetooth telephony application Cordless Telephony Profile, or initialization of RFCOMM and establishment of DLC [4] in the case of a serial port-based application Serial Port Profile (although neither TCS or RFCOMM use the term 'connection' for this).

Connection establishment is completed when the acceptor accepts the connection establishment request ('connect_est_acc' is application protocol dependent).

7.4 ESTABLISHMENT OF ADDITIONAL CONNECTION

When a Bluetooth device has established one connection with another Bluetooth device, it may be available for establishment of:

- A second connection on the same channel, and/or
- A second channel on the same link, and/or
- A second physical link.

If the new establishment procedure is to be towards the same device, the security part of the establishment depends on the security modes used. If the new establishment procedure is to be towards a new remote device, the device should behave according to active modes independent of the fact that it already has another physical link established (unless allowed co-incident radio and baseband events have to be handled).

8 DEFINITIONS

In the following, terms written with capital letters refer to states.

8.1 GENERAL DEFINITIONS

Mode A set of directives that defines how a device will respond to certain events.

Idle As seen from a remote device, a Bluetooth device is idle, or is in idle mode, when there is no link established between them.

Bond A relation between two Bluetooth devices defined by creating, exchanging and storing a common link key. The bond is created through the bonding or LMP-pairing procedures.

8.2 CONNECTION-RELATED DEFINITIONS

Physical channel A synchronized Bluetooth baseband-compliant RF hopping sequence.

Piconet A set of Bluetooth devices sharing the same physical channel defined by the master parameters (clock and BD_ADDR).

Physical link A Baseband-level connection[1] between two devices established using paging. A physical link comprises a sequence of transmission slots on a physical channel alternating between master and slave transmission slots.

ACL link An asynchronous (packet-switched) connection[1] between two devices created on LMP level. Traffic on an ACL link uses ACL packets to be transmitted.

SCO link A synchronous (circuit-switched) connection[1] for reserved bandwidth communications; e.g. voice between two devices, created on the LMP level by reserving slots periodically on a physical channel. Traffic on an SCO link uses SCO packets to be transmitted. SCO links can be established only after an ACL link has first been established.

Link Shorthand for an ACL link.

PAGE A baseband state where a device transmits page trains, and processes any eventual responses to the page trains.

PAGE_SCAN A baseband state where a device listens for page trains.

1. The term ’connection’ used here is not identical to the definition below. It is used in the absence of a more concise term.

Page The transmission by a device of page trains containing the Device Access Code of the device to which the physical link is requested.

Page scan The listening by a device for page trains containing its own Device Access Code.

Channel A logical connection on L2CAP level between two devices serving a single application or higher layer protocol.

Connection A connection between two peer applications or higher layer protocols mapped onto a channel.

Connecting A phase in the communication between devices when a connection between them is being established. (Connecting phase follows after the link establishment phase is completed.)

Connect (to service) The establishment of a connection to a service. If not already done, this includes establishment of a physical link, link and channel as well.

8.3 DEVICE-RELATED DEFINITIONS

Discoverable device A Bluetooth device in range that will respond to an inquiry (normally in addition to responding to page).

Silent device A Bluetooth device appears as silent to a remote device if it does not respond to inquiries made by the remote device. A device may be silent due to being non-discoverable or due to baseband congestion while being discoverable.

Connectable device A Bluetooth device in range that will respond to a page.

Trusted device A paired device that is explicitly marked as trusted.

Paired device A Bluetooth device with which a link key has been exchanged (either before connection establishment was requested or during connecting phase).

Pre-paired device A Bluetooth device with which a link key was exchanged, and the link key is stored, before link establishment.

Un-paired device A Bluetooth device for which there was no exchanged link key available before connection establishment was request.

Known device A Bluetooth device for which at least the BD_ADDR is stored.

Un-known device A Bluetooth device for which no information (BD_ADDR, link key or other) is stored.

Authenticated device A Bluetooth device whose identity has been verified during the lifetime of the current link, based on the authentication procedure.

8.4 PROCEDURE-RELATED DEFINITIONS

Paging A procedure for establishing a physical link of ACL type on baseband level, consisting of a page action of the initiator and a page scan action of the responding device.

Link establishment A procedure for establishing a link on LMP level. A link is established when both devices have agreed that LMP setup is completed.

Channel establishment A procedure for establishing a channel on L2CAP level.

Connection establishment A procedure for creating a connection mapped onto a channel.

Creation of a trusted relationship A procedure where the remote device is marked as a trusted device. This includes storing a common link key for future authentication and pairing (if the link key is not available).

Creation of a secure connection. A procedure of establishing a connection, including authentication and encryption.

Device discovery A procedure for retrieving the Bluetooth device address, clock, class-of-device field and used page scan mode from discoverable devices.

Name discovery A procedure for retrieving the user-friendly name (the Bluetooth device name) of a connectable device.

Service discovery Procedures for querying and browsing for services offered by or through another Bluetooth device.

8.5 SECURITY-RELATED DEFINITIONS

Authentication A generic procedure based on LMP-authentication if a link key exists or on LMP-pairing if no link key exists.

LMP-authentication An LMP level procedure for verifying the identity of a remote device. The procedure is based on a challenge-response mechanism using a random number, a secret key and the BD_ADDR of the non-initiating device. The secret key used can be a previously exchanged link key or an initialization key created based on a PIN (as used when pairing).

Authorization A procedure where a user of a Bluetooth device grants a specific (remote) Bluetooth device access to a specific service. Authorization

implies that the identity of the remote device can be verified through authentication.

Authorize The act of granting a specific Bluetooth device access to a specific service. It may be based upon user confirmation, or given the existence of a trusted relationship.

LMP-pairing A procedure that authenticates two devices, based on a PIN, and subsequently creates a common link key that can be used as a basis for a trusted relationship or a (single) secure connection. The procedure consists of the steps: creation of an initialization key (based on a random number and a PIN), LMP-authentication based on the initialization key and creation of a common link key.

Bonding A dedicated procedure for performing the first authentication, where a common link key is created and stored for future use.

Trusting The marking of a paired device as trusted. Trust marking can be done by the user, or automatically by the device (e.g. when in pairable mode) after a successful pairing.

9 ANNEX A (NORMATIVE): TIMERS AND CONSTANTS

The following timers are required by this profile.

Timer name	Recommended value	Description	Comment
$T_{GAP}(100)$	10.24 s	Normal time span that a Bluetooth device performs inquiry.	Used during inquiry and device discovery.
$T_{GAP}(101)$	10.625 ms	Minimum time in INQUIRY_SCAN.	A discoverable Bluetooth device enters INQUIRY_SCAN for at least $T_{GAP}(101)$ every $T_{GAP}(102)$.
$T_{GAP}(102)$	2.56 s	Maximum time between repeated INQUIRY_SCAN enterings.	Maximum value of the inquiry scan interval, $T_{inquiry\ scan}$.
$T_{GAP}(103)$	30.72 s	A Bluetooth device shall not be in a discoverable mode less than $T_{GAP}(103)$.	Minimum time to be discoverable.
$T_{GAP}(104)$	1 min	A Bluetooth device should not be in limited discoverable mode more than $T_{GAP}(104)$.	Recommended upper limit.

Table 9.1: Defined GAP timers

10 ANNEX B (INFORMATIVE): INFORMATION FLOWS OF RELATED PROCEDURES

10.1 LMP-AUTHENTICATION

The specification of authentication on link level is found in [2].

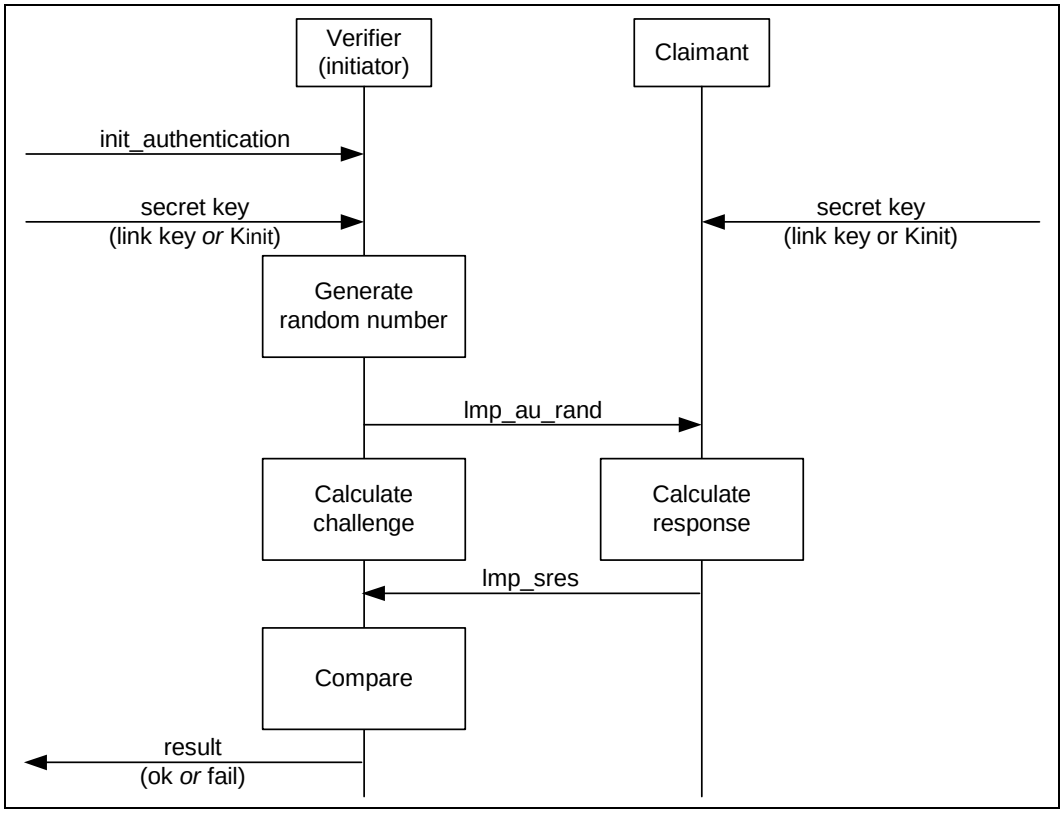


Figure 10.1: LMP-authentication as defined by [2].

The secret key used here may be either an already exchanged link key or an initialization key created in the LMP-pairing procedure.

10.2 LMP-PAIRING

The specification of pairing on link level is found in [2].

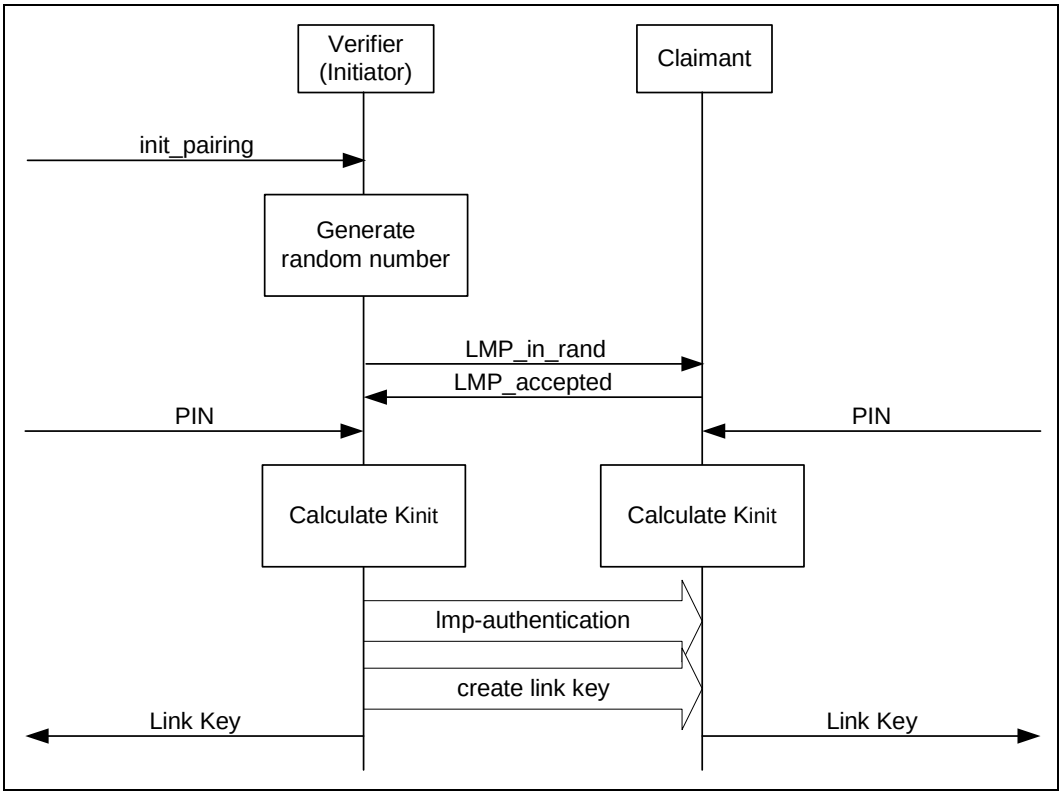


Figure 10.2: LMP-pairing as defined in [2].

The PIN used here is PN_{BB}.

The create link key procedure is described in section 3.3.4 of [2] and section 14.2.2 of [1]. In case the link key is based on a combination key, a mutual authentication takes place and shall be performed irrespective of current security mode.

10.3 SERVICE DISCOVERY

The Service Discovery Protocol [6] specifies what PDUs are used over-the-air to inquire about services and service attributes. The procedures for discovery of supported services and capabilities using the Service Discovery Protocol are described in the Service Discovery Application Profile. This is just an example.

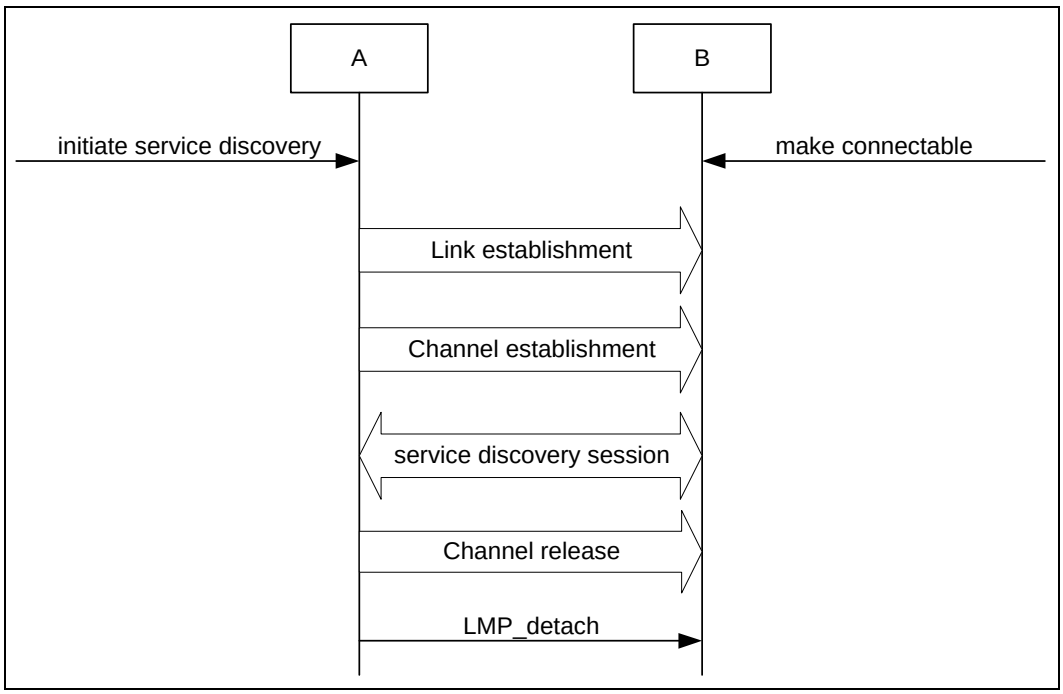


Figure 10.3: Service discovery procedure.

11 REFERENCES

[1] Bluetooth Baseband Specification

[2] Bluetooth Link Manager Protocol

[3] Bluetooth Logical Link Control and Adaptation Protocol

[4] Bluetooth RFCOMM

[5] Bluetooth Telephony Control Specification

[6] Bluetooth Service Discovery Protocol

[7] Bluetooth Service Discovery Application Profile

[8] Bluetooth Cordless Telephony Profile

[9] Bluetooth Serial Port Profile

[10] Bluetooth Security Architecture (white paper)

[11] Bluetooth Assigned Numbers

Part K:2

SERVICE DISCOVERY APPLICATION PROFILE

This document defines the features and procedures for an application in a Bluetooth device to discover services registered in other Bluetooth devices and retrieve any desired available information pertinent to these services.

CONTENTS

1 Introduction 66
1.1 Scope 66
1.2 Symbols and conventions 67
2 Profile overview 68
2.1 Profile stack 68
2.2 Configurations and roles 69
2.3 User requirements and scenarios 70
2.4 Profile fundamentals 71
2.5 Conformance 71
3 User interface aspects 72
3.1 Pairing 72
3.2 Mode selection 72
4 Application layer 73
4.1 The service discovery application 73
4.2 Service primitives abstractions 75
4.3 Message sequence charts (MSCs) 77
5 Service Discovery 79
5.1 An SDP PDU exchange example 80
6 L2CAP 82
6.1 Channel types 83
6.2 Signalling 83
6.3 Configuration options 83
6.3.1 Maximum Transmission Unit (MTU) 83
6.3.2 Flush Time-out 83
6.3.3 Quality of Service 84
6.4 SDP transactions and L2CAP connection lifetime 84
7 Link Manager 86
7.1 Capability overview 86
7.2 Error behavior 87
7.3 Link policy 87
8 Link control 88
8.1 Capability overview 88
8.2 Inquiry 89
8.3 Inquiry scan 90
8.4 Paging 90
8.5 Page scan 90
8.6 Error behavior 90

9 References 91

9.1 Normative references 91

10 Definitions 92

11 Appendix A (Informative): Service primitives and the Bluetooth PDUs 93

FOREWORD

Interoperability between devices from different manufacturers is provided for a specific service and use case, 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 use case(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

It is expected that the number of services that can be provided over Bluetooth links will increase in an undetermined (and possibly uncontrolled) manner. Therefore, procedures need to be established to aid a user of a Bluetooth-enabled device to sort the ever-increasing variety of services that will become available to him/her. While many of the Bluetooth-enabled services that may be encountered are currently unknown, a standardized procedure can still be put into place on how to locate and identify them.

The Bluetooth protocol stack contains a Service Discovery Protocol (SDP) BT_SDP_spec:[7] that is used to locate services that are available on or via devices in the vicinity of a Bluetooth enabled device. Having located what services are available in a device, a user may then select to use one or more of them. Selecting, accessing, and using a service is outside the scope of this document. Yet, even though SDP is not directly involved in accessing services, information retrieved via SDP facilitates service access by using it to properly condition the local Bluetooth stack to access the desired service.

The service discovery profile defines the protocols and procedures that shall be used by a service discovery application on a device to locate services in other Bluetooth-enabled devices using the Bluetooth Service Discovery Protocol (SDP). With regard to this profile, the service discovery application is a specific user-initiated application. In this aspect, this profile is in contrast to other profiles where service discovery interactions between two SDP entities in two Bluetooth-enabled devices result from the need to enable a particular transport service (e.g. RFCOMM, etc.), or a particular usage scenario (e.g. file transfer, cordless telephony, LAN AP, etc.) over these two devices. Service discovery interactions of the latter kind can be found within the appropriate Bluetooth usage scenario profile documents.

The service discovery in the other profile documents has a very narrow scope; e.g. learning about the protocols and related protocol parameters needed for accessing a particular service. Nevertheless, the fundamentals of the service discovery procedures covered in this profile document, and the use of the Bluetooth protocols in support of these procedures can be replicated in other profile documents as well. The only difference is that for the other profiles these procedures are initiated by application-level actions within the applications described by the corresponding profiles, as opposed to user-level actions for this profile.

SDP provides direct support for the following set of service inquiries:

- Search for services by service class;
- Search for services by service attributes; and
- Service browsing.

The generic service discovery application considered for this profile also covers the above service inquiry scenarios.

The former two cases represent searching for known and specific services. They provide answers to user questions like: “Is service A, or is service A with characteristics B and C, available?” The latter case represents a general service search and provides answers to questions like: “What services are available?” or “What services of type A are available?”

The above service inquiry scenarios can be realized two-fold:

- By performing the service searches on a particular device that a user ‘consciously’ has already connected to, and/or
- By performing the service searches by ‘unconsciously’ connecting to devices discovered in a device's vicinity.

Both of the above approaches require that devices need first to be discovered, then linked with, and then inquired about the services they support.

1.2 SYMBOLS AND CONVENTIONS

This profile uses the symbols and conventions specified in Section 1.2 of the Generic Access Profile [3].

2 PROFILE OVERVIEW

2.1 PROFILE STACK

Figure 2.1 shows the Bluetooth protocols and supporting entities involved in this profile.

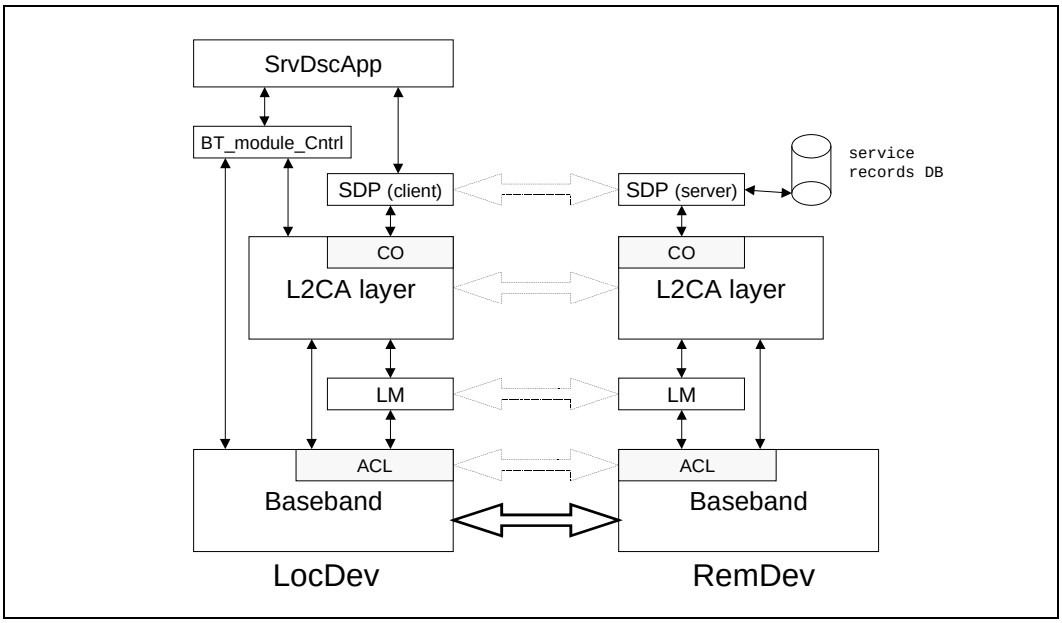


Figure 2.1: The Bluetooth protocol stack for the service discovery profile

The service discovery user application (SrvDscApp) in a local device (LocDev) interfaces with the Bluetooth SDP client to send service inquiries and receive service inquiry responses from the SDP servers of remote devices (RemDevs) BT_SDP_spec:[7]. SDP uses the connection-oriented (CO) transport service in L2CAP, which in turn uses the baseband asynchronous connectionless (ACL) links to ultimately carry the SDP PDUs over the air.

Service discovery is tightly related to discovering devices, and discovering devices is tightly related to performing inquiries and pages. Thus, the SrvDscApp interfaces with the baseband via the BT_module_Cntrl entity that instructs the Bluetooth module when to enter various search modes of operation.[1]

1. The BT_module_Cntrl may be part of a Bluetooth stack implementation (and thus be shared by many Bluetooth-aware applications) or a 'lower part' of the SrvDscApp. Since, no assumptions about any particular stack or SrvDscApp implementations are made, the BT_module_Cntrl entity represents a logical entity separate from the SrvDscApp, which may or may not be part of the SrvDscApp itself, a stack component, or any other appropriate piece of code.

The service records database (DB) shown in Figure 2.1 next to an SDP server is a logical entity that serves as a repository of service discovery-related information. The ‘physical form’ of this database is an implementation issue outside the scope of this profile.

2.2 CONFIGURATIONS AND ROLES

The following roles are defined in this profile:

- **Local device (LocDev):** A LocDev is the device that initiates the service discovery procedure. A LocDev must contain at least the *client* portion of the Bluetooth SDP architecture BT_SDP_spec:[7]. A LocDev contains the service discovery application (SrvDscApp) used by a user to initiate discoveries and display the results of these discoveries.
- **Remote Device(s) (RemDev(s)):** A RemDev is any device that participates in the service discovery process by responding to the service inquiries generated by a LocDev. A RemDev must contain at least the *server* portion of the Bluetooth SDP architecture BT_SDP_spec:[7]. A RemDev contains a service records database, which the server portion of SDP consults to create responses to service discovery requests.

The LocDev or RemDev role assigned to a device is neither permanent nor exclusive. A RemDev may also have a SrvDscApp installed into it as well as an SDP client, and a LocDev may also have an SDP server. In conjuction with which device has an SrvDscApp installed, an SDP-client installed, and an SDP-server installed, the assignment of devices to the above roles is relative to each individual SDP (and related) transaction and which device initiates the transaction. Thus, a device could be a LocDev for a particular SDP transaction, while at the very same time be a RemDev for another SDP transaction.

With respect to this profile, a device without a UI (directly or indirectly available) for entering user input and returning the results of service searches is not considered as a candidate for a LocDev. Nevertheless, even if such a device is not considered as a candidate for a LocDev, the procedures presented in the following sections can still apply if applications running in such a device need to execute a service discovery transaction.

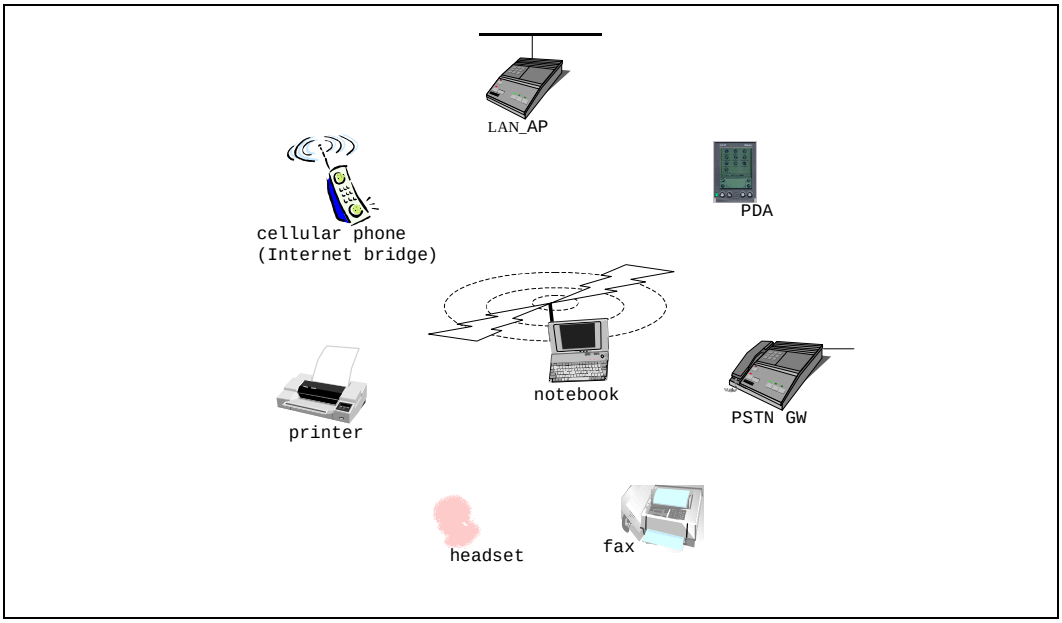


Figure 2.2: A typical service discovery scenario

The figure above shows a local device (the notebook) inquiring for services among a plethora of remote devices.

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are the following:

- Search for services by service class,
- Search for services by service attributes, and
- Service browsing.

The first two cases represent searching for known and specific services, as part of the user question “Is service A, or is service A with characteristics B and C, available?” The latter case represents a general service search that is a response to the user question “What services are available?”

This profile implies the presence of a Bluetooth-aware, user-level application, the SrvDscApp, in a LocDev that interfaces with the SDP protocol for locating services. In this aspect, this profile is unique as compared to other profiles. It is a profile that describes an application that interfaces to a specific Bluetooth protocol to take full advantage of it for the direct benefit of an end-user.

2.4 PROFILE FUNDAMENTALS

Before any two Bluetooth-equipped devices can communicate with each other the following may be needed:

- The devices need to be powered-on and initialized. Initialization may require providing a PIN for the creation of a link key, for device authorization and data encryption.
- A Bluetooth link has to be created, which may require the discovery of the other device's BD_ADDR via an inquiry process, and the paging of the other device.

While it may be seem natural to consider a LocDev serving as a Bluetooth master and the RemDev(s) serving as Bluetooth slave(s), there is no such requirement imposed on the devices participating in this profile. Service discovery as presented in this document can be initiated by either a master or a slave device at any point for which these devices are members of the same piconet. Also, a slave in a piconet may possibly initiate service discovery in a new piconet, provided that it notifies the master of the original piconet that it will be unavailable (possibly entering the hold operational mode) for a given amount of time.[2]

The profile does not require the use of authentication and/or encryption. If any of these procedures are used by any of the devices involved, service discovery will be performed only on the subset of devices that pass the authentication and encryption security ‘roadblocks’ that may impose to each other. In other words, any security restrictions for SDP transactions are dictated by the security restrictions already in place (if any) on the Bluetooth link.

2.5 CONFORMANCE

If conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies to 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 certification program.

2. Recall that a master of a piconet cannot initiate a new piconet. Since a piconet is ultimately identified by the BD_ADDR and the Bluetooth clock of its master, the latter piconet will be identical to and indistinguishable from the former.

3 USER INTERFACE ASPECTS

3.1 PAIRING

No particular requirements regarding pairing are imposed by this profile. Pairing may or may not be performed. Whenever a LocDev performs service discovery against as yet ‘unconnected’ RemDev(s), it shall be the responsibility of the SrvDscApp to allow pairing prior to connection, or to by-pass any devices that may require pairing first. This profile is focused on only performing service discovery whenever the LocDev can establish a legitimate and useful baseband link[3] with RemDev(s).

3.2 MODE SELECTION

This profile assumes that, under the guidance of the SrvDscApp, the LocDev shall be able to enter the inquiry and/or page states. It is also assumed that a RemDev with services that it wants to make available to other devices (e.g. printer, a LAN DAP, a PSTN gateway, etc.) shall be able to enter the inquiry scan and/or page scan states. For more information about the inquiry and page related states see Section 8.

Since the SrvDscApp may also perform service inquiries against already connected RemDevs, it is not mandatory according to the profile that a LocDev always be the master of a connection with a RemDev. Similarly, a RemDev may not always be the slave of a connection with a LocDev.

3. A legitimate and useful baseband link is a Bluetooth baseband link that is properly authenticated and encrypted (if so desired), whenever any of these options are activated by any of the devices participating in this profile.

4 APPLICATION LAYER

4.1 THE SERVICE DISCOVERY APPLICATION

In this subsection, the operational framework of the SrvDscApp is presented.[4] Figure 4.1 shows alternative possibilities for a SrvDscApp.

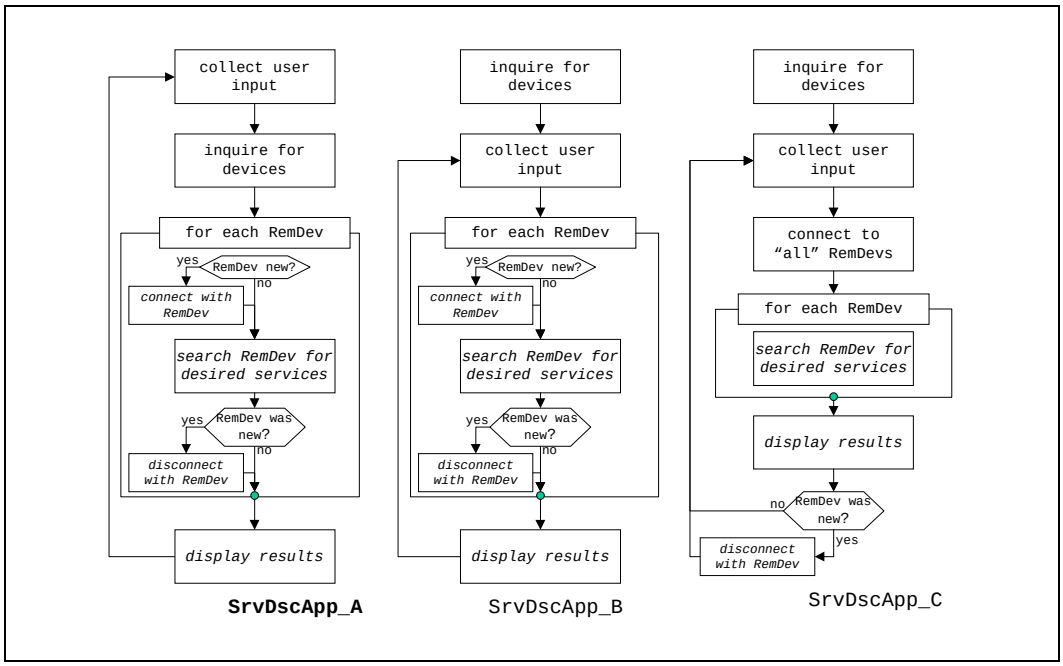


Figure 4.1: Three possible SrvDscApps

The SrvDscApp alternatives shown in Figure 4.1, which are not exhaustive by any means, achieve the same objectives but they follow different paths for achieving them. In the first alternative (SrvDscApp_A), the SrvDscApp on a LocDev inquires its user to provide information for the desired service search. Following this, the SrvDscApp searches for devices, via the Bluetooth inquiry procedure. For each device found, the LocDev will connect to it, perform any necessary link set-up, see related procedures in Generic Access Profile [3], and then inquire it for the desired services. In the second alternative (SrvDscApp_ B), the inquiry of devices is done prior to collecting user input for the service search.[5]

4. This profile does not dictate any particular implementation for a SevDisApp. It only presents the procedures needed to achieve its objectives.
5. Device inquiries may even occur by means outside the scope of a particular SrvDscApp implementation. But, since such other means are not guaranteed to exist, it is recommended that the SrvDscApp activates device inquiries too.

In the first two alternatives, page, link creation, and service discovery are done sequentially on a per RemDev basis; i.e., the LocDev does not page any new RemDev prior to completing the service search with a previous RemDev and (if necessary) disconnecting from it. In the last alternative (SrvDscApp_C), the LocDev, under the control of the SrvDscApp, will first page all RemDevs, then will create links with all of these devices (up to a maximum of 7 at a time), and then inquire all the connected devices for the desired services.

Just as an example, we focus on a SrvDscApp similar to the one represented by the SrvDscApp_A in Figure 4.1. In summary, SrvDscApp (for ease of notation, the suffix '_A' has been dropped) has the following features:

- The SrvDscApp activates Bluetooth inquiries following a user request for a service search,
- For any new RemDev found following an inquiry, the SrvDscApp will finish service discovery and terminate its link against this device prior to attempting to connect to the next RemDev,
- For any RemDev already connected, the LocDev does not disconnect following service discovery, and
- The user of the SrvDscApp has the option of a trusted and untrusted mode of operation, whereby the SrvDscApp permits connections –

  a) only with trusted RemDev, or

  b) with any of the devices above plus any newly discovered RemDevs that require nothing more beyond possibly pairing with the default all-zero PIN, or

  c) with any of the devices above, plus any additional RemDev for which the user explicitly enters a non-zero PIN.

The above options have to do with the degree of user involvement in configuring and interacting with the SrvDscApp and setting the security levels that the user is willing to accept for the service searches. When selecting options (a) or (b), then for the devices with which no legitimate connections can be established, it is assumed that the SrvDscApp ignores them without any cue to its user (however, this too is an implementation issue).

When a LocDev performs a service discovery search, it does so against three different types of RemDevs:

1. *trusted devices:* These are devices that are currently not connected with the LocDev but the LocDev device has already an established trusted relation with.
2. *unknown (new) devices:* These are untrusted devices that are currently not connected with the LocDev.
3. *connected devices:* These are devices that are already connected to the LocDev.

To discover type 1 or 2 RemDevs, the SrvDscApp needs to activate the Bluetooth inquiry and/or page processes. For type 3 RemDevs, the latter processes are needed. To perform its task, SrvDscApp needs to have access to the BD_ADDR of the devices in the vicinity of a LocDev, no matter whether these devices have been located via a Bluetooth inquiry process or are already connected to the LocDev. Thus, BT_module_Cntr in a LocDev shall maintain the list of devices in the vicinity of the LocDev and shall avail this list to the SrvDscApp.

4.2 SERVICE PRIMITIVES ABSTRACTIONS

This section briefly describes the functionality of a SrvDscApp. This functionality is presented in the form of service primitive abstractions that provide a formal framework for describing the user expectations from a SrvDscApp. It is assumed that the underlying Bluetooth stack can meet the objectives of these service primitive abstractions directly or indirectly.[6] The exact syntax and semantics of the service primitive abstractions (or simply “service primitives”) may be platform-dependent (e.g. an operating system, a hardware platform, like a PDA, a notebook computer, a cellular phone, etc.) and are beyond the scope of this profile. However, the functionality of these primitives is expected to be available to the SrvDscApp to accomplish its task.

Table 4.1 contains a minimum set of enabling service primitives to support a SrvDscApp. Low-level primitives like **openSearch**(.) or **closeSearch**(.) are not shown and are assumed to be part of the implementation of the primitives shown whenever necessary. Different implementations of the Bluetooth stack shall (at a minimum) enable the functions that these service primitives provide. For example, the **serviceSearch**(.) service primitive permits multiple identical operations to be handled at once. A stack implementation that requires an application to accomplish this function by iterating through the multiple identical operations one-at-a-time will be considered as enabling the function of this service primitive.[7] The service primitives shown next relate only to service primitives whose invocation result or relate to an over-the-air data exchange using the Bluetooth protocols. Additional service primitives can be envisioned relating to purely local operations like *service registration*, but these primitives are outside the scope of this profile.

6. These service primitive abstractions do *not* represent programming interfaces, even though they may be related to them. The word ‘directly’ is used to describe the possibility that the described function is the result of a single appropriate call of the underlying Bluetooth stack implementation. The word ‘indirectly’ is used to describe the possibility that the described function can be achieved by combining the results from multiple appropriate calls of the underlying Bluetooth stack implementation.
7. Even though the service primitives presented in this profile are assumed to act upon a local device for accessing *physically* remote devices, they are general enough to apply in cases where the ‘remote device’ characterization is only a logical concept; i.e. inquired service records and service providers are located within the same device that invokes these primitives. This general situation is outside the scope of this profile.

service primitive abstraction	resulted action
serviceBrowse (LIST(*RemDev)* LIST(*RemDevRelation*) LIST(*browseGroup*) *getRemDevName* *stopRule*)	a search for services (service browsing) that belong to the list of *browseGroup* services in the devices in the list of RemDevs; the search may be further qualified with a list of *RemDevRelation* parameters, whereby a user specifies the trust and connection relation of the devices to be searched; e.g. search only the devices that are in the *RemDev* list for which there is a trust relation already established; when the *getRemDevName* parameter is set to "yes," the names of the devices supporting the requested services are also returned; the search continues until the stopping rule *stopRule* is satisfied
serviceSearch (LIST(*RemDev)* LIST(*RemDevRelation*) LIST(*searchPattern*, *attributeList*) *getRemDevName* *stopRule*)	a search whether the devices listed in the list of RemDevs support services in the requested list of services; each service in the list must have a service search pattern that is a superset of the *searchPattern*; for each such service the values of the attributes contained in the corresponding *attributeList* are also retrieved; the search may be further qualified with a list of *RemDevRelation* parameters, whereby a user specifies the trust and connection relation of the devices to be searched (e.g. search only the devices that are in the *RemDev* list for which there is a trust relation already established); when the *getRemDevName* parameter is set to "yes," the names of the devices supporting the requested services are also returned; the search continues until the stopping rule *stopRule* is satisfied
enumerateRemDev (LIST(*classOfDevice*) *stopRule*)	a search for RemDev in the vicinity of a LocDev; RemDev searches may optionally be filtered using the list of *classOfDevice (*e.g. LAN APs); the search continues until the stopping rule *stopRule* is satisfied
terminatePrimitive (*primitiveHandle* *returnResults*)	a termination the actions executed as a result of invoking the services primitive identified by the *primitiveHandle*;* optionally, this service primitive may return any partially accumulated results related to the terminated service primitive

Table 4.1: Service primitives in support of SrvDscApp

*. It is assumed that each invocation of a service primitive can be identified by a *primitiveHandle,* the realization of which is implementation-dependent.

The *stopRule* parameter is used to guarantee a graceful termination of a service search. It could represent the number of search items found, or the duration of search, or both. A Bluetooth stack implementation may not expose this parameter, in which case it should provide guarantees that all searches terminate within a reasonable amount of time, for example, say, 120sec.

The **enumerateRemDev**(.) service primitive is directly related to the inquiry mode of operation for the baseband. It also relates to the collection of RemDev that a LocDev is currently connected with. This service is exported to the SrvDscApp via the BT_module_Cntr, see Figure 2.1. The interface between BT_module_Cntr and baseband is for activating Bluetooth inquiries and collecting the results of these inquiries. The interface between the BT_module_Cntrl and (an) L2CAP (implementation) is for keeping track of the RemDev that currently are connected to the LocDev.

The result of the **enumerateRemDev**(.) service primitive can be used with the **serviceSearch**(.) to search for desired services in the devices found. Once again, based on the implementation of the Bluetooth stack, this service primitive may not be provided explicitly, but its service may be provided within other service primitives; e.g. the **serviceSearch**(.).

Missing primitive parameters shall be interpreted (whenever appropriate) as a general service search on the remaining parameters. For example, if the LIST(*RemDev*) parameter is missing from the **serviceSearch**(.), it means that the search shall be performed against any device found in the vicinity of a LocDev. In this case, the first two service primitives may be combined to a single one.

The above service primitives return the requested information, whenever found. Based on the way that these service primitives are supported by a Bluetooth stack implementation, the results of a search may directly return by the corresponding calling function, or a pointer to a data structure may be returned that contains all the relevant information. Alternatively, a Bluetooth stack implementation may have altogether different means for providing the results of a search.

4.3 MESSAGE SEQUENCE CHARTS (MSCS)

This profile is concerned with three distinct Bluetooth procedures. Device discovery, device name discovery, service discovery. Note that each one of these procedures does not preclude any other; e.g. to connect to a RemDev, a LocDev may have to first discover it, and it may also ask for its name. The MSCs relating to the first two procedures (i.e., device and name discovery) are provided in section 2 of LM/HCI_MSCs:[6]. Sections 3, 4.1 and 4.2 of LM/HCI_MSCs:[6] provide the MSCs relating to the third procedure (i.e., service discovery). See also section 4 of BT_LM_spec:[4]. The first two procedures do not require host intervention, while the third does.

Figure 4.2 summarizes the key message exchange ‘phases’ encountered during the execution of this profile. Not all procedures are present at all times, and not all devices need to go through these procedures all the time. For example, if authentication is not required, the authentication phase in the figure will not be executed. If the SrvDsvApp needs to inquire for services on a specific RemDev with which the LocDev is currently connected, inquiries and pages

may not be executed. In the figure, the conditions under which particular phases are executed or not are also provided.

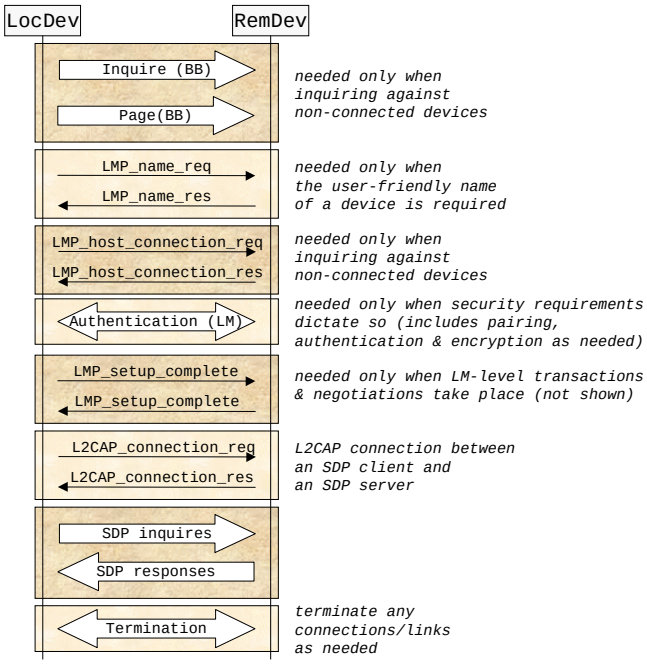


Figure 4.2: Bluetooth processes in support of this profile

In addition to the MSC in Figure 4.2, Annex A shows what Bluetooth procedures and PDUs are needed to support the service primitives presented in Section 4.2.

5 SERVICE DISCOVERY

The service discovery application does not make use of SDP as a means of accessing a service, but rather as a means of informing the user of a LocDev about the services that are available to his/her device by (and possibly via) RemDev(s). BT-aware applications running in a local device can also use the procedures described in this and the following sections to retrieve any pertinent information that will facilitate the application in accessing a desired service in a remote device.

Table 5.1 shows the SDP feature requirements in a LocDev and in a RemDev.

	SDP feature	Support in LocDev	Support in RemDev
1.	SDP client	M	O
2.	SDP server	O	M

Table 5.1: SDP feature requirements

Table 5.2 shows the SDP PDUs can be exchanged between devices following this profile.

SDP PDUs	Ability to Send		Ability to Receive	
	LocDev	RemDev	LocDev	RemDev
SDP_ErrorResponce	C1	M	M	C1
SDP_ServiceSearchRequest	M	C1	C1	M
SDP_ServiceSearchResponse	C1	M	M	C1
SDP_ServiceAttributeRequest	M	C1	C1	M
SDP_ServiceAttributeResponse	C1	M	M	C1
SDP_ServiceSearchAttributeRequest	M	C1	C1	M
SDP_ServiceSearchAttributeResponse	C1	M	M	C1
Comments:				
[C1]: With regard to this current profile, these PDU transmissions will not occur. Nevertheless, since a device could act as a LocDev on some occasions and as a RemDev on others, these PDU transmission may still take place between these devices.				

Table 5.2: Allowed SDP PDUs

5.1 AN SDP PDU EXCHANGE EXAMPLE

Figure 5.1 shows two examples of SDP PDU exchanges. In particular, it shows PDU exchange sequences for the inquiry and retrieval of any information pertinent to a particular Bluetooth profile.

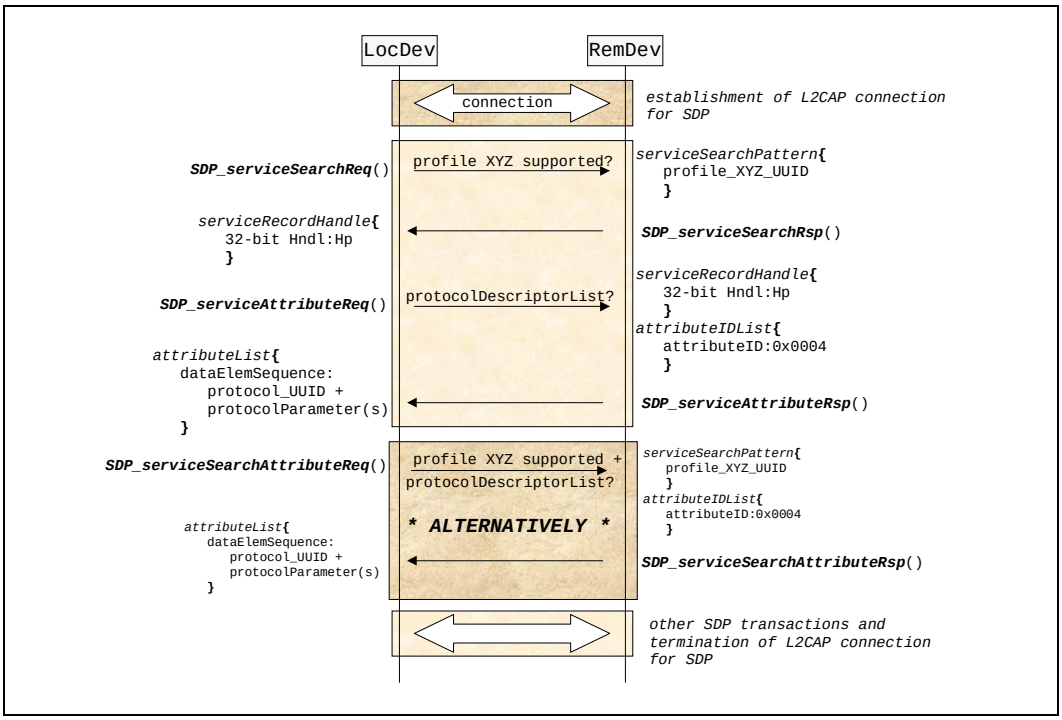


Figure 5.1: SDP PDU exchange examples for retrieving protocolDescriptorLists

For each PDU sent, the figure shows which device sends it (shown on the starting side of an arrow) and any relative information that this PDU carries (shown on the ending side of an arrow). Note that the LocDev sends request PDUs, while the RemDev sends back response PDUs.

Two alternatives are shown utilizing different SDP PDUs to ultimately retrieve the same information – the *protocolDescriptorList* attribute from devices that support a specific Bluetooth profile. With the first alternative, the desired information is derived in two steps.

- The LocDev sends an *SDP_serviceSearchReq* PDU which contains a service search pattern composed of the UUID associated with the desired profile; see section 4.3 of BT_ASN:[2]. The desired profile (profile ‘XYZ’) is identified by its UUID, denoted in the figure as ‘profile_XYZ_UUID.’ In its response PDU, the SDP server returns one or more 32-bit service record handles whose corresponding service records contain the ‘profile_XYZ_UUID’ UUID. In the figure, only one such handle is shown, denoted as ‘prHndl’.
- The LocDev then enters prHndl in an *SDP*_serviceAttribute PDU together with one or more attribute IDs. In this example, the attribute of interest is the

protocolDescriptorList, whose attribute ID is 0x0004. The SDP server then, in its response, returns the requested protocol list.

In the event that no service record containing the desired service search pattern is found in the SDP server, the *SDP_serviceSearchResp* PDU will contain an empty *serviceRecordHandleList* and a *totalServiceRecordCount* parameter set to its minimum value; see section 4.5.2 of BT_SDP_spec:[7].

If the desired attributes do not exist in the SDP server, the *SDP_serviceAttributeResp* PDU will contain an empty *attributeList* and an *attributeListByteCount* parameter set to its minimum value, see section 4.6.2 of BT_SDP_spec:[7].

With the second alternative, the desired attributes are retrieved in one step:

- The LocDev sends an *SDP_serviceSearchAttributeReq* PDU where both the desired profile is included (service search pattern: profile_XYZ_UUID) and the desired attribute(s) is provided (attribute ID: 0x0004). In its response the SDP server will provide the requested attribute(s) from the service record(s) that matches the service search pattern.

In case no service record containing the desired service search pattern and/or the desired attribute(s) is found in the SDP server, the *SDP_serviceSearchAttributeResp* PDU will contain an empty *attributeLists* and an *attributeListsByteCount* parameter set to its minimum value, see section 4.7.2 of BT_SDP_spec:[7].

While, in the example in Figure 5.1, only very few service attributes are shown retrieved by the SDP client, additional information could and should be requested. Particularly in cases where service information is to be cached for future use, an SDP client should also request any pertinent information that can aid in assessing whether cached information has become stale.
The service attributes *serviceDatabaseState*, *serviceRecordState*, and *serviceInfoTimeToLive* have been defined for this purpose in BT_SDP_spec:[7]; see sections 5.2.4, 5.1.3 and 5.1.8 respectively.

6 L2CAP

The following text, together with the associated subclauses, defines the mandatory requirements with regard to this profile.

	L2CAP procedure		Support in LocDev	Support in RemDev
1.	Channel types			
		Connection-oriented channel	M	M
		Connectionless channel	X1	X1
2.	Signalling			
		Connection Establishment	M	C1
		Configuration	M	M
		Connection Termination	M	C2
		Echo	M	M
		Command Rejection	M	M
3.	Configuration Parameter Options			
		Maximum Transmission Unit	M	M
		Flush Time-out	M	M
		Quality of Service	O	O

Comments:

[X1]: This feature is not used in this profile, but its use by other applications running simultaneously with this profile is not excluded.

[C1]: An SDP server shall not (and cannot) initiate an L2CAP connection for SDP transactions. Nevertheless, the device that the SDP server resides in may also have an SDP client that may initiate an L2CAP connection for SDP transactions. Such action does not contradict the execution of this profile. In any case, a RemDev shall be able to process incoming requests for connection establishment.

[C2] Under normal operation, an SDP server shall not initiate the process of terminating an L2CAP connection for SDP. However, exceptional cases, such as when a RemDev shuts down during the execution on an SDP transaction, cannot be excluded. In such a case, prior to the final power-off, the RemDev may gracefully (or not!) terminate all its active L2CAP connections by sending connection termination PDUs. In any case, a RemDev shall always be able to process incoming requests for connection termination.

Table 6.1: L2CAP procedures

6.1 CHANNEL TYPES

In this profile, only connection-oriented channels shall be used. In particular, no L2CAP broadcasts are to be used for this profile.

6.2 SIGNALLING

For the purpose of retrieving SDP-related information, only a LocDev can initiate an L2CAP connection request and issue an L2CAP connection request PDU; for exceptions, see comments C1 and C2 on Table 6.1. Likewise with the corresponding L2CAP connection terminations, and the same exceptional comments C1 and C2 on Table 6.1 apply. Other than that, SDAP does not impose any additional restrictions or requirements on L2CAP signalling.

In the PSM field of the Connection Request packet, the value 0x0001 (see section 5.2 of BT_L2CAP_spec:[5]) shall be used to indicate the request for creation of an L2CAP connection for accessing the SDP layer.

6.3 CONFIGURATION OPTIONS

This section describes the usage of configuration options in the service discovery profile.

6.3.1 Maximum Transmission Unit (MTU)

This profile does not impose any additional restrictions to MTU beyond the ones stated in section 6.1 of BT_L2CAP_spec:[5]. If no MTU negotiation takes place, the default MTU value in section 6.1 of BT_L2CAP_spec:[5] shall be used.

For efficient use of the communication resources, the MTU shall be selected as large as possible, while respecting any physical constraints imposed by the devices involved, and the need that these devices continue honoring any already agreed upon QoS contracts with other devices and/or applications. It is expected that during the lifetime of an L2CAP connection for SDP transactions (also referred to as the ‘SDP session’, see Section 6.4) between two devices, any one of these devices may become engaged in an L2CAP connection with another device and/or application. If this new connection has ‘non-default’ QoS requirements, the MTU for the aforementioned SDP session is allowed to be re-negotiated during the lifetime of this SDP session, to accommodate the QoS constraints of the new L2CAP connection.

6.3.2 Flush Time-out

The SDP transactions are carried over an L2CAP reliable channel. The flush time-out value (see section 6.2 of BT_L2CAP_spec:[5]) shall be set to its default value 0xFFFF.

6.3.3 Quality of Service

The use of Quality of Service (QoS) and QoS negotiation is optional. If QoS is to be negotiated, the default settings in section 6.4 of BT_L2CAP_spec:[5] shall be used. In particular, SDP traffic shall be treated as a best-effort service type traffic.

6.4 SDP TRANSACTIONS AND L2CAP CONNECTION LIFETIME

While, in general, SDP transactions comprise a sequence of service request-and-response PDU exchanges, SDP itself constitutes a connectionless datagram service in that no SDP-level connections are formed prior to any SDP PDU exchange. SDP delegates the creation of connections on its behalf to the L2CAP layer. It is thus the responsibility of SDP – or, more correctly, of the SDP layer – to request the L2CAP layer to ‘tear down’ these connections on its behalf as well.

Since SDP servers are considered stateless, ‘tearing down’ an L2CAP connection after a service request PDU is sent (as a true connectionless service may imply) will be detrimental to the SDP transaction. Moreover, significant performance penalty will have to be paid if, for each SDP PDU transmission, a new L2CAP connection is to be created. Thus, L2CAP connections for SDP transactions shall last more than the transmission of a single SDP PDU.

An SDP *session* between an SDP client and an SDP server represents the time interval that the client and the server have the same L2CAP connection continuously present. A *minimal* SDP transaction will represent a single exchange of an SDP request PDU transmission from an SDP client to an SDP server, and the transmission of a corresponding SDP response PDU from the SDP server back to the SDP client. With respect to this profile, under normal operational conditions, the minimum duration of an SDP session shall be the duration of a minimal SDP transaction.

An SDP session may last less than the minimum required in the event of unrecoverable (processing or link) errors in layers below SDP in the LocDev and RemDev, or in the SDP layer and the service records database in the RemDev. An SDP session may also be interrupted by user intervention that may terminate the SDP session prior to the completion of an SDP transaction.

The above minimum duration of an SDP session guarantees smooth execution of the SDP transactions. For improved performance, implementers may allow SDP sessions to last longer than the minimum duration of an SDP session. As a general implementation guideline, an SDP session shall be maintained for as long as there is a need to interact with a specific device. Since the latter time is in general unpredictable, SDP implementations may maintain timers used to time periods of SDP transaction inactivity over a specific SDP session.

SDP implementations may also rely on explicit input received from a higher layer (probably initiated from the SrvDscApp itself) to open and close an SDP session with a particular device using low level primitives; e.g. **openSearch**(.) and **closeSearch**(.). Finally, an implementation may permit users to interrupt an SDP session at any time, see the **terminatePrimitive**(.) service primitive in Section 4.2.

Normally, an SDP session shall not terminate by a RemDev. Yet, such an event can indeed occur, either having the RemDev gracefully terminating the SDP session, using the L2CAP connection termination PDU, or abnormally terminating the SDP by stopping responding to SDP requests or L2CAP signalling commands. Such an event may be an indication of an exceptional condition that SDP client/server implementers should consider addressing for the smooth execution of this profile. If a termination event initiates from a RemDev, an SDP client may want to consider clearing any information obtained by this RemDev. Such an exceptional event may imply that the SDP server has (or is about to) shut-down, in which case any service information retrieved from this server should automatically become stale.

7 LINK MANAGER

7.1 CAPABILITY OVERVIEW

In this section, the LMP layer is discussed. In the table below, all LMP features are listed. The table shows which LMP features are mandatory to support with respect to this service discovery profile, which are optional and which are excluded. The reason for excluding features is that they may degrade operation of devices in this use case. Therefore, these features shall never be activated by a unit active in this use case.

If any of the rules stated below are violated, the units shall behave as defined in Section 7.2.

Traffic generated during service discovery interactions has no particular QoS requirements. As such, no particular provision of the Bluetooth link is required to support this profile.

	LM Procedure	Support in LMP	Support in LocDev	Support in RemDev
1.	Authentication	M	C1	C1
2.	Pairing	M		
3.	Change link key	M		
4.	Change the current link key	M		
4.	Encryption	O	C1	C1
5.	Clock offset request	M		
6.	Timing accuracy information request	O		
7.	LMP version	M		
8.	Supported features	M		
9.	Switch of master slave role	O		
10.	Name request	M		
11.	Detach	M		
12.	Hold mode	M		
13.	Sniff mode	O		
14.	Park mode	O		
15.	Power control	O		

Table 7.1: LMP procedures

	LM Procedure	Support in LMP	Support in LocDev	Support in RemDev
16.	Channel quality driven DM/DH	O		
17.	Quality of service	M		
18.	SCO links	O	X1	X1
19.	Control of multi-slot packets	M		
20.	Concluding parameter negotiation	M		
21.	Host connection	M		
Comments:				
[C1] No authentication or encryption is required specifically by this profile. This profile will, however, not attempt to change the existing operational settings for these procedures. Nevertheless, when this profile is executed all by itself, the default operational settings are: - authentication: no active - encryption: no active In the latter case, a LocDev will always comply with the security requirements imposed by a RemDev. If it cannot comply, it will bypass the RemDev. [X1]: This feature is not used in this profile, but its use by other applications running simultaneously with this profile is not excluded.				

Table 7.1: LMP procedures

7.2 ERROR BEHAVIOR

If a unit tries to use a mandatory feature, and the other unit replies that it is not supported, the initiating unit shall send an LMP_detach PDU with detach reason "unsupported LMP feature."

A unit shall always be able to handle the rejection of the request for an optional feature.

7.3 LINK POLICY

There are no fixed master-slave roles for the execution of this profile.

This profile does not state any requirements on which low-power modes to use, or when to use them. It is up to the Link Manager of each device to decide and request special link features as seen appropriate.

8 LINK CONTROL

8.1 CAPABILITY OVERVIEW

The following table lists all features on the LC level

	Procedure	Support in baseband	Support in LocDev	Support in RemDev
1.	Inquiry	M	C1	
2.	Inquiry scan	M		C2
3.	Paging	M	C1	
4.	Page scan			
A	Type R0	M		C3
B	Type R1	M		C3
C	Type R2	M		C3
5.	Packet types			
A	ID packet	M		
B	NULL packet	M		
C	POLL packet	M		
D	FHS packet	M		
E	DM1 packet	M		
F	DH1 packet	M		
G	DM3 packet	O		
H	DH3 packet	O		
I	DM5 packet	O		
J	DH5 packet	O		
K	AUX packet	M	X1	X1
L	HV1 packet	M	X1	X1
M	HV2 packet	O	X1	X1
N	HV3 packet	O	X1	X1
O	DV packet	M	X1	X1
6.	Inter-piconet capabilities	O		
7.	Voice codec			

Table 8.1: LC features

	Procedure	Support in baseband	Support in LocDev	Support in RemDev
A	A-law	O	X1	X1
B	µ-law	O	X1	X1
C	CVSD	O	X1	X1
Comments:				
[C1]: This mandatory LC feature will be activated under the control of the SrvDscApp.				
[C2]: This mandatory LC feature is a settable device policy (outside the scope of this profile) that is activated whenever a device is to operate in a discoverable (public) mode.				
[C3] This mandatory LC feature is a settable device policy (outside the scope of this profile) that is activated whenever a device is to operate in a discoverable or connectable (private) mode.				
[X1]: These features are not used in this profile, but their use by other applications running simultaneously with this profile is not excluded.				

Table 8.1: LC features

For the next four subsections, it is assumed that a LocDev is to perform service searches with originally unconnected RemDevs. It thus needs to inquire for and page (or only page) these RemDevs. None of the following four subsections apply whenever a LocDev performs service searches with RemDevs to which it is already connected.

8.2 INQUIRY

Whenever instructed by the SrvDscApp, the LocDev shall advise its baseband to enter the inquiry state. Entry into this state may or may not be immediate, however, depending on QoS requirements of any already existing and ongoing connections.

The user of the SrvDscApp shall be able to set the criteria for the duration of an inquiry, see *stopRule* service primitive parameter in Section 4.2. Nevertheless, the actual residence time in the inquiry state must comply with the recommendation given in section 10.7.3 of Bluetooth Baseband Specification [1].

When inquiry is invoked in a LocDev, the general inquiry procedure shall be used using a GIAC as described in Section 6.1 of Bluetooth GAP_profile:[3].

Instead of a GIAC, an appropriate DIAC can be used to narrow down the scope of the inquiry. Since the only defined DIAC (referred to as the LIAC) does not reflect any specific device or service categories, the use of DIACs is of limited (but non-zero) benefit in this profile. In particular, the profile does not exclude (but neither does it encourage) performing inquiries according to the limited inquiry procedure described in Section 6.2 of GAP_profile:[3].The information contained in the Class of Device field in the FHS packet returned by the ‘inquired devices’ can be used as a filter to limit the number of devices to page and connect to for subsequent SDP transactions.

8.3 INQUIRY SCAN

Inquiry scans are device-dependent policies outside the scope of this profile. Devices that operate in a discoverable mode of operation, see Section 4.1 of GAP_profile:[3], could be discovered by inquiries sent by other devices.

To be discovered by an inquiry resulting from a SrvDscApp action, a RemDev must enter inquiry scans using the GIAC; see general discoverable mode in Section 4.1.3 of GAP_profile:[3]. A DIAC can be used instead of a GIAC. As previously mentioned, the use of DIACs are of limited (but non-zero) benefit in this profile. In particular, performing inquiry scans according to the limited discoverable procedure described in Section 6.2 of GAP_profile:[3] is not excluded, but is not encouraged either.

8.4 PAGING

Whenever the SrvDscApp needs to connect to a specific RemDev for inquiring about its service records, the LocDev will advise its baseband to enter the page state. Entry into this state may or may not be immediate, however, depending on QoS requirements of any already existing and ongoing connections.

Depending on the paging class (R0, R1, or R2) indicated by a RemDev device, the LocDev shall page accordingly. The total residence time in the page state must comply with the recommendation given in section 10.6.3 of BT_BB_spec:[1]. For the pages, the 48-bit BD_ADDR of the RemDev must be used.

8.5 PAGE SCAN

Just like inquiry scans, page scans are device-dependent policies outside the scope of this profile. Devices that operate in a connectable mode of operation, see Section 4.2.2 of GAP_profile:[3], could establish Bluetooth links with other devices from pages sent by these other devices. To establish a link with a RemDev, a LocDev must send a page that results from a SrvDscApp action using the RemDev's 48-bit BD_ADDR.

8.6 ERROR BEHAVIOR

Since most features on the LC level have to be activated by LMP procedures, errors will usually be caught at that layer. However, there are some LC procedures that are independent of the LMP layer, such as inquiry or paging. Misuse of such features is difficult or sometimes impossible to detect. There is no mechanism defined to detect or prevent such improper use.

9 REFERENCES

9.1 NORMATIVE REFERENCES

[1] Baseband specification (see Volume 1, Part B)

[2] Bluetooth Assigned Numbers (see Volume 1, Appendix VIII)

[3] Generic Access Profile (see Volume 2, Part K1)

[4] Link Manager Protocol (see Volume 1, Part C)

[5] Logical Link Control and Adaptation Protocol Specification (see Volume 1, Part D)

[6] Message Sequence Charts between Host–Host Controller/Link Manager (see Volume 1, Appendix IX)

[7] Service Discovery Protocol (see Volume 1, Part E)

10 DEFINITIONS

Term	Definition
conscious	(usually referred to) a process that requires the explicit intervention of a user to be accomplished
known	(with respect to a specific device) opposite to *unknown*; a known devices is not necessarily a *paired* device
new (RemDev)	(with regard to this profile) an additional remote device (RemDev) that is discovered during a Bluetooth inquiry, and that is not already connected to local device (LocDev)
private	a mode of operation whereby a device can only be found via Bluetooth baseband pages; i.e. it only enters page scans
public	a mode of operation whereby a device can be found via Bluetooth baseband inquiries; i.e. it enters into inquiry scans. A public device also enters into page scans (contrast this with *private*)
unconscious	opposite to *conscious*
unknown	(with respect to a specific device) any other device that a specific device has no record of

11 APPENDIX A (INFORMATIVE): SERVICE PRIMITIVES AND THE BLUETOOTH PDUs

In this Annex, we relate the service primitives shown in section 4.2 with the various Bluetooth PDUs which support these primitives. The table below only shows the actions taken at the higher involved Bluetooth layer. Thus, unless specifically stated, the low-level inquiries and pages needed to discover and connect to Bluetooth devices are not discussed in detail.

service primitive	(highest layer) Bluetooth PDUs involved
serviceBrowse (LIST(*RemDev*) LIST(*RemDevRelation*) LIST(*browseGroup*) *getRemDevName* *stopRule*)	For the subset of *RemDev* that satisfy the *RemDevRelation*, this service primitive will cause the LocDev to send: an *SDP_ServiceSearchRequest* PDU and receives a corresponding response PDU, see section 4.5 in BT_SDP_spec:[7]; an *SDP_ServiceAttributeRequest* PDU and receives a corresponding response PDU, see section 4.6 in BT_SDP_spec:[7]. The first transaction above identifies the SDP servers that contain pertinent service records, while the second transaction retrieves the desired information; Alternatively, the two transactions above are combined to one: LocDev sends an *SDP_ServiceSearchAttributeRequest* PDU and receives a corresponding response PDU, see section 4.7 in BT_SDP_spec:[7] In either of the above cases, the corresponding SDP transaction may last a number of request and response PDU exchanges, due to the L2CAP MTU limitation. If the *getRemDevName* parameter is set to ‘yes’, then for each RemDev involved in the execution of this service primitive, the service primitive will cause a sequence of *LMP_name_request*() LM level PDUs to be sent by the LocDev.* The corresponding RemDev responds with a *LMP_name_response*() LM level PDU containing the requested user-friendly device name.
serviceSearch (LIST(*RemDev*) LIST(*RemDevRelation*) LIST(*searchPattern*, *attributeList*) *getRemDevName* *stopRule*)	same as above

Table 11.1: Bluetooth PDUs related to the service primitives in Section 4.2

service primitive	(highest layer) Bluetooth PDUs involved
enumerateRemDev (LIST(*classOfDevice*) *stopRule*)	This service primitive will cause a Bluetooth baseband *inquiry* process. The inquiry will 'indiscriminately†' find devices residing in the vicinity of the LocDev. Prior to returning the results of this inquiry the LocDev may filter them using the classOfDevice qualifier.
terminatePrimitive (*primitiveHandle returnResults*)	This service primitive will cause the termination of any outstanding operation caused by the invocation of the service primitive identified by the *primitiveHandle* parameter. This may cause an L2CAP connection termination request PDU to be sent from the LocDev to the RemDev, and the subsequent transmission of an L2CAP termination response PDU. It the LocDev is connecting to the RemDev only for the purposes of an SDP transaction, the baseband link will also be severed by the transmission of an LMP_detach LM level PDU.

Table 11.1: Bluetooth PDUs related to the service primitives in Section 4.2

*. If the information requested is already stored (cached) in the LocDev, this service primitive may not have to cause the described LM level PDU transaction.

†. The inquiries considered here use the GIAC. No CoD-specific DIACs have been defined. Nevertheless, the use of appropriate DIACs whenever possible is not excluded and is not outside the scope of this profile.

Part K:3

CORDLESS TELEPHONY PROFILE

This profile defines the features and procedures that are required for interoperability between different units active in the ‘3-in-1 phone’ use case. The scope of this profile includes the following layers/protocols/profiles: Bluetooth Baseband, Link Manager Protocol, L2CAP, Service Discovery Protocol, Telephony Control Protocol Specification (TCS-Binary) and the General Access Profile.

CONTENTS

1 Introduction 100
1.1 Scope 100
1.2 Profile Dependencies 100
1.3 Symbols and conventions 101
1.3.1 Requirement status symbols 101
1.3.2 Signalling diagram conventions 102
1.3.3 Notation for timers and counters 102
2 Profile overview 103
2.1 Profile stack 103
2.2 Configurations and roles 104
2.3 User requirements and scenarios 105
2.4 Profile fundamentals 106
2.5 Feature definitions 106
2.6 Conformance 107
3 Application layer 108
4 TCS-BIN procedures 110
4.1 Connection Management 110
4.1.1 Connecting to a GW 110
4.1.2 Connecting to another TL 110
4.2 Call Control procedures 111
4.2.1 Sides 111
4.2.2 Call class 111
4.2.3 Call request 111
4.2.4 Overlap sending 111
4.2.5 Call proceeding 111
4.2.6 Call confirmation 111
4.2.7 Call connection 112
4.2.8 Non-selected user clearing 112
4.2.9 In-band tones and announcements 112
4.2.10 Failure of call establishment 112
4.2.11 Call clearing 112
4.2.12 Call information 113
4.3 Supplementary services 113
4.3.1 DTMF signalling 113
4.3.2 Calling line identity 113
4.3.3 Register recall 113
4.4 Group Management procedures 114

4.4.1 Obtain Access Rights 114
4.4.2 Configuration distribution 114
4.4.2.1 Link loss detection by GW 114
4.4.3 Periodic key update 115
4.4.4 Fast inter-member access 115
4.5 Connectionless procedures 115
4.6 TCS-BIN Message overview 116
4.7 Information Element overview 117
4.7.1 Bearer capability 118
4.7.2 Called party number 118
4.7.3 Calling party number 118
4.7.4 Cause 119
4.8 Link loss 119
5 Service Discovery procedures 120
6 L2CAP procedures 121
6.1 Channel types 121
6.2 Configuration options 121
6.2.1 Maximum Transmission unit 121
6.2.2 Flush timeout option 121
6.2.3 Quality of Service 121
7 LMP procedures overview 122
7.1 Master-slave switch 123
7.2 Link policy 123
8 LC features 124
8.1 Inquiry scan 125
8.2 Inter-piconet capabilities 125
9 General Access Profile Interoperability Requirements 126
9.1 Modes 126
9.2 Security aspects 126
9.3 Idle mode procedures 127
9.3.1 Bonding 127

10 Annex A (Informative): Signalling flows 128

10.1 Outgoing external call without post-dialling.....128

10.2 Outgoing external call with post-dialling.....129

10.3 Incoming external call, SETUP delivered on connectionless channel130

10.4 Incoming external call, SETUP delivered on connection-oriented channel130

10.5 Call Clearing.....131

10.6 DTMF signalling 131

10.7 DTMF signalling failure 131

10.8 Access rights request.....132

10.9 Configuration distribution 132

10.10 Periodic key update 133

10.11 Fast inter-member access.....133

11 Timers and counters 135

12 References 136

13 List of Figures.....137

14 List of Tables 138

1 INTRODUCTION

1.1 SCOPE

The Cordless Telephony profile defines the protocols and procedures that shall be used by devices implementing the use case called ‘3-in-1 phone’.

The ‘3-in-1 phone’ is a solution for providing an extra mode of operation to cellular phones, using Bluetooth as a short-range bearer for accessing fixed network telephony services via a base station. However, the 3-in-1 phone use case can also be applied generally for wireless telephony in a residential or small office environment, for example for cordless-only telephony or cordless telephony services in a PC – hence the profile name ‘Cordless Telephony’.

This use case includes making calls via the base station, making direct intercom calls between two terminals, and accessing supplementary services provided by the external network.

1.2 PROFILE DEPENDENCIES

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. As indicated in the figure, the Cordless Telephony profile is dependent only upon the Generic access profile. The terminology, user interface and security aspects, modes and procedures as defined in the Generic access profile are applicable to this profile, unless explicitly stated otherwise.

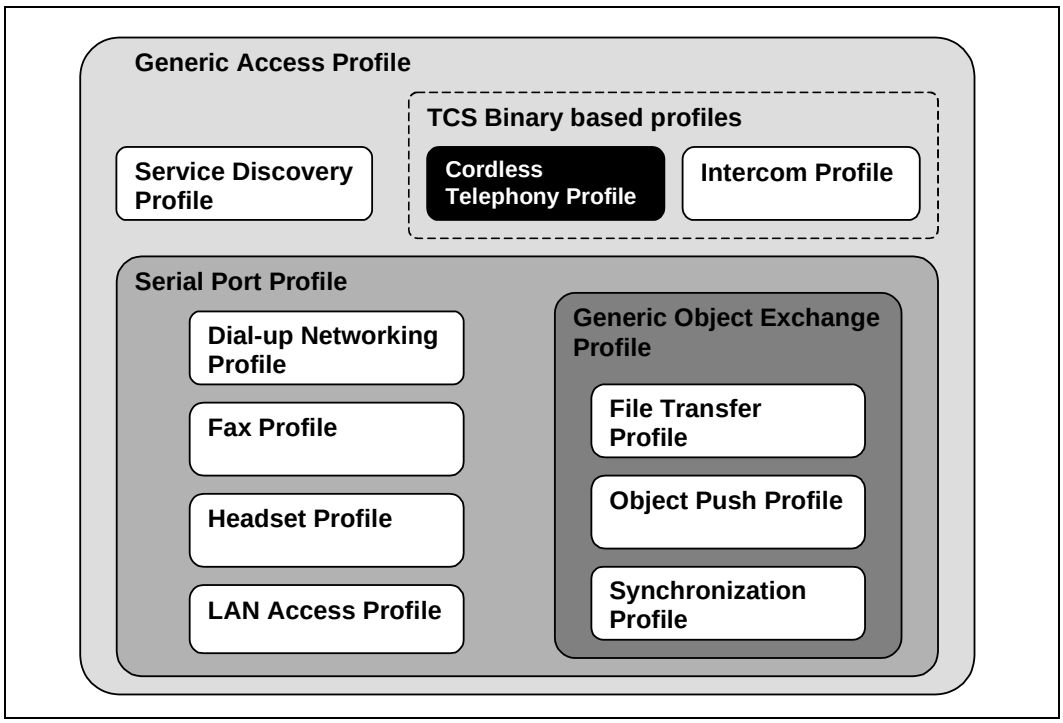


Figure 1.1: Bluetooth Profiles

1.3 SYMBOLS AND CONVENTIONS

1.3.1 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);

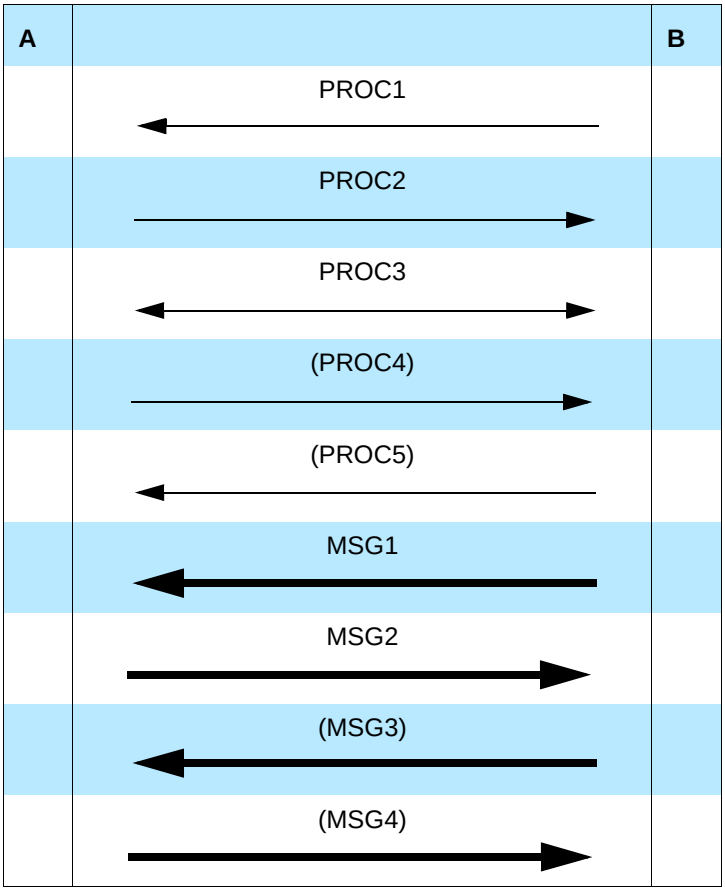
’X’ for excluded (used for capabilities that may be supported by the unit, but which shall never be used in the profile);

’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.

1.3.2 Signalling diagram conventions

The following arrows are used in diagrams describing procedures:



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.

1.3.3 Notation for timers and counters

Timers and counters may be introduced specific to this profile. To distinguish them from timers (counters) used in the Bluetooth protocol specifications and other profiles, these timers (counters) are named in the following format: ‘T_{CTP-}*nnn*’ (’N_{CTP}*nnn*’).

2 PROFILE OVERVIEW

2.1 PROFILE STACK

Figure 2.1 below shows the protocols as used within this profile:

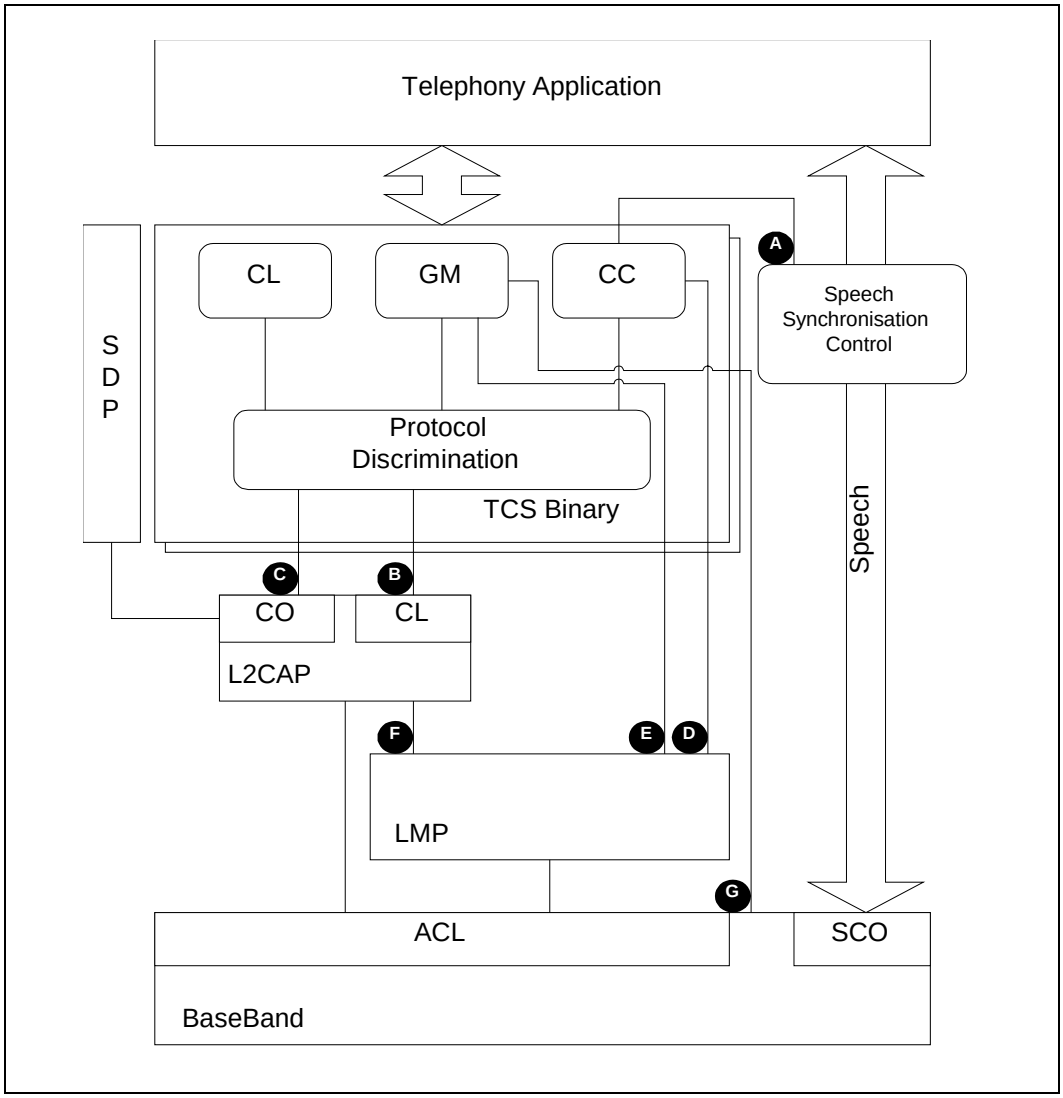


Figure 2.1: Protocol model

This profile will define the requirements for each of the layers in the model above for the Cordless Telephony profile.

In the profile, the interfaces in Figure 2.1 above are used for the following purposes:

A) The Call Control entity uses this interface to the speech synchronization control to connect and disconnect the internal speech paths.

B) This interface is used by the GW to send and by the TL to receive broadcast TCS-Binary messages.

C) This interface is used to deliver all TCS messages that are sent on a connection-oriented (point-to-point) L2CAP channel.

D) This interface is used by the Call Control entity to control the Link Manager directly for the purpose of establishing and releasing SCO links.

E) This interface is used by the Group Management to control Link Manager functions when initializing and for key handling purposes.

F) This interface is not within the scope of this profile.

G) This interface is used by the Group Management entity to control the LC/Baseband directly to enable inquiry, paging, inquiry scan and page scan.

2.2 CONFIGURATIONS AND ROLES

The following two roles are defined for this profile:

Gateway (GW) – The GW acts as a terminal endpoint from the external network point of view and handles all interworking towards that network. The GW is the central point with respect to external calls, which means that it handles all call set-up requests to/from the external network. Examples of devices that can act as a gateway include a PSTN home base station, an ISDN home base station, a GSM gateway, a satellite gateway and an H.323 gateway.

With respect to this profile, the gateway may have the functionality to support multiple terminals being active at once, or be of a simple kind where only one terminal may be active. The simple gateway will not support multiple ringing terminals, multiple active calls or services involving more than one terminal simultaneously.

Terminal (TL) – The TL is the wireless user terminal, which may for example be a cordless telephone, a dual-mode cellular/cordless phone or a PC. Note that the scope of this profile with respect to a dual-mode cellular/cordless phone acting as TL is only the cordless mode.

The Cordless Telephony profile supports a topology of one gateway (GW) and a small number (≤7) of terminals (TLs)[1]. Figure 2.2 below shows an example of the considered architecture:

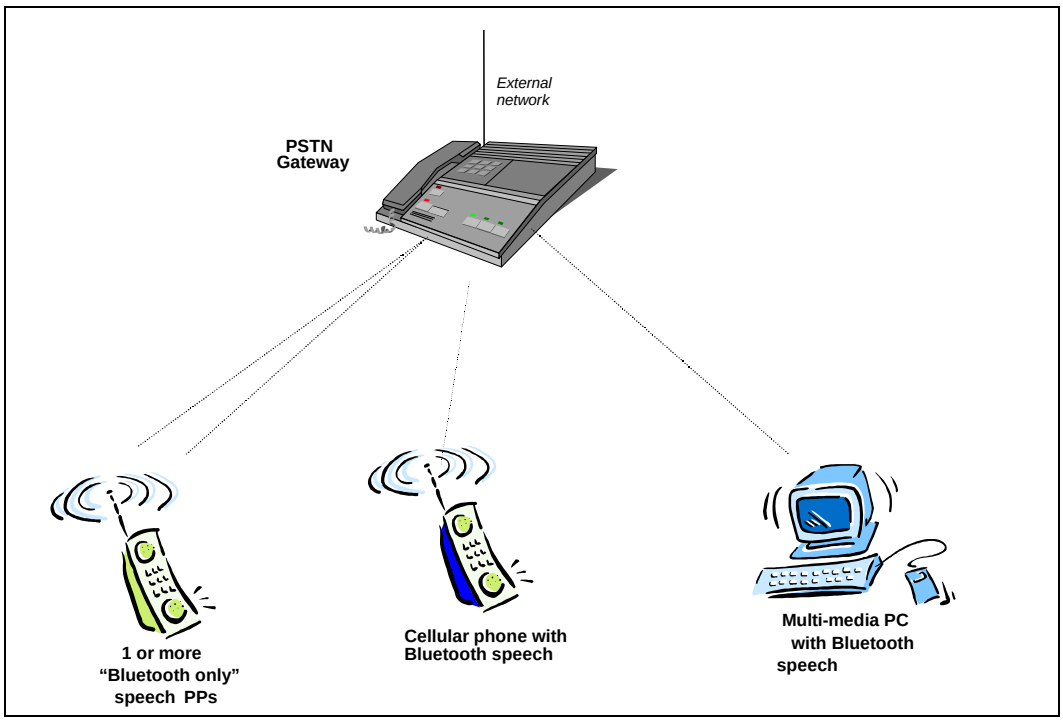


Figure 2.2: System configuration example

2.3 USER REQUIREMENTS AND SCENARIOS

The following scenarios are covered by this profile:

1. Connecting to the gateway so that incoming calls can be routed to the TL and outgoing calls can be originated.
2. Making a call from a TL to a user on the network that the gateway is connected to.
3. Receiving a call from the network that the gateway is connected to.
4. Making direct calls between two terminals.
5. Using supplementary services provided by the external network by means of DTMF signalling and register recall (hook flash).

1. Optionally, more terminals may be supported.

2.4 PROFILE FUNDAMENTALS

The GW is normally the master of the piconet in the Cordless Telephony profile. As master, the GW will control the power mode of the TLs and may broadcast information to the TLs.

A TL that is out of range of a GW searches for it by periodically trying to page it. A GW shall devote as much of its free capacity as possible (considering power limitations and ongoing signalling) to page scanning in order to allow roaming TLs that enter the range of the GW to find it as quickly as possible. This scheme minimizes ‘air pollution’ and gives reasonable access time when coming into range of the GW. When a TL has successfully paged a GW, a master-slave switch shall be performed since the GW shall be the master. A connection-oriented L2CAP channel and, possibly, a L2CAP connectionless channel are established to be used for all TCS signalling during that Cordless Telephony session.

A TL that is within range of a GW shall normally be in park mode when it is not engaged in calls. This mode is power-efficient, allows for reasonable call set-up times and allows broadcasting to the attached TLs.

Upon arrival of an incoming call, or when a TL wants to make an outgoing call, the GW shall be put in active mode. The L2CAP channels (see above) are used for all TCS control signalling. Voice is transported using SCO links.

For security purposes, authentication of TLs and GW is used, and all user data is encrypted. To facilitate secure communication between cordless units, the WUG concept (see TCS Binary, Section 3) is used. The GW always acts as WUG master.

2.5 FEATURE DEFINITIONS

Calling line identification presentation (CLIP) – The ability to provide the calling party number to the called party before accepting the call.

Call information – The ability to provide additional information during the active phase of a call.

Connection Management – The ability to accept and (TLs only) request connections for the purposes of TCS-Bin procedures.

DTMF signalling – The ability, in external calls, to send a DTMF signal over the external network to the other party.

Incoming external call – A call originating from the external network connected to the GW.

Initialization – The infrequent process whereby a TL receives access rights to a certain GW.

Intercom call – A call originating from a TL towards another TL.

Multi-terminal support –

1. In the GW, the ability to handle multiple active terminals being registered at the same time[2]
2. In the TL, the support for a Wireless User Group (WUG)

On hook – The ability to indicate the action of going on-hook (e.g. to terminate a call), and release of all radio resources related to that call.

Outgoing external call – A call originated by a TL towards the external network connected to the GW.

Post-dialling – The ability to send dialling information after the outgoing call request set-up message is sent.

Register recall – The ability of the TL to request ‘register recall’, and of the GW to transmit the request to the local network. Register recall means to seize a register (with dial tone) to permit input of further digits or other actions. In some markets, this is referred to as ‘hook flash’.

2.6 CONFORMANCE

If conformance to this profile is claimed, all capabilities indicated as mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies to 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 certification program.

Note that the Intercom Profile is used for intercom calls. This means that a TL claiming conformance to the Cordless Telephony profile must conform to Intercom Profile.

2. Note that a GW may support multiple active terminals but not a Wireless User Group (WUG).

3 APPLICATION LAYER

The following text, together with the associated sub-clauses, defines the feature requirements with regard to this profile.

Table 3.1 shows the feature requirements made by this profile.

Item no.	Feature	Support in TL	Support in GW
1.	Connection Management	M	M
2.	Outgoing external call	M	M
3.	Incoming external call	M	M
4.	Intercom call	M	N/A
5.	On hook	M	M
6.	Post-dialling	O	O
7.	Multi-terminal support	O	O
8.	Call information	O	O
9.	Calling line identification presentation (CLIP)	M	O
10.	DTMF signalling	M	M
11.	Register recall	M	M

Table 3.1: Application layer features

Table 3.2 maps each feature to the procedures used for that feature, and shows if the procedure is optional, mandatory or conditional for that feature. The procedures are described in the referenced section.

Feature	Procedure	Ref.	Support in TL	Support in GW
1. Connection Management	Connecting to a GW	4.1.1	M	M
	Connecting to a TL	4.1.2	M	N/A
2. Outgoing external call	Call request	4.2.3	M	M
	Overlap sending	4.2.4	C2	C2
	Call proceeding	4.2.5	C2	C2
	Call confirmation	4.2.6	M	O
	Call connection	4.2.7	M	M
	In-band tones and announcements	4.2.9	M	O

Table 3.2: Application layer feature to procedure mapping

Feature	Procedure	Ref.	Support in TL	Support in GW
3. Incoming external call	Call request	4.2.3	M	M
	Call confirmation	4.2.6	M	M
	Call connection	4.2.7	M	M
	Non-selected user clearing	4.2.8	M	M
	In-band tones and announcements	4.2.9	M	O
4. Intercom call	NOTE 1			
5. On hook	Call clearing	4.2.11	M	M
6. Post-dialling	Overlap sending	4.2.4	M	M
	Call proceeding	4.2.5	M	M
7. Multi-terminal support	Obtain access rights	4.4.1	M	O
	Configuration distri-bution	4.4.1	M	O
	Fast inter-member access	4.4.4	M	O
	Periodic key update	4.4.3	M	O
8. Call information	Call information	4.2.12	M	M
9. Calling line identification presentation (CLIP)	Calling line identity	4.3.2	M	M
10. DTMF signalling	DTMF signalling	4.3.1	M	M
11. Register recall	Register recall	4.3.3	M	M
C2: IF feature 6 THEN M else N/A				

Table 3.2: Application layer feature to procedure mapping

Note 1: For intercom calls, the intercom profile is used. Before initiating the intercom call, the TL which is initiating the call may optionally use the fast inter-member access procedure to speed up the call set-up.

4 TCS-BIN PROCEDURES

The following text together with the associated sub-clauses defines mandatory requirements with regard to this profile.

When describing TCS-BIN procedures, this section provides additional information concerning lower layer handling. The normative reference for TCS-BIN procedures is TCS Binary.

Annex A contains signalling flows that illustrate the procedures in this section.

4.1 CONNECTION MANAGEMENT

4.1.1 Connecting to a GW

When a TL connects to the GW, the link is configured and the L2CAP connection that is used for further signalling during that TCS-BIN session is set up and configured. The TL which is connecting is responsible for setting up the connection-oriented L2CAP channel.

Only trusted TLs are allowed to connect to the GW.

Note that, in order to avoid the paging delay at call set-up and to enable broadcasted messages, the TL establishes a L2CAP connection to the GW when it comes into range, and not before every call. This L2CAP connection remains until the radio link is lost or the TL stops being active in this profile. This means that the L2CAP connections used may be idle (i.e. not used to transfer data) for long periods of time.

A GW supporting feature 7, ‘Multi-terminal support’, uses a connectionless L2CAP channel for TCS-BIN broadcasted messages. A TL is added to the connectionless group when it connects to the GW.

4.1.2 Connecting to another TL

In the case of an intercom call, the TL which initiates the call establishes a direct link to the other TL. See the Intercom Profile for a description of these procedures.

If the TL has the capability to participate in two piconets at the same time, the TL may remain a member of the GW piconet and participate in signalling towards the GW during the intercom call.

If the TL does not have the capability to participate in two piconets at the same time, it must detach from the GW while the intercom call is active. After the intercom call is finished, the TL must re-establish the connection to the GW.

4.2 CALL CONTROL PROCEDURES

4.2.1 Sides

This section describes which sides shall be assumed for the purpose of reading TCS Binary.

In an outgoing external call, the TL is the outgoing side and the GW is the incoming side. In an incoming external call, the TL which terminates the call is the incoming side and the GW is the outgoing side.

Refer to the Intercom Profile for the sides assumed in intercom calls.

4.2.2 Call class

This section describes the usage of call classes in the Cordless Telephony profile.

An *external call* is a call between a TL and a third party connected via an external network (PSTN, ISDN, GSM or other). The call class used in SETUP messages for external calls (outgoing and incoming) is ‘external call’.

An *intercom call* is a call between two TLs, which may be setup with GW support if the two TLs are members of the same WUG. Refer to Intercom Profile for call class usage in intercom calls.

4.2.3 Call request

This procedure shall be performed as defined in TCS Binary.

4.2.4 Overlap sending

This procedure shall be performed as defined in TCS Binary.

4.2.5 Call proceeding

This procedure shall be performed as defined in TCS Binary.

4.2.6 Call confirmation

This procedure shall be performed as defined in TCS Binary.

If the call is an incoming external call, and the SETUP message was delivered on a connection-oriented channel, the incoming side must acknowledge the SETUP message by performing the call confirmation procedure.

4.2.7 Call connection

This procedure shall be performed as defined in TCS Binary. The following text defines the mandatory requirements with regard to this profile.

If the bearer capability for this call is ‘Synchronous Connection-Oriented’, the SCO link establishment sub-procedure (see LMP, Section 3.21) shall be initiated before sending a CONNECT.

If the bearer capability for this call is ‘Synchronous Connection-Oriented’, the audio path shall be connected to by a unit when it receives a CONNECT or CONNECT ACKNOWLEDGE.

4.2.8 Non-selected user clearing

This procedure shall be performed as defined in TCS Binary. Additionally, the text in 4.2.11 defines the mandatory requirements with regard to this profile concerning call clearing.

4.2.9 In-band tones and announcements

This procedure shall be performed as defined in TCS Binary. The following text defines the mandatory requirements with regard to this profile.

Only the GW may provide in-band tones and announcements. The SCO link establishment sub-procedure (see Link Manager Protocol, Section 3.21) is initiated before sending a Progress Indicator information element #8, “In-band information or appropriate pattern is now available”.

The audio path shall be connected to by a TL when it receives a Progress Indicator information element #8, “In-band information or appropriate pattern is now available”.

4.2.10 Failure of call establishment

This procedure shall be performed as defined in TCS Binary. Additionally, the text in 4.2.11 defines the mandatory requirements with regard to this profile concerning call clearing.

4.2.11 Call clearing

All call clearing and call collision procedures as defined in TCS Binary shall be supported by both GW and TL. For a specification of the complete behavior, see TCS Binary. This section describes how the lower layers are used to release circuit switched (SCO) connections.

A unit shall release the SCO link by invoking the appropriate LMP sub-procedure (see Link Manager Protocol, Section 3.21) when a unit has received a RELEASE message.

A unit shall release the SCO link (if not already released) by invoking the appropriate LMP sub-procedure (see Link Manager Protocol, Section 3.21) when it has received a RELEASE COMPLETE message.

4.2.12 Call information

This procedure shall be performed as defined in TCS Binary.

4.3 SUPPLEMENTARY SERVICES

Supplementary services can be either internal services within the WUG, or external services provided by the network the GW is connected to.

The exact set of external supplementary services is not defined in this profile and is dependent on the network the GW is connected to. This profile provides the means for accessing them; for example through the use of DTMF signalling and register recall.

The required support for internal services and DTMF signalling is defined in the following sub-clauses.

4.3.1 DTMF signalling

The capability to request DTMF signalling towards the external network is mandatory for the TL. The capability to accept DTMF signalling requests is mandatory for the GW.

Depending on the network the GW is connected to, it shall translate the DTMF messages to the appropriate in-band or out-of-band signalling. If the network has no DTMF signalling capability, or if the GW for some reason is unable to perform DTMF signalling towards the external network, the GW shall reject the request for DTMF signalling as described below. In the START DTMF REJECT message, the GW shall use Cause #29, “Facilities rejected”.

4.3.2 Calling line identity

This procedure shall be performed as defined in TCS Binary.

It is recommended that all GWs that are connected to networks that provide calling line identity have the capability to provide this information to the user.

4.3.3 Register recall

This procedure shall be performed as defined in TCS Binary.

4.4 GROUP MANAGEMENT PROCEDURES

4.4.1 Obtain Access Rights

This procedure shall be performed as defined in TCS Binary.

A TL which wants to become member of a WUG may initiate this procedure towards a GW. The GW may accept or reject the request depending, for example, on configuration, or if the user has physical access to the base.

A GW which accepts the access rights request shall add the TL to the WUG and initiate the Configuration distribution procedure.

4.4.2 Configuration distribution

This procedure shall be performed as defined in TCS Binary.

Because of the security implications of this procedure, a TL is not forced to store the key information received during this procedure. In addition, GW may always reject the ACCESS RIGHTS REQUEST from a TL because of implementation-dependent reasons. For example, the user may be required to press a button on the GW before being granted access to the group.

Note that for intercom calls, two TLs that are members of the WUG do not need to perform the initialization procedure described in the Intercom profile (see Intercom Profile) if they use the keys distributed in the Configuration distribution procedure.

A TL which stores link keys during the Configuration Distribution procedure shall never overwrite existing link keys to other WUG members. Only if there was previously no link key to a specific device shall the key obtained during the Configuration Distribution procedure be used.

In addition to the link-loss handling described in Section 4.8, Section 4.4.2.1 applies for this procedure.

4.4.2.1 Link loss detection by GW

If the GW detects loss of link before receiving the INFO ACCEPT message, it shall consider the WUG update to be terminated unsuccessfully and consider the TL detached. If the GW detects loss of link after receiving the INFO ACCEPT message, it shall consider the WUG update to be terminated successfully.

4.4.3 Periodic key update

The K_{master} to be used during a GW-TL connection is issued to the TL when connecting to a GW. The K_{master} is intended to be a key valid for a single session only, but since the GW piconet is operational all the time, this would mean that the same K_{master} would always be used. In order to increase the security level, the K_{master} is changed periodically.

Timer $T_{CTP}400$ determines the interval between key changes. When $T_{CTP}400$ expires, the GW tries to do a periodic key update on all TLs. However, some TLs may be out of range or powered off, or the procedure may fail for some other reason. The new key in these cases is given to the TL when it attaches the next time. After there has been an attempt to update all TLs, $T_{CTP}400$ is reset.

The periodic key update for one TL is performed as follows. First, if the TL was parked, it is unparked. Then, the new link key is issued. After this, the new link key is activated by turning encryption off and back on. Finally, the TL may be parked.

If any of the sub-procedures fails, further sub-procedures will not be performed on that TL. The GW shall proceed with updating the next TL.

4.4.4 Fast inter-member access

The Fast inter-member access procedure is used when two TLs that are members of the same WUG need to establish a piconet of their own. This may be needed when an intercom call shall be established. Refer to TCS Binary for a definition of the procedure.

The TL_T may detach from the GW after having sent the LISTEN ACCEPT message by terminating the L2CAP channel to the GW and sending a LMP_detach.

The TL_I may detach from the GW after having received the LISTEN ACCEPT message by terminating the L2CAP channel to the GW and sending a LMP_detach.

4.5 CONNECTIONLESS PROCEDURES

TCS-BIN Connectionless (CL) messaging is not within the scope of the Cordless Telephony profile.

4.6 TCS-BIN MESSAGE OVERVIEW

This section defines the allowed TCS-BIN messages in the Cordless Telephony profile.

Message	**Ability to Send**		**Ability to Receive**	
	TL	**GW**	**TL**	**GW**
Access rights accept	N/A	O	C1	N/A
Access rights reject	N/A	O	C1	N/A
Access rights request	C1	N/A	N/A	O
Alerting	M	O	M	M
Call Proceeding	C2	C2	M	M
Connect	M	M	M	M
Connect Acknowledge	M	M	M	M
Disconnect	M	M	M	M
Info suggest	N/A	O	C1	N/A
Info accept	C1	N/A	N/A	O
Information	M	O	O	M
Listen request	C1	N/A	N/A	O
Listen suggest	N/A	O	C1	N/A
Listen accept	C1	O	C1	O
Listen reject	C1	O	C1	O
Progress	N/A	O	M	N/A
Release	M	M	M	M
Release Complete	M	M	M	M
Setup	M	M	M	M
Setup Acknowledge	N/A	O	O	N/A
Start DTMF	M	N/A	N/A	M
Start DTMF Acknowledge	N/A	M	M	N/A
Start DTMF Reject	N/A	M	M	N/A
Stop DTMF	M	N/A	N/A	M
Stop DTMF Acknowledge	N/A	M	M	N/A
C1: IF feature 7 THEN M else N/A C2: IF feature 6 THEN M else N/A				

Table 4.1: TCS-BIN messages

4.7 INFORMATION ELEMENT OVERVIEW

This section together with the associated sub-clauses defines the allowed information elements used in TCS-BIN messages in the Cordless Telephony profile.

Information Element	**Ability to Send**		**Ability to Receive**	
	TL	**GW**	**TL**	**GW**
Message type	M	M	M	M
Audio control	N/A	N/A	N/A	N/A
Bearer capability	M	M	M	M
Call class	M	M	M	M
Called party number	M	O	O	M
Calling party number	O	C2	M	O
Cause	M	M	M	M
Clock offset	C1	O	C1	O
Company-specific	O	O	O	O
Configuration data	N/A	O	C1	N/A
Destination CID	N/A	N/A	N/A	N/A
Keypad facility	M	N/A	N/A	M
Progress indicator	N/A	O	M	N/A
SCO handle	M	M	M	M
Sending complete	M	N/A	N/A	M
Signal	N/A	M	M	N/A
C1: IF feature 7 THEN M else N/A C2: IF feature 9 THEN M else N/A				

Table 4.2: TCS-BIN information elements

The following subsections define restrictions that apply to the contents of the TCS-BIN information elements in the Cordless Telephony profile. Note that in the tables, only fields where restrictions apply are shown. If a field is not shown in a table, it means that all values defined in TCS Binary for that field are allowed.

For those information elements not listed below, no restrictions apply.

4.7.1 Bearer capability

The following restrictions apply to the contents of the Bearer capability information element:

Field	Values allowed
Link type	SCO, None

Table 4.3: Restrictions to contents of Bearer capability information element

4.7.2 Called party number

Maximum information element length is 27 octets, thus allowing a maximum of 24 number digits.

4.7.3 Calling party number

Maximum information element length is 28 octets, thus allowing a maximum of 24 number digits.

4.7.4 Cause

The following restrictions apply to the contents of the Cause information element:

Field	Values allowed
Cause value	#1 – “Unassigned (unallocated number)” #3 – “No route to destination” #16 – “Normal call clearing” #17 – “User busy” #18 – “No user responding” #19 – “No answer from user (user alerted)” #21 – “Call rejected by user” #22 – “Number changed” #26 – “Non-selected user clearing” #28 – “Invalid number format (incomplete number” #29 – “Facilities rejected” #34 – “No circuit/channel available” #41 – “Temporary failure” #44 – “Requested circuit/channel not available” #58 – “Bearer capability not presently available” #65 – “Bearer capability not implemented” #69 – “Requested facility not implemented” #102 – “Recovery on timer expiry”

Table 4.4: Restrictions to contents of Cause information element

4.8 LINK LOSS

If a unit in a CC state other than *Null* detects loss of link, it shall immediately go to the *Null* state. Release procedures shall in this case not be performed.

A unit in any GM state which detects loss of link shall consider itself to be in the null state. Any ongoing GM procedure shall immediately be aborted and considered to be terminated unsuccessfully.

5 SERVICE DISCOVERY PROCEDURES

Table 5.1 below lists all entries in the SDP database of the GW defined by this profile. The ‘Status’ column indicates whether the presence of this field is mandatory or optional.

The codes assigned to the mnemonic’s used in the ‘Value’ column, and the codes assigned to the attribute identifiers, can be found in the Bluetooth Assigned Numbers.

Item	Definition:	Type:	Value:	Status	Default
Service Class ID List				M	
Service Class #0		UUID	Generic Telephony	O	
Service Class #1		UUID	Cordless Tele-phony	M	
Protocol Descriptor List				M	
Protocol #0		UUID	L2CAP	M	
Protocol #1		UUID	TCS-BIN-CORD-LESS	M	
Service Name	Display-able Text name	String	Service-provider defined	O	’Cord-less Tele-phony’
External Network		UInt8	0x01=PSTN 0x02=ISDN 0x03=GSM 0x04=CDMA 0x05=Analogue cellular 0x06=Packet-switched 0x07=Other	O	
BluetoothProfile-DescriptorList				M	
Profile #0		UUID	Cordless Tele-phony	M	
Parameter for Profile #0	Version	UInt16	0x0100*	O	0x100

Table 5.1: SDP entry for GW service

*. Indicating version 1.0

6 L2CAP PROCEDURES

The following text, together with the associated sub-clauses, define the mandatory requirements with regard to this profile.

6.1 CHANNEL TYPES

In this profile, both connection-oriented channels and connectionless channels are used.

Connectionless channels are used to broadcast information from the GW to the TLs. Only the GW shall use connectionless channels for sending. Refer to the Bluetooth Security Architecture White paper for information on the security implications of using L2CAP connectionless traffic.

In this profile, only the TL may initiate the establishment of connection-oriented channels. When connecting to the GW, the TL shall use the value 0x0007 (TCS-BIN-CORDLESS) in the PSM field of the Connection Request packet. For PSM usage in intercom calls, see Intercom Profile.

6.2 CONFIGURATION OPTIONS

This section describes the usage of configuration options in the Cordless Telephony Profile.

6.2.1 Maximum Transmission unit

The minimum MTU that a L2CAP implementation used for this profile should support is 171 octets. This means that the maximum number of TLs supported by this profile is 7.

6.2.2 Flush timeout option

The flush timeout value used for both the GW and the TL shall be the default value of 0xFFFF.

6.2.3 Quality of Service

Negotiation of Quality of Service is optional.

7 LMP PROCEDURES OVERVIEW

In this section the LMP layer is discussed. In the table below, all LMP features are listed. In the table it is shown what LMP features are mandatory to support with respect to the Cordless Telephony profile, which are optional and which are excluded. The reason for excluding features is that they may degrade operation of devices in this profile. Therefore, these features shall never be activated by a unit active in this profile.

	Procedure	**Support in LMP**	**Support in TL**	**Support in GW**
1.	Authentication	M		
2.	Pairing	M		
3.	Change link key	M		
4.	Change the current link key	M		
5.	Encryption	O	M	M
6.	Clock offset request	M		
7.	Slot offset information	O		
8.	Timing accuracy information request	O		
9.	LMP version	M		
10.	Supported features	M		
11.	Switch of master slave role	O	M	C1
12.	Name request	M		
13.	Detach	M		
14.	Hold mode	O		
15.	Sniff mode	O		
16.	Park mode	O	M	M
17.	Power control	O		
18.	Channel-quality driven DM/DH	O		
19.	Quality of service	M		
20.	SCO links	O	M	M
21.	Control of multi-slot packets	O		
22.	Paging scheme	O		
23.	Link supervision	M		
24.	Connection establishment	M		
C1: IF feature 7 THEN M else N/A				

Table 7.1: LMP procedures

7.1 MASTER-SLAVE SWITCH

A GW supporting feature 7, ‘Multi-terminal support’, must always be the master of the piconet. Such a GW will request a master-slave switch when a TL connects. If the TL rejects the request, the GW may detach it. Thus, a TL which does not accept master-slave switch requests can not be guaranteed service by all GWs.

7.2 LINK POLICY

The GW shall be as conservative as possible when deciding what power mode to put the TLs in. This means that when a TL is not engaged in signalling, the GW shall put it in a low-power mode. The recommended low-power mode to use is the park mode.

The low-power mode parameters shall be chosen such that the TL can always return to the active state within 300 ms.

If the GW can save power during a call, it may use the sniff mode. A TL may request to be put in the sniff mode.

8 LC FEATURES

The following table lists all features on the LC level.

	Procedure	Support in TL	Support in GW
1.	Inquiry		X
2.	Inquiry scan	X	
3.	Paging		X
4.	Page scan		
A	Type R0		
B	Type R1		
C	Type R2		
5.	Packet types		
A	ID packet		
B	NULL packet		
C	POLL packet		
D	FHS packet		
E	DM1 packet		
F	DH1 packet		
G	DM3 packet		
H	DH3 packet		
I	DM5 packet		
J	DH5 packet		
K	AUX packet	X	X
L	HV1 packet		
M	HV2 packet		
N	HV3 packet	M	M
O	DV packet	X	X
6.	Inter-piconet capabilities	O	C1
7.	Voice codec		
A	A-law		
B	μ-law		
C	CVSD	M	M
C1: IF feature 7 THEN M else O			

Table 8.1: LC features

8.1 INQUIRY SCAN

A device which is active in the GW role of the Cordless Telephony profile shall, in the Class of Device field:

1. Set the ‘Telephony’ bit in the Service Class field
2. Indicate ‘Phone’ as Major Device class

This may be used by an inquiring device to filter the inquiry responses.

8.2 INTER-PICONET CAPABILITIES

Inter-piconet capability is the capability, as master, to keep the synchronization of a piconet while page scanning in free slots and allowing for new members to join the piconet. While a new unit is joining the piconet (until the master-slave switch has been performed), operation may temporarily be degraded for the other members.

A GW which supports feature 7, ‘Multiple terminal support’, shall have inter-piconet capabilities. The TL may have inter-piconet capabilities.

9 GENERAL 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 Generic Access Profile.

9.1 MODES

The table shows the support status for Modes within this profile.

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

Table 9.1: Modes

9.2 SECURITY ASPECTS

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

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

Table 9.2: Security aspects

9.3 IDLE MODE PROCEDURES

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

	Procedure	Support in TL	Support in GW
1	General inquiry	M	N/A
2	Limited inquiry	O	N/A
3	Name discovery	O	N/A
4	Device discovery	O	N/A
5	Bonding	M	M

Table 9.3: Idle mode procedures

9.3.1 Bonding

It is mandatory for the TL to support initiation of bonding, and for the GW to accept bonding.

10 ANNEX A (INFORMATIVE): SIGNALLING FLOWS

This annex contains signalling diagrams that are used to clarify the interworking between units. This annex is informative only. The diagrams do not represent all possible signalling flows as defined by this profile.

10.1 OUTGOING EXTERNAL CALL WITHOUT POST-DIALLING

The following sequence shows the successful case when the TL does not use overlap sending:

```
TL                                                  GW
                    SETUP
          ========================>
              (CALL PROCEEDING)
          <========================
                  (ALERTING)
          <========================
            SCO LINK ESTABLISHMENT
           <---------------------------------------
                    CONNECT
          <========================
            CONNECT ACKNOWLEDGE
          ========================>
```

Figure 10.1: TL-originated call when overlap sending is not used

10.2 OUTGOING EXTERNAL CALL WITH POST-DIALLING

The following sequence shows the successful case when post-dialling is used.

TL		GW
	SETUP ==========>	
	SETUP ACKNOWLEDGE <==========	
This message is repeated until the GW has enough dialling information	INFORMATION ==========>	
	(CALL PROCEEDING) <==========	When the GW has sufficient information to complete the call, CALL PROCEEDING, ALERTING or CONNECT is sent.
	(ALERTING) <==========	
	SCO LINK ESTABLISHMENT <----------	
	CONNECT <==========	
	CONNECT ACKNOWLEDGE ==========>	

Figure 10.2: Outgoing external call with post-dialling

10.3 INCOMING EXTERNAL CALL, SETUP DELIVERED ON CONNECTIONLESS CHANNEL

The figure below shows the allowed signalling flow in the successful case:

TL	GW
SETUP <==========	
(UNPARK) ---------->	
SCO LINK ESTABLISHMENT ---------->	
CONNECT ==========>	
CONNECT ACKNOWLEDGE <==========	

Figure 10.3: Incoming external call, SETUP delivered on connectionless channel

10.4 INCOMING EXTERNAL CALL, SETUP DELIVERED ON CONNECTION-ORIENTED CHANNEL

The figure below shows the allowed signalling flow in the successful case:

TL	GW
SETUP <==========	
(UNPARK) ---------->	
ALERTING ==========>	
SCO LINK ESTABLISHMENT ---------->	
CONNECT ==========>	
CONNECT ACKNOWLEDGE <==========	

Figure 10.4: Incoming external call, SETUP delivered on connection-oriented channel

10.5 CALL CLEARING

The figure below shows the allowed signalling flow in the successful case when the TL initiates call clearing:

```
TL                                                    GW
                    DISCONNECT
            ========================>
                     RELEASE
            <========================
                 SCO LINK RELEASE
            <----------------------------------------
                 RELEASE COMPLETE
            ========================>
```

Figure 10.5: Call Clearing signalling flow, successful case

10.6 DTMF SIGNALLING

The figure below shows the allowed signalling flow in the successful case:

```
TL                                                    GW
                    START DTMF
            ========================>
            START DTMF ACKNOWLEDGE
            <========================
                    STOP DTMF
            ========================>
            STOP DTMF ACKNOWLEDGE
            <========================
```

Figure 10.6: DTMF signalling, successful case

10.7 DTMF SIGNALLING FAILURE

The figure below shows the allowed signalling flow in the unsuccessful case:

```
TL                                                    GW
                    START DTMF
            ========================>
                 START DTMF REJECT
            <========================
```

Figure 10.7: DTMF signalling, unsuccessful case

10.8 ACCESS RIGHTS REQUEST

The figure below shows the allowed signalling flow in the successful case:

TL		GW
	ACCESS RIGHTS REQUEST ==========>	
	ACCESS RIGHTS ACCEPT <==========	

Figure 10.8: Signalling diagram for Access Rights Request

10.9 CONFIGURATION DISTRIBUTION

The figure below shows the allowed signalling flow in the successful case:

TL		GW
	(UNPARK) <----------	
	LMP_USE_SEMI_PERMANENT_KEY <----------	For additional security, GW uses a point-to-point key to distribute WUG info.
	START ENCRYPTION <----------	
	INFO SUGGEST <==========	
	INFO ACCEPT ==========>	
	CHANGE CURRENT LINK KEY <----------	GW switches back to a temporary key
	START ENCRYPTION <----------	
	(PARK) <----------	

Figure 10.9: Signalling diagram for Configuration distribution

10.10 PERIODIC KEY UPDATE

The figure below shows the allowed signalling flow in the successful case:

```
TL                                                GW
                    (UNPARK)
            <---------------------------------------
        CHANGE THE CURRENT LINK KEY
            <---------------------------------------
             TURN OFF ENCRYPTION
            <---------------------------------------
              TURN ON ENCRYPTION
            <---------------------------------------
                    (PARK)
            <---------------------------------------
```

Figure 10.10: Signalling diagram for periodic key update

10.11 FAST INTER-MEMBER ACCESS

The figure below shows the allowed signalling flow in the successful case:

```
TLO                                               GW
                    (UNPARK)
            ---------------------------------------->
                 LISTEN REQUEST
            ========================>
                 LISTEN ACCEPT
            <========================
            (L2CAP CHANNEL RELEASE)
            ------------------------------------->
                    (DETACH)
            ------------------------------------->
```

Figure 10.11: Signalling diagram for Fast inter-member access, originating side

The figure below shows the valid sub-procedure sequence between the TL_T and GW:

TL_T		**GW**
	(UNPARK) <----------	
	LISTEN SUGGEST <==========	
(as soon as possible after sending the LISTEN ACCEPT, the TL_T starts page scanning)	LISTEN ACCEPT ==========>	
	(L2CAP CHANNEL RELEASE) ---------->	
	(DETACH) ---------->	

Figure 10.12: Signalling diagram for Fast inter-member access, terminating side

11 TIMERS AND COUNTERS

Timer name	Proposed value	Description	Comment
$T_{CTP}400$	1 week	Time between periodic key updates, depending on the required security level	

Table 11.1: Defined timers

12 REFERENCES

[1] Bluetooth Baseband Specification

[2] Bluetooth Link Manager Protocol

[3] Bluetooth Logical Link Control and Adaptation Protocol Specification

[4] Bluetooth Telephony Control Protocol Specification

[5] Bluetooth Service Discovery Protocol

[6] Bluetooth Intercom Profile

[7] Bluetooth Assigned Numbers

[8] Thomas Müller, Security Architecture Whitepaper, version 0.5

13 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles 101
Figure 2.1: Protocol model 103
Figure 2.2: System configuration example 105
Figure 10.1: TL-originated call when overlap sending is not used 128
Figure 10.2: Outgoing external call with post-dialling 129
Figure 10.3: Incoming external call, SETUP delivered on connectionless channel 130
Figure 10.4: Incoming external call, SETUP delivered on connection-oriented channel 130
Figure 10.5: Call Clearing signalling flow, successful case 131
Figure 10.6: DTMF signalling, successful case 131
Figure 10.7: DTMF signalling, unsuccessful case 131
Figure 10.8: Signalling diagram for Access Rights Request 132
Figure 10.9: Signalling diagram for Configuration distribution 132
Figure 10.10: Signalling diagram for periodic key update 133
Figure 10.11: Signalling diagram for Fast inter-member access, originating side 133
Figure 10.12: Signalling diagram for Fast inter-member access, terminating side 134

14 LIST OF TABLES

Table 3.1: Application layer features 108
Table 3.2: Application layer feature to procedure mapping 108
Table 4.1: TCS-BIN messages 116
Table 4.2: TCS-BIN information elements 117
Table 4.3: Restrictions to contents of Bearer capability information element 118
Table 4.4: Restrictions to contents of Cause information element 119
Table 5.1: SDP entry for GW service 120
Table 7.1: LMP procedures 122
Table 8.1: LC features 124
Table 9.1: Modes 126
Table 9.2: Security aspects 126
Table 9.3: Idle mode procedures 127
Table 11.1: Defined timers 135

Part K:4

INTERCOM PROFILE

This profile defines the requirements for Bluetooth devices necessary for the support of the intercom functionality within the 3-in-1 phone use case. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the 3-in-1 phone use case.

CONTENTS

1 Introduction 143
1.1 Scope 143
1.2 Profile Dependencies 143
1.3 Symbols and conventions 144
1.3.1 Requirement status symbols 144
1.3.2 Signalling diagram conventions 144
2 Profile Overview 145
2.1 Profile stack 145
2.2 Configuration and roles 146
2.3 User requirements and scenarios 146
2.4 Profile fundamentals 147
2.5 Feature definitions 147
2.6 Conformance 147
3 Application layer 148
4 TCS Binary 149
4.1 Call Control procedures 149
4.1.1 Call request 149
4.1.2 Call confirmation 149
4.1.3 Call connection 149
4.1.4 Failure of call establishment 149
4.1.5 Call clearing 150
4.1.6 Call information 150
4.2 TCS Binary Message overview 150
4.3 Information Element overview 151
4.3.1 Bearer capability 152
4.3.2 Call class 152
4.3.3 Cause 152
4.4 Link loss 152
5 SDP Interoperability Requirements 153
6 L2CAP Interoperability Requirements 154
6.1 Channel types 154
6.2 Configuration options 154
6.2.1 Maximum Transmission unit 154
6.2.2 Flush timeout option 154
6.2.3 Quality of Service 154

7 Link Manager (LM) Interoperability Requirements 155
7.1 Capability overview 155
8 Link Control (LC) Interoperability Requirements 156
8.1 Capability overview 156
8.2 Class of Device 157
9 Generic Access Profile 158
9.1 Modes 158
9.2 Security aspects 158
9.3 Idle mode procedures 158
10 Annex A (Informative): Signalling flows 159
10.1 Call establishment 159
10.2 Call Clearing 160
11 Timers and counters 161
12 List of Figures 162
13 List of Tables 163

1 INTRODUCTION

1.1 SCOPE

This Intercom profile defines the protocols and procedures that shall be used by devices implementing the intercom part of the usage model called ‘3-in-1 phone’. More popularly, this is often referred to as the ‘walkie-talkie’ usage of Bluetooth.

1.2 PROFILE DEPENDENCIES

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. As indicated in the figure, the Intercom profile is dependent only upon the Generic Access Profile – details are provided in Section 9.

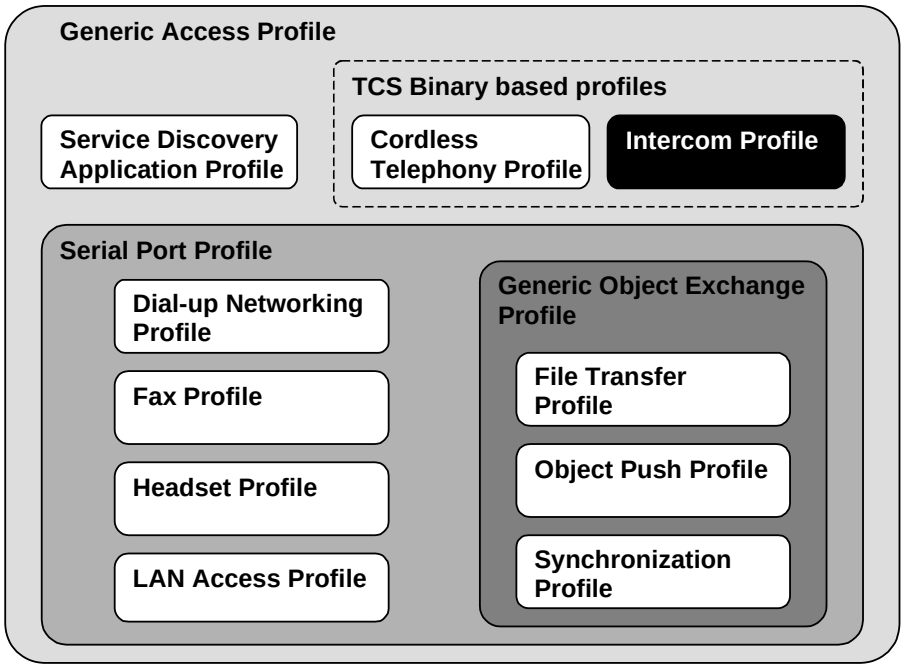


Figure 1.1: Bluetooth Profiles

1.3 SYMBOLS AND CONVENTIONS

1.3.1 Requirement status symbols

In this document, the following symbols are used:

- ‘M’ for mandatory to support
- ‘O’ for optional to support
- ‘X’ for excluded (used for capabilities that may be supported by the unit but shall never be used in the use case)
- ‘C’ for conditional to support
- ‘N/A’ for not applicable (in the given context it is impossible to use this capability)

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

1.3.2 Signalling diagram conventions

The following arrows are used in diagrams describing procedures:

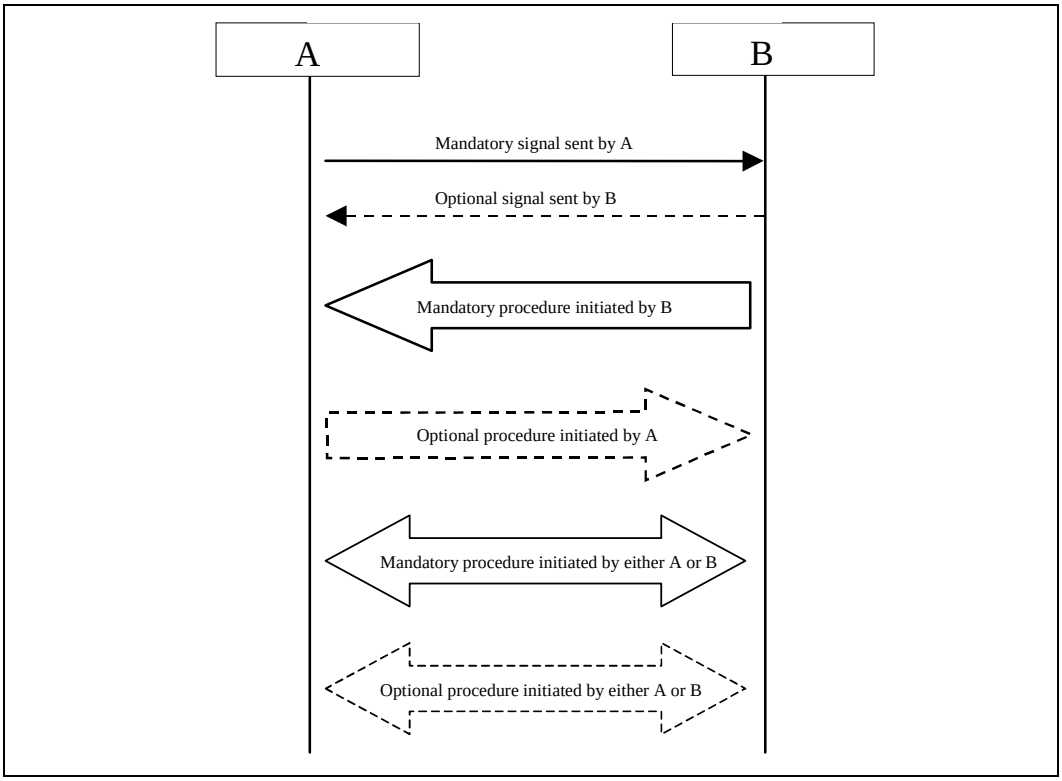


Figure 1.2: Arrows used in signalling diagrams

2 PROFILE OVERVIEW

2.1 PROFILE STACK

Figure 2.1 below shows the protocols as used within this profile:

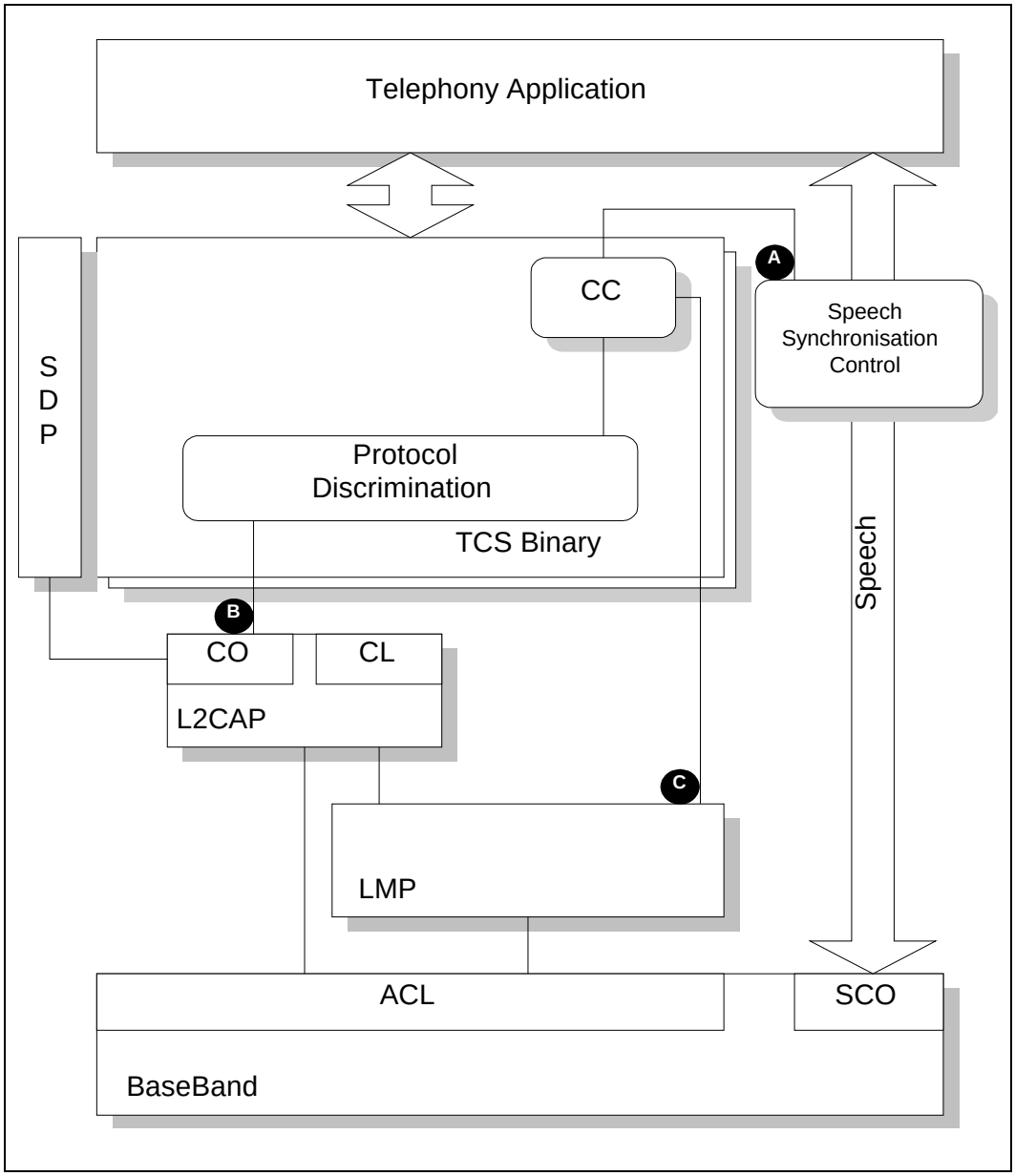


Figure 2.1: Intercom Profile Stack

This profile will define the requirements for each of the layers in the model above.

In the profile, the interfaces in Figure 2.1 above are used for the following purposes:

A) The Call Control entity uses this interface to the speech synchronization control to connect and disconnect the internal speech paths;

B) Used to deliver TCS messages on the connection-oriented (point-to-point) L2CAP channel;

C) Used by the Call Control entity to control the Link Manager directly for the purpose of establishing and releasing SCO links;

Note that, for initialization purposes, it is additionally required to control the LC/Baseband directly, to enable inquiry, paging, inquiry scan, page scan.

2.2 CONFIGURATION AND ROLES

The figure below shows a typical configuration of devices for which the Intercom profile is applicable:

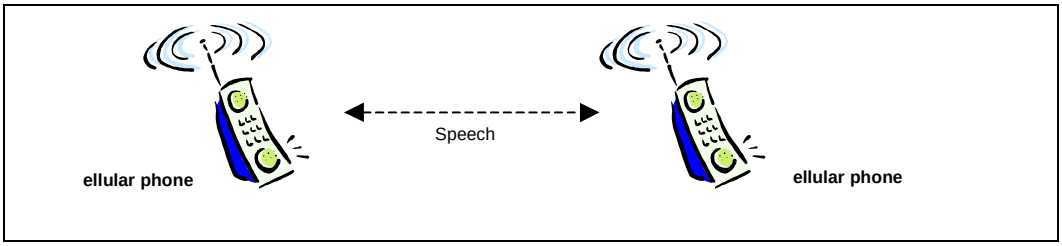


Figure 2.2: Intercom profile, example

As the intercom usage is completely symmetrical, there are no specific roles defined. A device supporting the Intercom profile will generally be denoted as Terminal (TL).

2.3 USER REQUIREMENTS AND SCENARIOS

The Intercom profile defines the protocols and procedures that shall be used by devices implementing the intercom part of the use case called ‘3-in-1 phone’.

The scenarios targeted by this use case are typically those where a direct speech link is required between two devices (phone, computer, …), established using telephony-based signalling.

A typical scenario is the following:

- Two (cellular) phone users engaged in a speech call, on a direct phone-to-phone connection using Bluetooth only.

2.4 PROFILE FUNDAMENTALS

Here is a brief summary of the interactions that take place when a terminal wants to establish an intercom call towards another terminal. In the description below, the term initiator (A-party) and acceptor (B-party) will be used to designate the direction of the call.

1. If the initiator of the intercom call does not have the Bluetooth Address of the acceptor, it has to obtain this; e.g. using the Device discovery procedure – see Section 6.4 of Generic Access profile.
2. The profile does not mandate a particular security mode. If users of either device (initiator/acceptor) want to enforce security in the execution of this profile, the authentication procedure (see Section 5.1 of Generic Access profile) has to be performed to create a secure connection.
3. The initiator establishes the link and channel as indicated in Section 7 of the Generic Access profile. Based on the security requirements enforced by users of either device, authentication may be performed and encryption may be enabled.
4. The intercom call is established.
5. After the intercom call has been cleared, the channel and link will be released as well.

2.5 FEATURE DEFINITIONS

Call information **–** The ability to provide additional information during the active phase of a call.

Intercom call – A speech call between two terminals.

On hook **–** The ability to indicate the action of going on-hook (e.g. to terminate a call) and release of all radio resources related to that call.

2.6 CONFORMANCE

When conformance to this profile is claimed, all capabilities indicated 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 certification program.

3 APPLICATION LAYER

The following text together with the associated sub-clauses defines the feature requirements with regard to this profile.

Table 3.1 below shows the feature requirements made by this profile.

Item no.	Feature	Support
1.	Intercom call	M
2.	On hook	M
3.	Call information	O

Table 3.1: Application layer features

Table 3.2 below maps each feature to the TCS Binary procedures used for that feature and shows whether the procedure is optional, mandatory or conditional for that feature.

Item no.	Feature	Procedure	Ref.	Support
1.	Intercom call	Call request	4.1.1	M
		Call confirmation	4.1.2	M
		Call connection	4.1.3	M
2.	On hook	Call clearing	4.1.5	M
3.	Call information	Call information	4.1.6	M

Table 3.2: Application layer feature to procedure mapping

4 TCS BINARY

The following text together with the associated sub-clauses defines the mandatory requirements with regard to this profile.

When describing TCS Binary procedures, this chapter provides additional information concerning lower layer handling. The normative reference for TCS Binary procedures is TCS Binary.

Annex A contains signalling flows that illustrate the procedures in this chapter.

4.1 CALL CONTROL PROCEDURES

4.1.1 Call request

This procedure shall be performed as defined in Section 2.2.1 of TCS Binary. In addition, the following applies: before a call request can be made, a connection-oriented L2CAP channel needs to be established between the two devices, using the procedures as indicated in Section 6. When the L2CAP channel has been established, the terminating side will start timer T_{ic}(100). When, at expiry of timer T_{ic}(100), the terminating side has not received the SETUP message initiating the call request, it may terminate the L2CAP channel. Receiving the SETUP message before expiry of T_{IC}(100) will cancel the timer.

4.1.2 Call confirmation

This procedure shall be performed as defined in Section 2.2.5 of TCS Binary.

4.1.3 Call connection

This procedure shall be performed as defined in Section 2.2.6 of TCS Binary. The following text defines the mandatory requirements with regard to this profile.

The SCO link establishment sub-procedure (see LMP, Section 3.21) shall be initiated before sending a CONNECT.

The speech path shall be connected by a unit when it receives a CONNECT or CONNECT ACKNOWLEDGE.

4.1.4 Failure of call establishment

This procedure shall be performed as defined in Section 2.2.10 of TCS Binary. Additionally, the text in Section 4.1.5 defines the mandatory requirements with regard to this profile concerning call clearing.

4.1.5 Call clearing

All call-clearing and call-collision procedures as defined in Section 2.3 of TCS Binary shall be supported by the TL.

In addition, the following applies: after the last call-clearing message has been sent, a unit shall:

- release the SCO link by invoking the appropriate LMP sub-procedure (see LMP, Section 3.21.5), if not already released.
- terminate the L2CAP channel used for TCS Call-control signalling (if not already terminated) and detach the other unit.

4.1.6 Call information

This procedure shall be performed as defined in Section 2.2.7 of TCS Binary.

4.2 TCS BINARY MESSAGE OVERVIEW

This section defines the allowed TCS Binary messages in the Intercom profile. Messages not mentioned are not applicable.

Message	Support
Alerting	M
Connect	M
Connect Acknowledge	M
Disconnect	M
Information	O
Release	M
Release Complete	M
Setup	M

Table 4.1: TCS Binary messages

4.3 INFORMATION ELEMENT OVERVIEW

This section together with the associated sub-clauses defines the allowed information elements used in TCS Binary messages in the Cordless Telephony profile.

Information element	Support
Message type	M
Audio control	O
Bearer capability	M
Call class	M
Called party number	O
Calling party number	O
Cause	M
Clock offset	N/A
Company-specific	O
Configuration data	N/A
Destination CID	N/A
Keypad facility	O
Progress indicator	N/A
SCO handle	M
Sending complete	O
Signal	O

Table 4.2: TCS Binary information elements

The following subsections define restrictions that apply to the contents of the TCS Binary information elements in the Intercom profile. Note that, in the tables, only fields where restrictions apply are shown. If a field is not shown in a table, it means that all values defined in Section 7 of TCS Binary for that field are allowed.

For those information elements not listed below, no restrictions apply.

4.3.1 Bearer capability

The following restrictions apply to the contents of the Bearer capability information element:

Field	Values allowed
Link type	SCO, None
User information layer 1	CVSD

Table 4.3: Restrictions to contents of Bearer capability information element

4.3.2 Call class

The following restrictions apply to the contents of the Call class information element:

Field	Values allowed
Call class	Intercom call

Table 4.4: Restrictions to contents of Call class information element

4.3.3 Cause

The following restrictions apply to the contents of the Cause information element:

Field	Values allowed
Cause value	#16 – “Normal call clearing” #17 – “User busy”, #18 – “No user responding”, #19 – “No answer from user (user alerted)”, #21 – “Call rejected by user” #34 – “No circuit/channel available”, #41 – “Temporary failure”, #44 – “Requested circuit/channel not available”, #58 – “Bearer capability not presently available”, #65 – “Bearer capability not implemented”, #69 – “Requested facility not implemented”, #102 – “Recovery on timer expiry”

Table 4.5: Restrictions to contents of Cause information element

4.4 LINK LOSS

If a unit in a CC state other than *Null* detects loss of link, it shall immediately go to the *Null* state. Call clearing procedures shall in this case not be performed.

5 SDP INTEROPERABILITY REQUIREMENTS

Table 5.1 lists all intercom-related entries in the SDP database. For each field, the Status column indicates whether the presence of this field is mandatory or optional.

The codes assigned to the mnemonic's used in the Value column as well as the codes assigned to the attribute identifiers (if not specifically mentioned in the AttrID column) can be found in the Bluetooth Assigned Numbers section.

Item	Definition	Type	Value	AttrID	Status	Default
ServiceClassIDList					M	
ServiceClass0		UUID	Generic Telephony		M	
ServiceClass1		UUID	Intercom		M	
Protocol Descriptor List					M	
Protocol0		UUID	L2CAP		M	
Protocol1		UUID	TCS-BIN		M	
BluetoothProfileDe-scriptorList					O	
Profile0	Sup-ported Profiles	UUID	Intercom		M	Intercom
Param0	Profile Version	Uint16	0x0100*		M	0x0100
Service Name	Display-able Text name	String	Service-provider defined		O	"Intercom"

Table 5.1: Service Record

*. Indicating version 1.0

6 L2CAP INTEROPERABILITY REQUIREMENTS

The following text together with the associated sub-clauses define the mandatory requirements with regard to this profile.

6.1 CHANNEL TYPES

In this profile, only connection-oriented channels are used. In the PSM field of the Connection Request packet, the default value for TCS-BIN, 0x0005 (see Section 3.2 of Assigned Numbers) shall be used.

6.2 CONFIGURATION OPTIONS

This section describes the usage of configuration options.

6.2.1 Maximum Transmission unit

The minimum MTU that a L2CAP implementation used for this profile should support is 3 octets.

6.2.2 Flush timeout option

The flush timeout value used for both the GW and the TL shall be the default value of 0xFFFF.

6.2.3 Quality of Service

Negotiation of Quality of Service is optional.

7 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

7.1 CAPABILITY OVERVIEW

In the table below, all LM capabilities are listed. In the table it is shown what LMP features are mandatory to support with respect to this profile and which are optional.

	Procedure	Support in LMP	Support
1.	Authentication	M	
2.	Pairing	M	
3.	Change link key	M	
4.	Change the current link key	M	
5.	Encryption	O	
6.	Clock offset request	M	
7.	Slot offset information	O	
8.	Timing accuracy information request	O	
9.	LMP version	M	
10.	Supported features	M	
11.	Switch of master slave role	O	
12.	Name request	M	
13.	Detach	M	
14.	Hold mode	O	
15.	Sniff mode	O	
16.	Park mode	O	
17.	Power control	O	
18.	Channel quality driven DM/DH	O	
19.	Quality of service	M	
20.	SCO links	O	M
21.	Control of multi-slot packets	O	
22.	Paging scheme	O	
23.	Link supervision	M	
24.	Connection establishment	M	

Table 7.1: LMP procedures

8 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

8.1 CAPABILITY OVERVIEW

The following table lists all capabilities on the LC level.

	Capabilities	Support
1.	Inquiry	
2.	Inquiry scan	
3.	Paging	
4.	Page scan	
A	Type R0	
B	Type R1	
C	Type R2	
5.	Packet types	
A	ID packet	
B	NULL packet	
C	POLL packet	
D	FHS packet	
E	DM1 packet	
F	DH1 packet	
G	DM3 packet	
H	DH3 packet	
I	DM5 packet	
J	DH5 packet	
K	AUX packet	X
L	HV1 packet	
M	HV2 packet	
N	HV3 packet	
O	DV packet	
6.	Inter-piconet capabilities	

Table 8.1: Baseband/LC capabilities

	Capabilities	Support
7.	Voice codec	
A	A-law	
B	μ-law	
C	CVSD	M

Table 8.1: Baseband/LC capabilities

8.2 CLASS OF DEVICE

The Class of Device field shall be set to the following:

1. Set the ‘Generic Telephony’ bit in the Service Class field
2. Indicate ‘Phone’ as Major Device class

9 GENERIC ACCESS PROFILE

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

9.1 MODES

The table shows the support status for Modes within this profile.

	Procedure	Support
1	Discoverability modes	
	Non-discoverable mode	M
	Limited discoverable mode	O
	General discoverable mode	M
2	Connectability modes	
	Non-connectable mode	N/A
	Connectable mode	M
3	Pairing modes	
	Non-pairable mode	O
	Pairable mode	C3
C3: If the bonding procedure is supported, support for pairable mode is mandatory, otherwise optional		

Table 9.1: Modes

9.2 SECURITY ASPECTS

No changes to the requirements as stated in the Generic Access Profile.

9.3 IDLE MODE PROCEDURES

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

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

Table 9.2: Idle mode procedures

10 ANNEX A (INFORMATIVE): SIGNALLING FLOWS

This annex contains signalling diagrams that are used to clarify the interworking between units. This annex is informative only. The diagrams do not represent all possible signalling flows as defined by this profile.

10.1 CALL ESTABLISHMENT

The figure below shows the allowed signalling flow in the successful case:

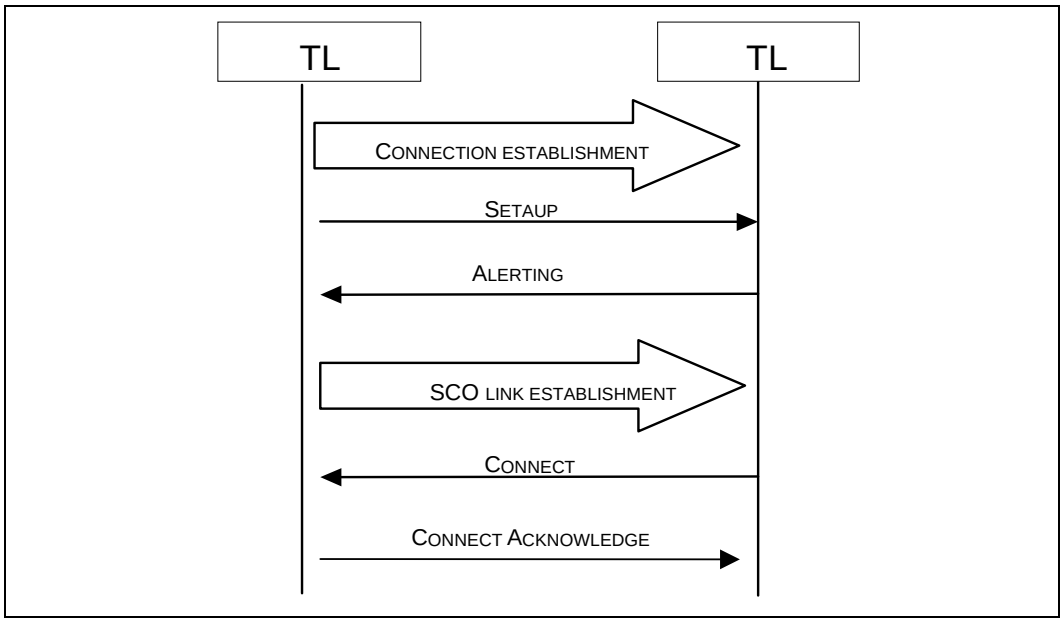


Figure 10.1: Call establishment

10.2 CALL CLEARING

The figure below shows the allowed signalling flow for the call clearing:

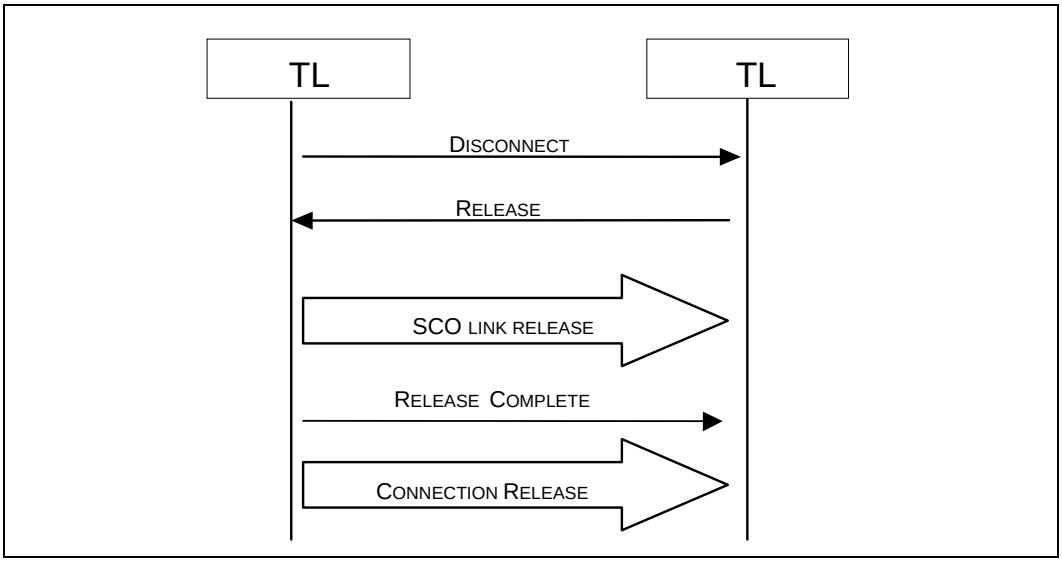


Figure 10.2: Call Clearing signalling flow, successful case

11 TIMERS AND COUNTERS

Timer name	Proposed value	Description	Comment
$T_{IC}(100)$	10s	Time between L2CAP connection establishment and call request initiation	

12 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles 143
Figure 1.2: Arrows used in signalling diagrams 144
Figure 2.1: Intercom Profile Stack 145
Figure 2.2: Intercom profile, example 146
Figure 10.1: Call establishment 159
Figure 10.2: Call Clearing signalling flow, successful case 160

13 LIST OF TABLES

Table 3.1: Application layer features....148
Table 3.2: Application layer feature to procedure mapping....148
Table 4.1: TCS Binary messages 150
Table 4.2: TCS Binary information elements 151
Table 4.3: Restrictions to contents of Bearer capability information element 152
Table 4.4: Restrictions to contents of Call class information element....152
Table 4.5: Restrictions to contents of Cause information element....152
Table 5.1: Service Record....153
Table 7.1: LMP procedures....155
Table 8.1: Baseband/LC capabilities....156
Table 9.1: Modes 158
Table 9.2: Idle mode procedures 158

Part K:5

SERIAL PORT PROFILE

This profile defines the requirements for Bluetooth devices necessary for setting up emulated serial cable connections using RFCOMM between two peer devices. The requirements are expressed in terms of services provided to applications, and by defining the features and procedures that are required for interoperability between Bluetooth devices.

CONTENTS

1 Introduction 170
1.1 Scope 170
1.2 Bluetooth Profile Structure 170
1.3 Symbols and conventions 170
2 Profile overview 171
2.1 Profile stack 171
2.2 Configurations and roles 172
2.3 User requirements and scenarios 172
2.4 Profile fundamentals 173
2.5 Conformance 173
3 Application layer 174
3.1 Procedure overview 174
3.1.1 Establish link and set up virtual serial connection 174
3.1.2 Accept link and establish virtual serial connection 175
3.1.3 Register Service record in local SDP database 175
3.2 Power mode and link loss handling 175
4 RFCOMM Interoperability Requirements 177
4.1 RS232 control signals 178
4.2 Remote Line Status indication 178
4.3 Remote Port Negotiation 178
5 L2CAP Interoperability Requirements 179
5.1 Channel types 179
5.2 Signalling 179
5.3 Configuration options 180
5.3.1 Maximum Transmission unit 180
5.3.2 Flush Timeout 180
5.3.3 Quality of Service 180
6 SDP Interoperability Requirements 181
6.1 SDP Service Records for Serial Port Profile 181
6.2 SDP Procedures 182
7 Link Manager (LM) Interoperability Requirements 183
7.1 Capability overview 183
7.2 Error behavior 183
7.3 Link policy 183

8 Link Control (LC) Interoperability Requirements 184

8.1 Capability overview 184

8.2 Inquiry 185

8.3 Inquiry scan 185

8.4 Paging 185

8.5 Error behavior 185

9 References 186

10 List of Figures 187

11 List of Tables 188

FOREWORD

Interoperability between devices from different manufacturers is provided for a specific service and use case, 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 use case(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 Serial Port Profile defines the protocols and procedures that shall be used by devices using Bluetooth for RS232 (or similar) serial cable emulation.

The scenario covered by this profile deals with legacy applications using Bluetooth as a cable replacement, through a virtual serial port abstraction (which in itself is operating system-dependent).

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.

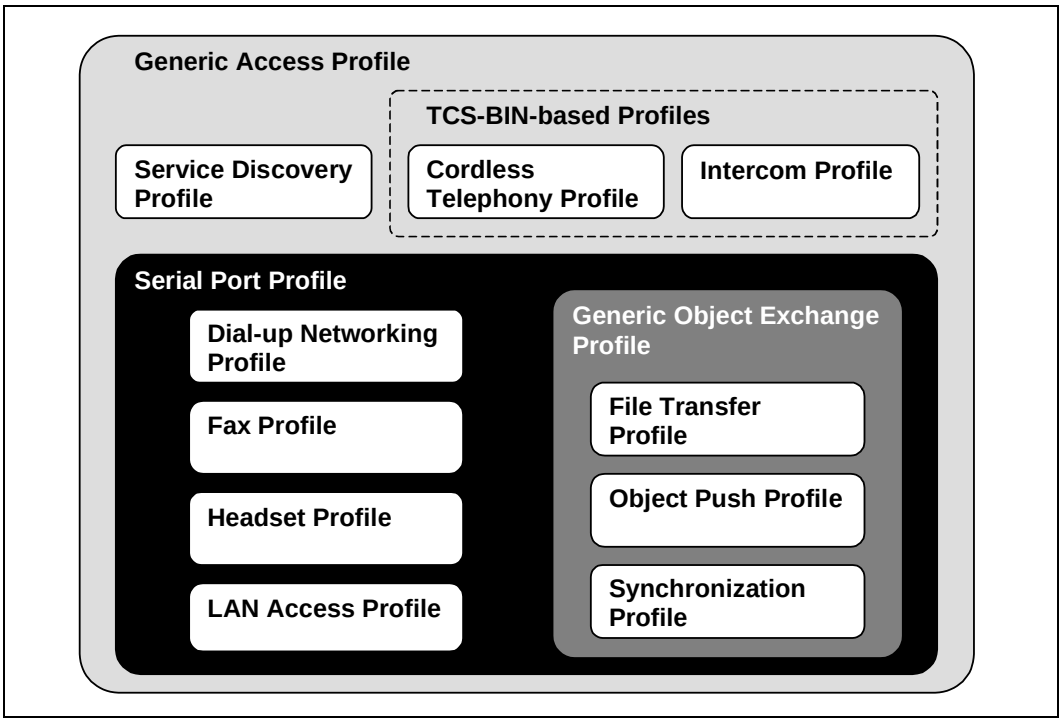


Figure 1.1: Bluetooth Profiles

1.3 SYMBOLS AND CONVENTIONS

This profile uses the symbols and conventions specified in Section 1.2 of the Generic Access Profile [9].

2 PROFILE OVERVIEW

2.1 PROFILE STACK

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

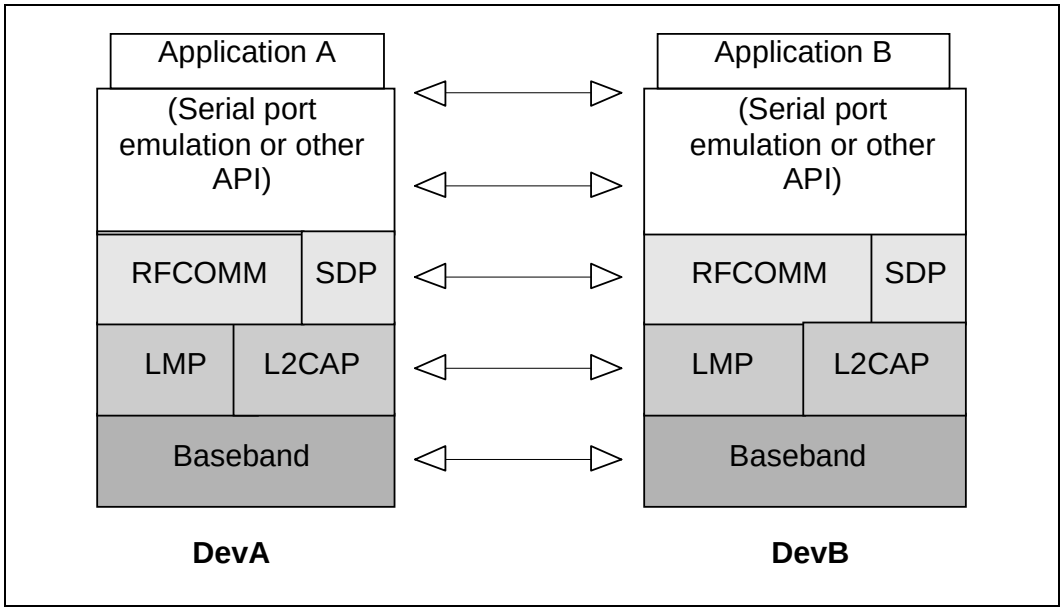


Figure 2.1: Protocol model

The Baseband [1], LMP [2] and L2CAP [3] are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM [4] is the Bluetooth adaptation of GSM TS 07.10 [5], providing a transport protocol for serial port emulation. SDP is the Bluetooth Service Discovery Protocol [6].

The port emulation layer shown in Figure 2.1 is the entity emulating the serial port, or providing an API to applications.

The applications on both sides are typically legacy applications, able and wanting to communicate over a serial cable (which in this case is emulated). But legacy applications cannot know about Bluetooth procedures for setting up emulated serial cables, which is why they need help from some sort of Bluetooth-aware helper application on both sides. (These issues are not explicitly addressed in this profile; the major concern here is for Bluetooth interoperability.)

2.2 CONFIGURATIONS AND ROLES

The figure below shows one possible configuration of devices for this profile:

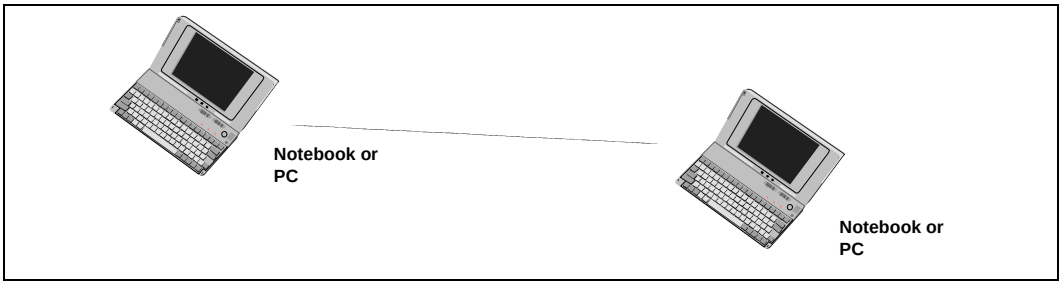


Figure 2.2: Serial Port profile, example with two notebooks.

The following roles are defined for this profile:

Device A (DevA) – This is the device that takes initiative to form a connection to another device (DevA is the *Initiator* according to Section 2.2 in GAP [9]).

Device B (DevB) – This is the device that waits for another device to take initiative to connect. (DevB is the *Acceptor* according to Section 2.2 in GAP [9]).

Note that the order of connection (from DevA to DevB) does not necessarily have to have anything to do with the order in which the legacy applications are started on each side respectively.

Informational note: For purposes of mapping the Serial Port profile to the conventional serial port architecture, both DevA and DevB can be either a Data Circuit Endpoint (DCE) or a Data Terminal Endpoint (DTE). (The RFCOMM protocol is designed to be independent of DTE-DCE or DTE-DTE relationships.)

2.3 USER REQUIREMENTS AND SCENARIOS

The scenario covered by this profile is the following:

- Setting up virtual serial ports (or equivalent) on two devices (e.g. PCs) and connecting these with Bluetooth, to emulate a serial cable between the two devices. Any legacy application may be run on either device, using the virtual serial port as if there were a real serial cable connecting the two devices (with RS232 control signalling).

This profile requires support for one-slot packets only. This means that this profile ensures that data rates up to 128 kbps can be used. Support for higher rates is optional.

Only one connection at a time is dealt with in this profile. Consequently, only point-to-point configurations are considered. However, this should not be construed as imposing any limitation on concurrence; multiple executions of this profile should be able to run concurrently in the same device. This also includes taking on the two different roles (as DevA and DevB) concurrently.

2.4 PROFILE FUNDAMENTALS

For the execution of this profile, use of security features such as authorization, authentication and encryption is optional. Support for authentication and encryption is mandatory, such that the device can take part in the corresponding procedures if requested from a peer device. If use of security features is desired, the two devices are paired during the connection establishment phase (if not earlier), see GAP, Section 7.

Bonding is not explicitly used in this profile, and thus, support for bonding is optional.

Link establishment is initiated by DevA. Service discovery procedures have to be performed to set up an emulated serial cable connection.

There are no fixed master slave roles.

RFCOMM is used to transport the user data, modem control signals and configuration commands.

2.5 CONFORMANCE

When conformance to this profile is claimed, all capabilities indicated 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 certification program.

3 APPLICATION LAYER

This section describes the feature requirements on units complying with the Serial Port profile.

This profile is built upon the Generic Access Profile [9].

- When reading [9], the A-party (the connection initiator) is equivalent to DevA and the B-party is equivalent to the DevB.
- All the mandatory requirements defined in [9] are mandatory for this profile.
- Unless otherwise stated below, all the optional requirements defined in [9] are optional for this profile.

3.1 PROCEDURE OVERVIEW

Table 3.1 shows the required procedures:

	Procedure	Support in DevA	Support in DevB
1.	Establish link and set up virtual serial connection.	M	X
2.	Accept link and establish virtual serial connection.	X	M
3.	Register Service record for application in local SDP database.	X	M

Table 3.1: Application layer procedures

3.1.1 Establish link and set up virtual serial connection

This procedure refers to performing the steps necessary to establish a connection to an emulated serial port (or equivalent) in a remote device. The steps in this procedure are:

1. Submit a query using SDP to find out the RFCOMM Server channel number of the desired application in the remote device.
   This might include a browsing capability to let the user select among available ports (or services) in the peer device. Or, if it is known exactly which service to contact, it is sufficient look up the necessary parameters using the Service Class ID associated with the desired service.
2. Optionally, require authentication of the remote device to be performed. Also optionally, require encryption to be turned on.
3. Request a new L2CAP channel to the remote RFCOMM entity.
4. Initiate an RFCOMM session on the L2CAP channel.

5. Start a new data link connection on the RFCOMM session, using the aforementioned server channel number.

After step 5, the virtual serial cable connection is ready to be used for communication between applications on both sides.

Note: If there already exists an RFCOMM session between the devices when setting up a new data link connection, the new connection must be established on the existing RFCOMM session. (This is equivalent to skipping over steps 3 and 4 above.)

Note: The order between steps 1 and 2 is not critical (may be the other way round).

3.1.2 Accept link and establish virtual serial connection

This procedure refers to taking part in the following steps:

1. If requested by the remote device, take part in authentication procedure and, upon further request, turn on encryption.
2. Accept a new channel establishment indication from L2CAP.
3. Accept an RFCOMM session establishment on that channel.
4. Accept a new data link connection on the RFCOMM session. This may trigger a local request to authenticate the remote device and turn on encryption, if the user has required that for the emulated serial port being connected to (and authentication/encryption procedures have not already been carried out).

Note: steps 1 and 4 may be experienced as isolated events when there already exists an RFCOMM session to the remote device.

3.1.3 Register Service record in local SDP database

This procedure refers to registration of a service record for an emulated serial port (or equivalent) in the SDP database. This implies the existence of a Service Database, and the ability to respond to SDP queries.

All services/applications reachable through RFCOMM need to provide an SDP service record that includes the parameters necessary to reach the corresponding service/application, see Section 6.1. In order to support legacy applications running on virtual serial ports, the service registration must be done by some helper-application, which is aiding the user in setting up the port.

3.2 POWER MODE AND LINK LOSS HANDLING

Since the power requirements may be quite different for units active in the Serial Port profile, it is not required to use any of the power-saving modes. However, requests to use a low-power mode shall, if possible, not be denied.

If sniff, park, or hold mode is used, neither RFCOMM DLCs nor the L2CAP channel are released.

If a unit detects the loss of link, RFCOMM shall be considered having been shut down. The disconnect DLC and shutdown RFCOMM procedures referenced in Section 4 shall not be performed. Before communication on higher layers can be resumed, the Initialize RFCOMM session procedure has to be performed.

4 RFCOMM INTEROPERABILITY REQUIREMENTS

This section describes the requirements on RFCOMM in units complying with the Serial Port profile.

	Procedure	**Ability to initiate**		**Ability to respond**	
		DevA	**DevB**	**DevA**	**DevB**
1.	Initialize RFCOMM session	M1	X1	X1	M1
2.	Shutdown RFCOMM session	M	M	M	M
3.	Establish DLC	M	X1	X1	M
4.	Disconnect DLC	M	M	M	M
5.	RS232 control signals	C1	C1	M	M
6.	Transfer information	M	M	N/A1	N/A1
7.	Test command	X	X	M	M
8.	Aggregate flow control	C1	C1	M	M
9.	Remote Line Status indication	O	O	M	M
10.	DLC parameter negotiation	O	O	M	M
11.	Remote port negotiation	O	O	M	M

Table 4.1: RFCOMM capabilities

M1: The ability to have more than one RFCOMM session operational concurrently is optional in the RFCOMM protocol. Although support of concurrence is encouraged where it makes sense, this profile does not mandate support of concurrent RFCOMM sessions in either DevA or DevB.

X1: Within the execution of the roles defined in this profile, these abilities will not be used.

N/A1: Information transfer is unacknowledged in the RFCOMM protocol.

C1: Which flow control mechanism to use (per-DLC, aggregate, or both) is an implementation issue. But, if an implementation cannot guarantee that there will always be buffers available for data received, the ability to send either per-DLC flow control or aggregate flow control is mandatory.

Some of the procedures are further commented in subsections below.

4.1 RS232 CONTROL SIGNALS

According to TS 07.10 [5], section 5.4.6.3.7, all devices are required to send information on all changes in RS232 control signals with the Modem Status Command.

However, since RFCOMM can be used with an adaptation layer implementing any kind of API (not only virtual serial ports), it is optional to use all RS232 control signals except flow control (the RTR signal in TS 07.10 [5]). This signal can be mapped on RTS/CTS or XON/XOFF or other API mechanisms, which is an implementation issue.

Informative note: To provide interoperability between devices actually using all RS232 control signals, and devices not using them, the former type of implementation must set the states of the appropriate signals in APIs/connectors to suitable default values depending on RFCOMM DLC state. The implementation must not rely on receiving any RS232 control information from peer devices. The dependency on RFCOMM DLC state may mean that DSR/DTR as well as DCD are set to high level when an RFCOMM DLC has been established, and that the same signals are set to low level if the corresponding DLC is closed for any reason.

4.2 REMOTE LINE STATUS INDICATION

It is required to inform the other device of any changes in RS232 line status with the Remote Line Status indication command, see [5], section 5.4.6.3.10, if the local device relays information from a physical serial port (or equivalent) where overrun-, parity- or framing-errors may occur.

4.3 REMOTE PORT NEGOTIATION

DevA may inform DevB of RS232 port settings with the Remote Port Negotiation Command, directly before DLC establishment. See [5], section 5.4.6.3.9. There is a requirement to do so if the API to the RFCOMM adaptation layer exposes those settings (e.g. baud rate, parity).

DevB is allowed to send the Remote Port Negotiation command.

Informative note: the information conveyed in the remote port negotiation procedure is expected to be useful only in type II devices (with physical serial port) according to section 1.2 in RFCOMM [4], or if data pacing is done at an emulated serial port interface for any reason. RFCOMM as such will not artificially limit the throughput based on baud rate settings, see RFCOMM [4], chapter 2.

5 L2CAP INTEROPERABILITY REQUIREMENTS

The following text together with the associated sub-clauses defines the mandatory requirements with regard to this profile.

	Procedure	Support in DevA/DevB
1.	Channel types	
	Connection-oriented channel	M
	Connectionless channel	X1
2.	Signalling	
	Connection Establishment	M
	Configuration	M
	Connection Termination	M
	Echo	M
	Command Rejection	M
3.	Configuration Parameter Options	
	Maximum Transmission Unit	M
	Flush Timeout	M
	Quality of Service	O

Table 5.1: L2CAP procedures

X1: Connectionless channel is not used within the execution of this profile, but concurrent use by other profiles/applications is not excluded.

5.1 CHANNEL TYPES

In this profile, only connection-oriented channels shall be used. This implies that broadcasts will not be used in this profile.

In the PSM field of the Connection Request packet, the value for RFCOMM defined in the Assigned Numbers document [8], section 3.2 must be used.

5.2 SIGNALLING

Only DevA may issue an L2CAP Connection Request within the execution of this profile. Other than that, the Serial Port Profile does not impose any additional restrictions or requirements on L2CAP signalling.

5.3 CONFIGURATION OPTIONS

This section describes the usage of configuration options in the Serial Port Profile.

5.3.1 Maximum Transmission unit

This profile does not impose any restrictions on MTU sizes over the restrictions stated in L2CAP [3], section 6.1.

5.3.2 Flush Timeout

Serial Port data is carried over a reliable L2CAP channel. The flush timeout value shall be set to its default value 0xffff.

5.3.3 Quality of Service

Negotiation of Quality of Service is optional in this profile.

Recommendation: Implementations should try to keep an upper limit of 500 milliseconds on the latency incurred when going back from a low power mode to active mode.

6 SDP INTEROPERABILITY REQUIREMENTS

6.1 SDP SERVICE RECORDS FOR SERIAL PORT PROFILE

There are no SDP Service Records related to the Serial Port Profile in DevA.

The following table is a description of the Serial Port related entries in the SDP database of DevB. It is allowed to add further attributes to this service record.

Item	Definition	Type/Size	Value	AttributeID
ServiceClassIDList			Note1	0x0001
ServiceClass0	SerialPort / Note3	UUID/32-bit	Note1	
ProtocolDescriptorList				0x0004
Protocol0	L2CAP	UUID/32-bit	L2CAP /Note1	
Protocol1	RFCOMM	UUID/32-bit	RFCOMM /Note1	
ProtocolSpecificParameter0	Server Channel	Uint8	N = server channel #	
ServiceName	Displayable text name	DataElement/ String	“COM5” / Note4	Note2

Table 6.1: SDP Service Record

Notes:

1. Defined in the Assigned Numbers document [8].
2. For national language support for all “displayable” text string attributes, an offset has to be added to the LanguageBaseAttributeIDList value for the selected language (see the SDP Specification [6], section 5.1.14 for details).
3. The ‘SerialPort’ class of service is the most generic type of service. Addition of other, more specific services classes are not excluded by this profile.
4. The ServiceName attribute value suggested here is merely an example; a helper application setting up a serial port may give the port a more descriptive name.

6.2 SDP PROCEDURES

To retrieve the service records in support of this profile, the SDP client entity in DevA connects and interacts with the SDP server entity in DevB via the SDP and L2CAP procedures presented in sections 5 and 6 of SDAP [7]. In accordance to SDAP, DevA plays the role of the LocDev, while DevB plays the role of the RemDev.

7 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

7.1 CAPABILITY OVERVIEW

In addition to the requirements on supported procedures stated in the Link Manager specification itself (see Section 3 in the Link Manager Protocol), this profile also requires support for Encryption both in DevA and DevB.

7.2 ERROR BEHAVIOR

If a unit tries to use a mandatory feature, and the other unit replies that it is not supported, the initiating unit shall send an LMP_detach with detach reason "unsupported LMP feature."

A unit shall always be able to handle the rejection of the request for an optional feature.

7.3 LINK POLICY

There are no fixed master-slave roles for the execution of this profile.

This profile does not state any requirements on which low-power modes to use, or when to use them. That is up to the Link Manager of each device to decide and request as seen appropriate, within the limitations of the latency requirement stated in Section 5.3.3.

8 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

8.1 CAPABILITY OVERVIEW

The following table lists all capabilities on the LC level, and the extra requirements added to the ones in the baseband specification by this profile.

	Capabilities	Support in DevA	Support in DevB
1.	Inquiry		X1
2.	Inquiry scan	X1	
3.	Paging		X1
4.	Page scan		
A	Type R0	X1	
B	Type R1	X1	
C	Type R2	X1	
5.	Packet types		
A	ID packet		
B	NULL packet		
C	POLL packet		
D	FHS packet		
E	DM1 packet		
F	DH1 packet		
G	DM3 packet		
H	DH3 packet		
I	DM5 packet		
J	DH5 packet		
K	AUX packet	X1	X1
L	HV1 packet		
M	HV2 packet		
N	HV3 packet		
O	DV packet		
6.	Inter-piconet capabilities		

Table 8.1: Baseband/LC capabilities

	Capabilities	Support in DevA	Support in DevB
7.	Voice codec		
A	A-law		
B	μ-law		
C	CVSD		

Table 8.1: Baseband/LC capabilities

X1: These capabilities are not used within the execution of this profile, but concurrent use by other profiles/applications is not excluded.

8.2 INQUIRY

When inquiry is invoked in DevA, it shall use the General Inquiry procedure, see GAP [9], Section 6.1.

Only DevA may inquire within the execution of this profile.

8.3 INQUIRY SCAN

For inquiry scan, (at least) the GIAC shall be used, according to one of the discoverable modes defines in GAP [9], Section 4.1.2. and Section 4.1.3. That is, it is allowed to only use the Limited discoverable mode, if appropriate for the application(s) residing in DevB.

In the DevB INQUIRY RESPONSE messages, the Class of Device field will not contain any hint as to whether DevB is engaged in the execution of the Serial Port Profile or not. (This is due to the fact the generalized Serial Port service for legacy applications delivered by this profile does not fit within any of the major Service Class bits in the Class Of Device field definition.)

8.4 PAGING

Only DevA may page within the execution of this profile. The paging step will be skipped in DevA when execution of this profile begins when there already is a baseband connection between DevA and DevB. (In such a case the connection may have been set up as a result of previous paging by DevB.)

8.5 ERROR BEHAVIOR

Since most features on the LC level have to be activated by LMP procedures, errors will mostly be caught at that layer. However, there are some LC procedures that are independent of the LMP layer, e.g. inquiry or paging. Misuse of such features is difficult or sometimes impossible to detect. There is no mechanism defined to detect or prevent such improper use.

9 REFERENCES

[1] Bluetooth Special Interest Group, Bluetooth baseband specification

[2] Bluetooth Special Interest Group, Link Manager Protocol

[3] Bluetooth Special Interest Group, L2CAP Specification

[4] Bluetooth Special Interest Group, RFCOMM with TS 07.10

[5] ETSI, TS 101 369 (GSM 07.10) version 6.3.0

[6] Bluetooth Special Interest Group, Service Discovery Protocol (SDP)

[7] Bluetooth Special Interest Group, Service Discovery Application Profile

[8] Bluetooth Special Interest Group, Bluetooth Assigned Numbers

[9] Bluetooth Special Interest Group, Generic Access Profile

10 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles 170
Figure 2.1: Protocol model 171
Figure 2.2: Serial Port profile, example with two notebooks. 172

11 LIST OF TABLES

Table 3.1: Application layer procedures 174
Table 4.1: RFCOMM capabilities 177
Table 5.1: L2CAP procedures 179
Table 6.1: SDP Service Record 181
Table 8.1: Baseband/LC capabilities 184

Part K:6

HEADSET PROFILE

This profile defines the requirements for Bluetooth devices necessary to support the Headset use case. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the Headset use case.

CONTENTS

1 Introduction 193
1.1 Scope 193
1.2 Profile Dependencies 193
1.3 Symbols and conventions 194
1.3.1 Requirement status symbols 194
1.4 Signalling diagram conventions 195
2 Profile Overview 196
2.1 Profile stack 196
2.2 Configuration and roles 197
2.3 User requirements and scenarios 198
2.4 Profile fundamentals 198
2.5 Conformance 199
3 Application layer 200
4 Headset Control Interoperability Requirements....201
4.1 Introduction 201
4.2 Incoming audio connection 201
4.3 Outgoing audio connection 202
4.4 Audio connection release 203
4.5 Audio connection transfer 204
4.5.1 Audio connection transfer from AG to HS 204
4.5.2 Audio connection transfer from HS to AG 205
4.6 Remote audio volume control 205
4.7 AT Commands and Result Codes 207
4.7.1 General 207
4.7.2 AT capabilities re-used from V.250 and GSM 07.07....207
4.7.3 Bluetooth-defined AT capabilities 207
4.8 Lower layer handling 208
4.8.1 Connection handling without PARK mode 208
4.8.1.1 Connection establishment 208
4.8.1.2 Connection release 208
4.8.2 Connection handling with PARK mode 208
4.8.2.1 Connection establishment 208
4.8.2.2 Connection release 209

5 Serial Port Profile 210
5.1 RFCOMM Interoperability Requirements 210
5.2 L2CAP Interoperability Requirements 210
5.3 SDP Interoperability Requirements 211
5.4 Link Manager (LM) Interoperability Requirements 212
5.5 Link Control (LC) Interoperability Requirements 213
5.5.1 Class of Device 213
6 Generic Access Profile 214
6.1 Modes 214
6.2 Security aspects 214
6.3 Idle mode procedures 214
7 References 215
8 List of Figures 216
9 List of Tables 217

1 INTRODUCTION

1.1 SCOPE

This Headset profile defines the protocols and procedures that shall be used by devices implementing the usage model called ‘Ultimate Headset’. The most common examples of such devices are headsets, personal computers, and cellular phones.

The headset can be wirelessly connected for the purposes of acting as the device’s audio input and output mechanism, providing full duplex audio. The headset increases the user’s mobility while maintaining call privacy.

1.2 PROFILE DEPENDENCIES

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.

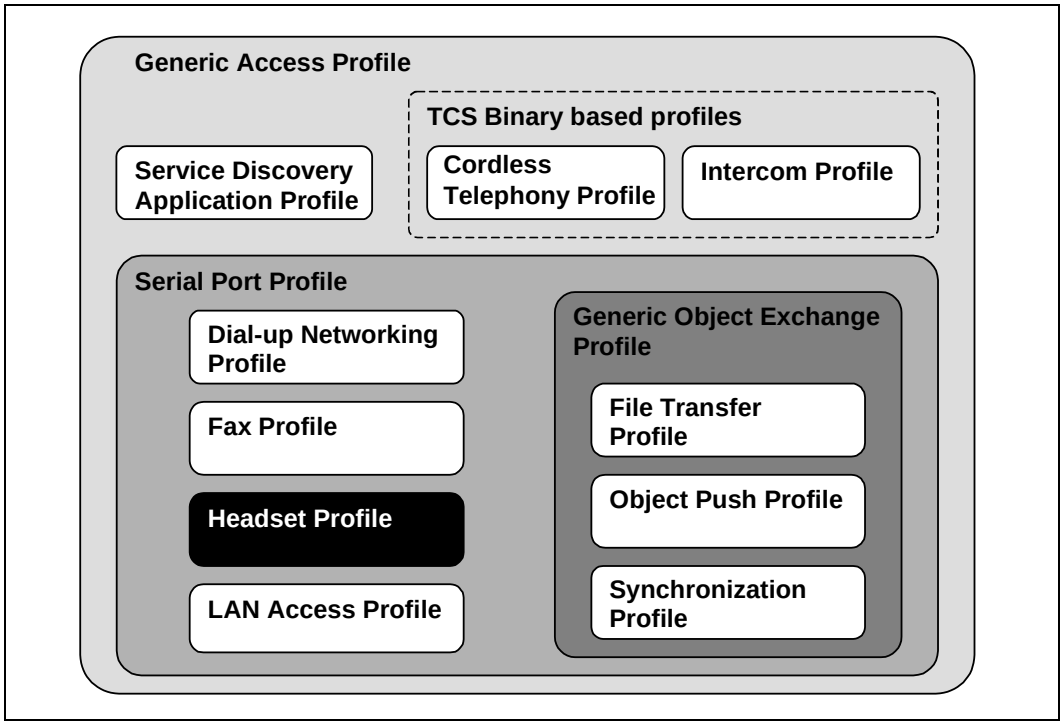


Figure 1.1: Bluetooth Profiles

As indicated in the figure, the Headset profile is dependent upon both the Serial Port Profile and the Generic access profile – details are provided in

Section 5, “Serial Port Profile,” on page 210 and Section 6, “Generic Access Profile,” on page 214.

1.3 SYMBOLS AND CONVENTIONS

1.3.1 Requirement status symbols

In this document, the following symbols are used:

- ‘M’ for mandatory to support
- ‘O’ for optional to support
- ‘X’ for excluded (used for capabilities that may be supported by the unit but shall never be used in this use case)
- ‘C’ for conditional to support
- ‘N/A’ for not applicable (in the given context it is impossible to use this capability)

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

1.4 SIGNALLING DIAGRAM CONVENTIONS

The following arrows are used in diagrams describing procedures:

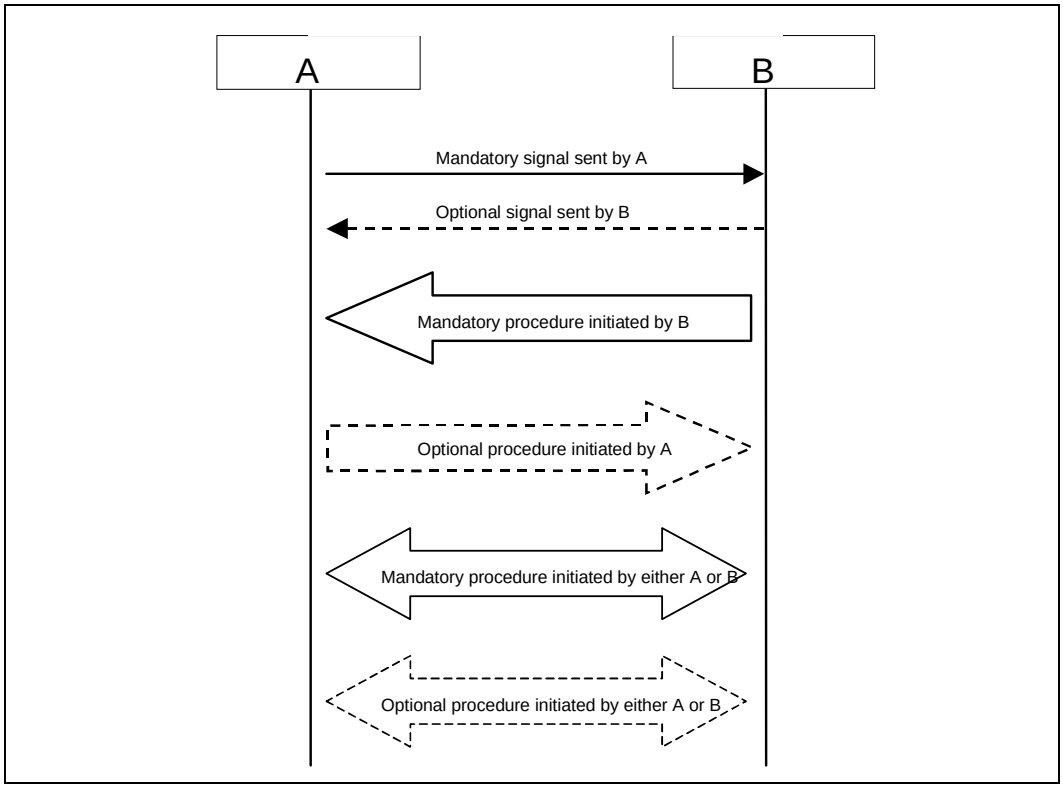


Figure 1.2: Arrows used in signalling diagrams

2 PROFILE OVERVIEW

2.1 PROFILE STACK

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

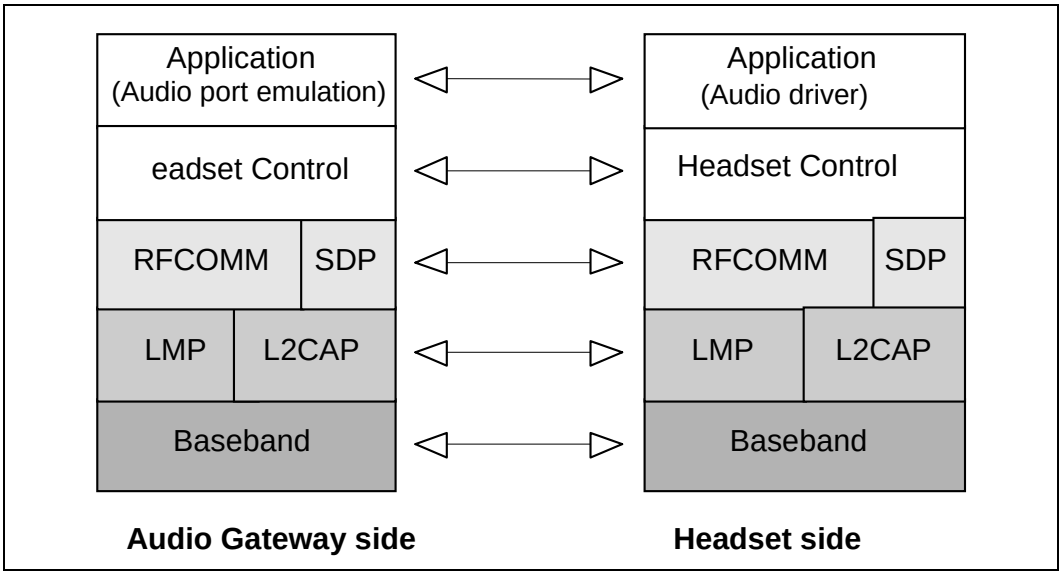


Figure 2.1: Protocol model

The Baseband, LMP and L2CAP are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM is the Bluetooth adaptation of GSM TS 07.10 [5]. SDP is the Bluetooth Service Discovery Protocol. Headset Control is the entity responsible for headset specific control signalling; this signalling is AT command based.

Note: although not shown in the model above, it is assumed by this profile that Headset Control has access to some lower layer procedures (for example SCO link establishment).

The audio port emulation layer shown in Figure 2.1 is the entity emulating the audio port on the cellular phone or PC, and the audio driver is the driver software in the headset.

For the shaded protocols/entities in Figure 2.1, the Serial Port Profile is used as base standard. For these protocols, all requirements stated in the Serial Port Profile apply except in those cases where this profile explicitly states deviations.

2.2 CONFIGURATION AND ROLES

The figures below show two typical configurations of devices for which the Headset profile is applicable:

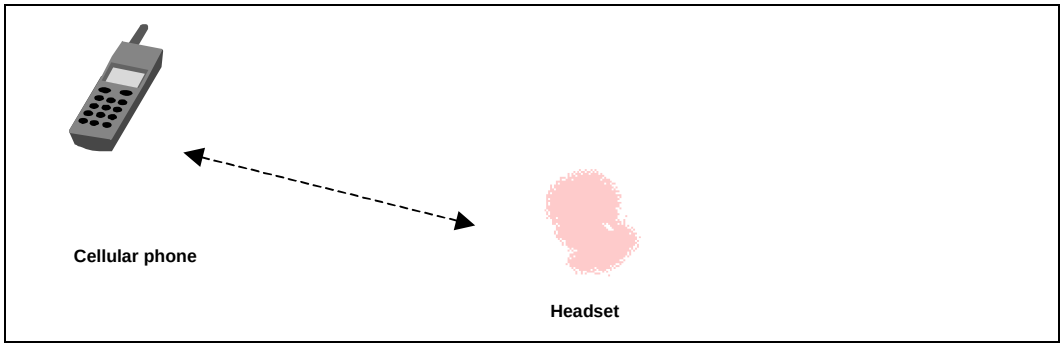


Figure 2.2: Headset profile, example with cellular phone

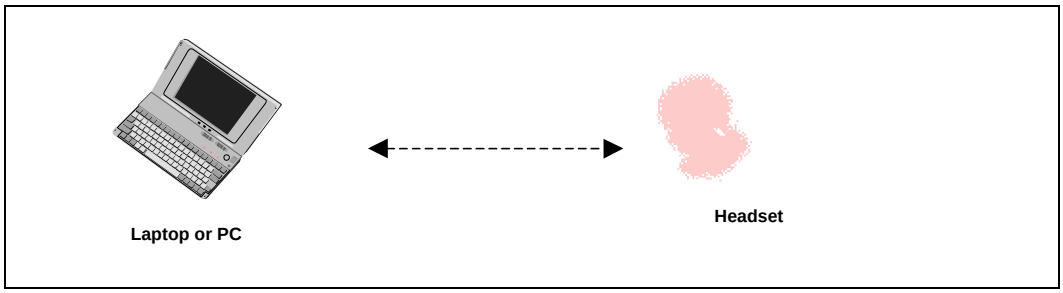


Figure 2.3: Headset profile, example with personal computer

The following roles are defined for this profile:

Audio Gateway **(AG)** – This is the device that is the gateway of the audio, both for input and output. Typical devices acting as Audio Gateways are cellular phones and personal computer.

Headset **(HS)** – This is the device acting as the Audio Gateway's remote audio input and output mechanism.

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

2.3 USER REQUIREMENTS AND SCENARIOS

The Headset profile defines the protocols and procedures that shall be used by devices implementing the use case called ‘Ultimate Headset’.

The following restrictions apply to this profile:

a) For this profile, it is assumed that the ultimate headset use case is the only use case active between the two devices;

b) The profile mandates the usage of CVSD for transmission of audio (for the Bluetooth part). The resulting audio is monophonic, with a quality that – under normal circumstances – will not have perceived audio degradation.

c) Between headset and audio gateway, only one audio connection at a time is supported;

d) The audio gateway controls the SCO link establishment and release. The headset directly connects (disconnects) the internal audio streams upon SCO link establishment (release). Valid speech exists on the SCO link in both directions, once established;

e) The profile offers only basic interoperability – for example, handling of multiple calls at the audio gateway is not included;

f) The only assumption on the headset’s user interface is the possibility to detect a user initiated action (e.g. pressing a button).

2.4 PROFILE FUNDAMENTALS

A headset may be able to use the services of audio gateway without the creation of a secure connection. It is up to the user, if he/she wants to enforce security on devices that support authentication and/or encryption in the execution of this profile. If baseband authentication and/or encryption is desired, the two devices have to create a secure connection, using the GAP authentication procedure as described in Section 5.1 of the Generic Access profile. This procedure may then include entering a PIN code, and will include creation of link keys. In most cases, the headset will be a device with a limited user interface, so the (fixed) pin code of the headset will be used during the GAP authentication procedure.

A link has to be established when a call is initiated or received. Normally, this requires paging of the other device but, optionally, it may require unparking.

There are no fixed master/slave roles.

The audio gateway and headset provide serial port emulation. For the serial port emulation, RFCOMM is used. The serial port emulation is used to transport the user data including modem control signals and AT commands from the headset to the audio gateway. AT commands are parsed by the audio gateway and responses are sent to the headset.

2.5 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 certification program.

3 APPLICATION LAYER

This section describes the feature requirements on units complying with the Headset profile.

Table 3.1 shows the feature requirements made by this profile.

	Feature	Support in HS	Support in AG
1.	Incoming audio connection	M	M
2.	Outgoing audio connection	M	O
3.	Audio connection transfer	M	M
4.	Remote audio volume control	O	O

Table 3.1: Application layer procedures

In the table above, incoming and outgoing shall be interpreted from the headset (HS) point of view.

Table 3.2 maps each feature to the procedures used for that feature. All procedures are mandatory if the feature is supported.

	Feature	Procedure	Ref.
1.	Incoming audio connection	Incoming audio connection establishment	4.2
		Audio connection release	4.4
2.	Outgoing audio connection	Outgoing audio connection establishment	4.3
		Audio connection release	4.4
3.	Audio connection transfer	Audio connection transfer	4.5
4.	Remote audio volume control	Remote audio volume control	4.6

Table 3.2: Application layer feature to procedure mapping

4 HEADSET CONTROL INTEROPERABILITY REQUIREMENTS

4.1 INTRODUCTION

The interoperability requirements for the Headset Control entity are completely contained in this chapter. Section 4.2 until 4.6 specify the requirements for the procedures directly relating to the application layer features.

Section 4.7 specifies the AT commands and results codes used for signalling purposes.

Section 4.8 specifies how the layers beneath the Headset Control entity are used to establish and release a connection.

4.2 INCOMING AUDIO CONNECTION

Upon an internal or user generated event, the AG will initiate connection establishment (see Section 4.8), and once the connection is established, will send an unsolicited result code RING to alert the user. The RING may be repeated for as long as the connection establishment is pending.

Optionally, the AG may provide an in-band ringing tone[1]. In this case, first SCO link establishment takes place.

1. The in-band ringing tone is used to alert the user in the headset earpiece when the user is wearing the headset on his head.

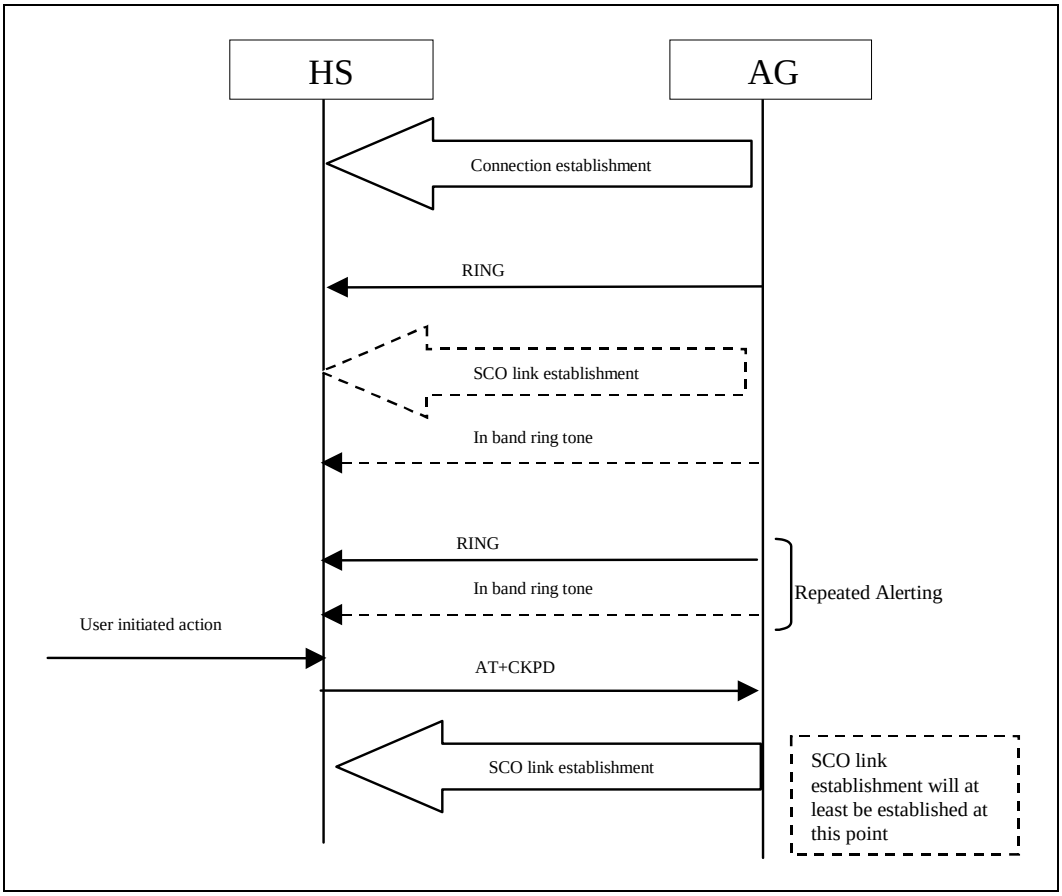


Figure 4.1: Incoming audio connection establishment

The user accepts the incoming audio connection by pressing the button on the headset. By doing this, the HS will send the AT+CKPD command (see Section 4.7) to the AG, whereupon the AG establishes the SCO link (if not already established).

4.3 OUTGOING AUDIO CONNECTION

An outgoing audio connection is initiated on the HS by pushing the button. The HS will initiate connection establishment (see Section 4.8), and will send the AT+CKPD command to the AG. Further internal actions may be needed on the AG to internally establish and/or route an audio stream to the HS[2].

The AG is responsible for establishing the SCO link.

2. For a cellular phone a cellular call may need to be established, e.g. using last dialled number, pre-programmed number. For a personal computer this e.g. relates to playing a wav file, or audio CD.

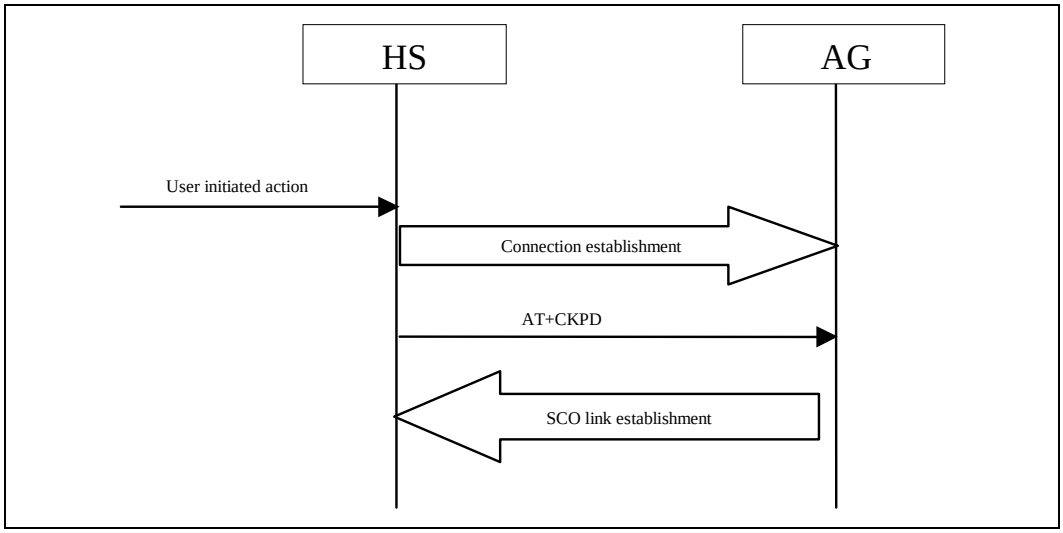


Figure 4.2: Outgoing audio connection establishment

4.4 AUDIO CONNECTION RELEASE

A call can be terminated either on the HS or on the AG. On the HS based upon the button being pushed, on the AG based upon internal actions or user intervention.

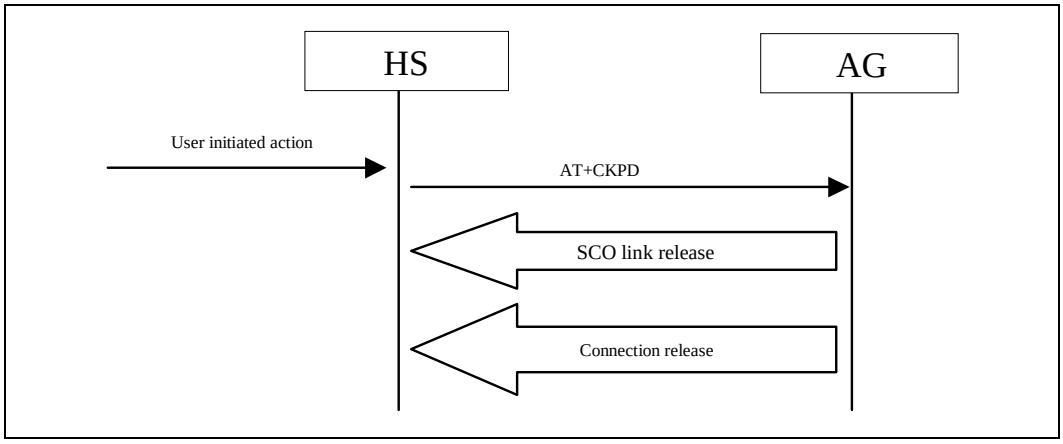


Figure 4.3: Audio connection release – HS initiated

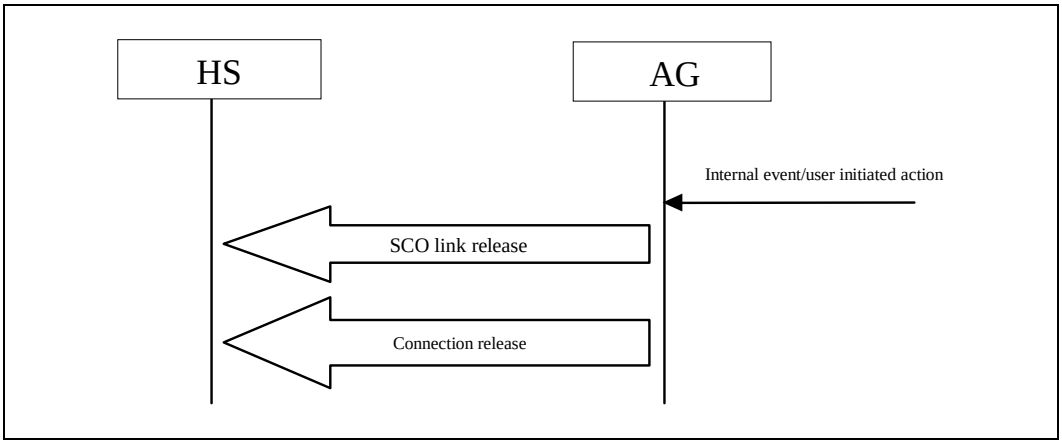


Figure 4.4: Audio connection release – AG initiated

Irrespective of the initiating side, the AG is responsible for releasing the connection (see Section 4.8).

4.5 AUDIO CONNECTION TRANSFER

An audio connection can be transferred from AG to HS or from HS to AG. The connection is transferred to the device initiating the transfer.

4.5.1 Audio connection transfer from AG to HS

The audio connection transfer from AG to HS is initiated by a user action on the HS side, which results in an AT+CKPD command being sent to the AG.

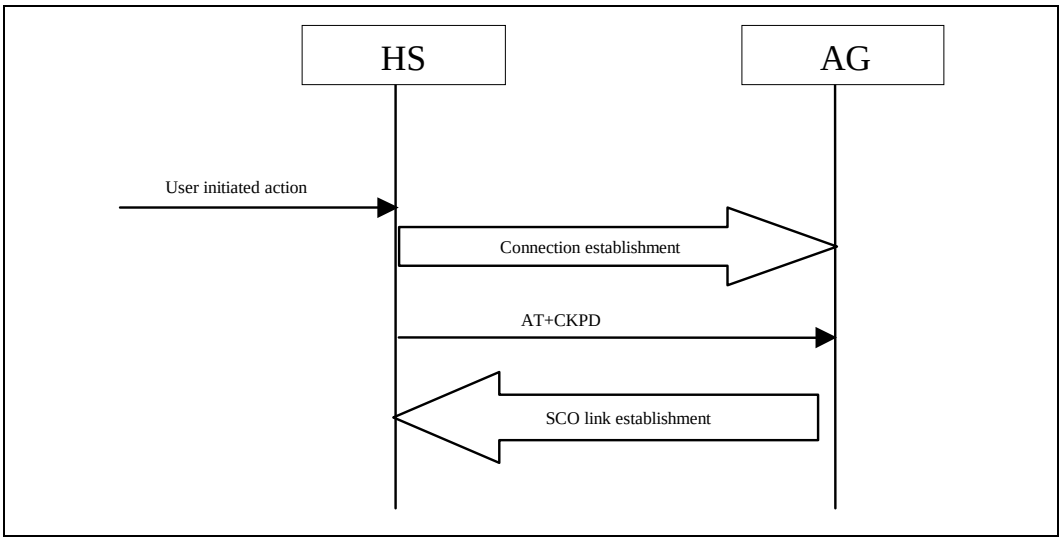


Figure 4.5: Audio connection transfer from AG to HS

4.5.2 Audio connection transfer from HS to AG

The audio connection transfer from HS to AG is initiated by a user action on the AG.

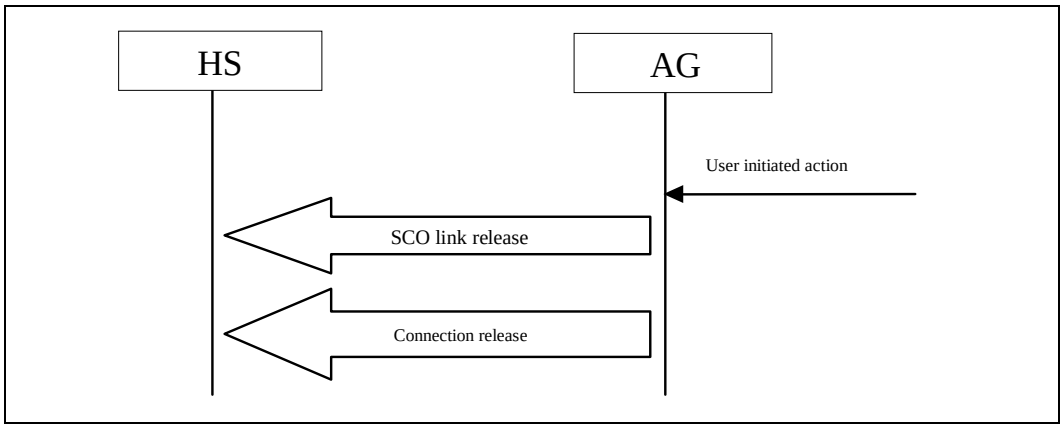


Figure 4.6: Audio connection transfer from HS to AG

4.6 REMOTE AUDIO VOLUME CONTROL

The AG can control the gain of the microphone and speaker of the HS by sending unsolicited result codes +VGM and +VGS respectively. There is no limit to the amount and order of result codes, as long as there is an active audio connection ongoing. When supporting the remote audio volume control, an implementation is not mandated to support both the control of the microphone volume and speaker volume.

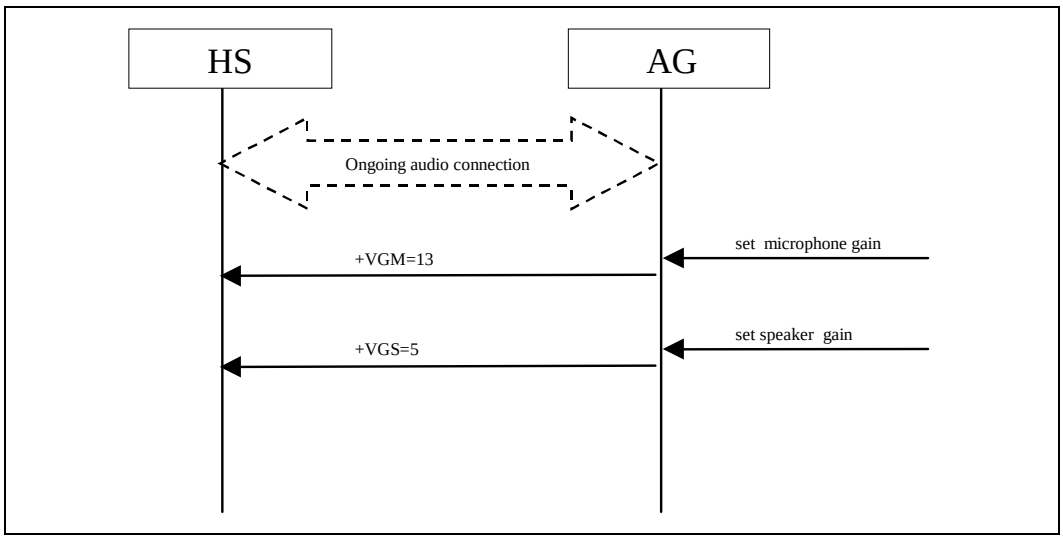


Figure 4.7: Audio volume control – example flow

Both the speaker and microphone gain are represented as parameter to the +VGS and +VGM, on a scale from 0 to 15. The values are absolute values, relating to a particular (implementation-dependent) volume level controlled by the HS.

The HS may store the VGS and VGM settings at connection release, to restore the volume levels at the next connection establishment. At connection establishment, the HS shall inform the AG of the (restored) volume levels using the AT commands +VGS and +VGM. In case local means are implemented on the HS to control the volume levels, the HS shall also use the AT commands +VGS and +VGM to inform the AG of any changes in the volume levels.

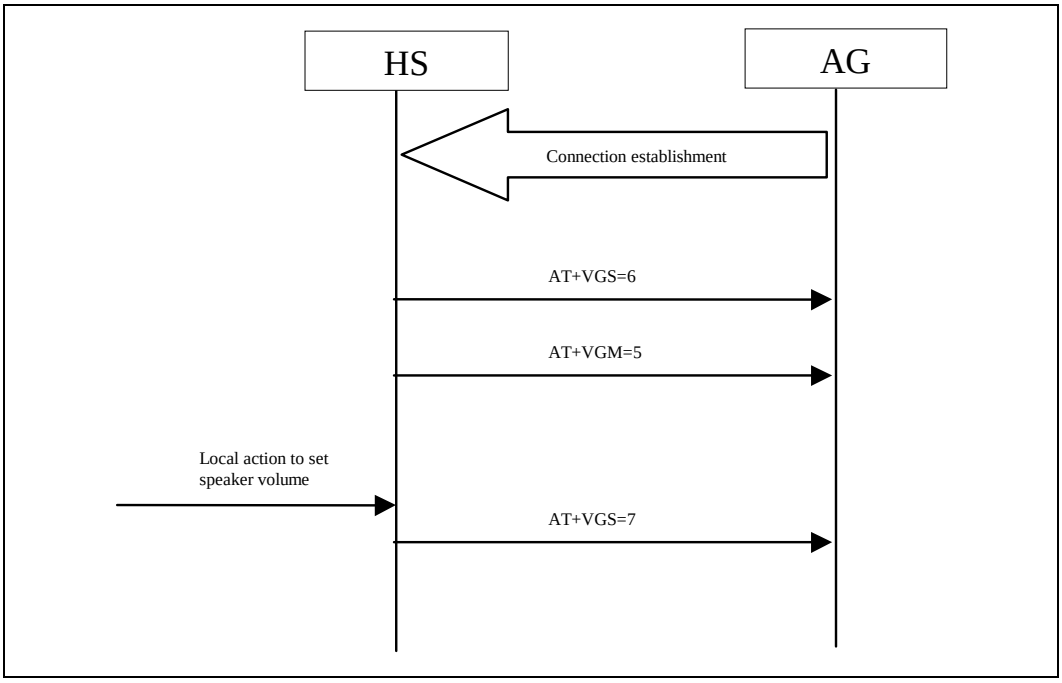


Figure 4.8: Volume level synchronization – example flow

4.7 AT COMMANDS AND RESULT CODES

4.7.1 General

For the exchange of the commands and unsolicited results codes, the format, syntax and procedures of V.250 [1] apply, with the exception that only one command (or unsolicited result code) per command line needs to be expected.

The headset profile uses a subset of AT commands and result codes from existing standards. These are listed in Section 4.7.2. For those AT commands and result codes where no existing commands applied, Section 4.7.3 defines additional ones.

4.7.2 AT capabilities re-used from V.250 and GSM 07.07

The mandatory set of AT commands and unsolicited result codes are indicated in Table 4.1 below.

AT capability	Description
RING	The Incoming call indication of V.250 [1], Section 6.3.4.
+CKPD	The keypad control command of GSM TS 07.07 [2], Section 8.7. For <keys>, the value of 200 indicates the Button on the headset being pushed. The <time> and <pause> parameters have no meaning in the headset profile.

Table 4.1: Mandatory AT capabilities

4.7.3 Bluetooth-defined AT capabilities

Optionally, the AT capabilities as indicated in Table 4.2 may be supported.

AT capability	Syntax	Description	Values
Microphone gain	+VGM=<gain>	Unsolicited result code issued by the AG to set the microphone gain of the HS. <gain> is an unsigned octet, relating to a particular (implementation-dependent) volume level controlled by the HS.	<gain>: 0-15
Speaker gain	+VGS=<gain>	Unsolicited result code issued by the AG to set the speaker gain of the HS. <gain> is an unsigned octet, relating to a particular (implementation-dependent) volume level controlled by the HS.	<gain>: 0-15

Table 4.2: Optional AT capabilities

AT capability	Syntax	Description	Values
Microphone gain level report	+VGM=<gain>	Command issued by the HS to report the current microphone gain level setting to the AG. <gain> is an unsigned octet, relating to a particular (implementation-dependent) volume level controlled by the HS.	<gain>: 0-15
Speaker gain level indica-tion report	+VGS=<gain>	Command issued by the HS to report the current speaker gain level setting to the AG. <gain> is an unsigned octet, relating to a particular (implementation-depen-dent) volume level controlled by the HS.	<gain>: 0-15

Table 4.2: Optional AT capabilities

4.8 LOWER LAYER HANDLING

This section describes how the layers below the Headset Control entity are used to establish and release a connection.Section 4.8.1 describes how connections are handled when the PARK mode is not supported. Section 4.8.2 describes how connections are handled when the PARK mode is supported.

4.8.1 Connection handling without PARK mode

4.8.1.1 Connection establishment

Both the HS and the AG can initiate connection establishment. If there is no RFCOMM session between the AG and the HS, the initiating device shall first initialize RFCOMM. Connection establishment shall be performed as described in Section 7.3 of GAP and Section 3 of SPP.

4.8.1.2 Connection release

When the audio connection is released, the connection may be released as well. The AG always initiates connection release.

4.8.2 Connection handling with PARK mode

4.8.2.1 Connection establishment

If the PARK mode is supported, the connection is established once (e.g. on the first request for an audio connection). Later, when an audio connection is required, the parked device is unparked. In this section, for correct interpretation of the flows given in Section 4.2 to 4.6, the connection establishment is referred to as *initial* connection establishment, whereas the unparking is referred to as connection establishment.

Initial connection establishment shall be performed as described in Section 7.3 of GAP and Section 3 of SPP. Both sides may initiate the initial connection establishment. After initial connection establishment, the park mode is activated.

In Figure 4.9 the behavior is described in case an audio connection needs to be established – the parked device will be unparked. The unpark can be initiated from either side, depending where the request for an audio connection originated. If the PARK mode is used, neither RFCOMM DLCs nor the L2CAP channel is released.

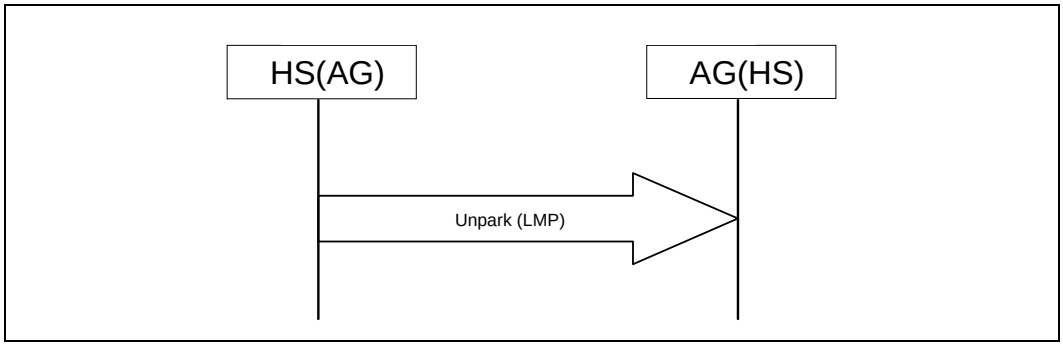


Figure 4.9: Connection establishment – Unparking a parked device

4.8.2.2 Connection release

When the audio connection is released, the connection is parked again, as indicated in Figure 4.10.

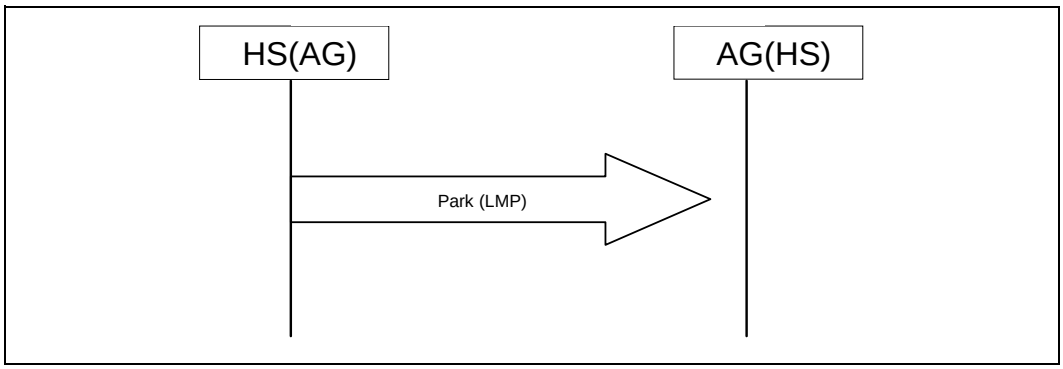


Figure 4.10: Connection release – Parking

When the audio connection is released, the complete connection may alternatively be released. The AG always initiates connection release.

5 SERIAL PORT PROFILE

This profile requires compliance with the Serial Port Profile. The following text together with the associated sub-clauses defines the requirements with regard to this profile, in addition to the requirements as defined in the Serial Port Profile.

As with the headset profile, both the AG and the HS can initiate connection establishment. For the purposes of reading the Serial Port Profile, both the AG and the HS can assume the role of Device A and B.

5.1 RFCOMM INTEROPERABILITY REQUIREMENTS

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

5.2 L2CAP INTEROPERABILITY REQUIREMENTS

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

5.3 SDP INTEROPERABILITY REQUIREMENTS

This profile defines following service records for the headset and the audio gateway respectively.

The codes assigned to the mnemonics used in the Value column as well as the codes assigned to the attribute identifiers (if not specifically mentioned in the AttrID column) can be found in the Bluetooth Assigned Numbers section.

Item	Definition	Type	Value	AttrID	Status	Default
ServiceClassIDList					M	
ServiceClass0		UUID	Headset		M	
ServiceClass1		UUID	Generic Audio		M	
ProtocolDescriptorList					M	
Protocol0		UUID	L2CAP		M	
Protocol1		UUID	RFCOMM		M	
Protocol Specific Parameter0	Server Channel	Uint8	N=server channel #		M	
BluetoothProfile DescriptorList					O	
Profile0	Supported Profiles	UUID	Headset		M	Headset
Param0	Profile Version	Uint16	0x0100*		M	0x0100
ServiceName	Display-able Text name	String	Service-provider defined		O	‘Headset’
Remote audio volume control		Bool-ean	True/False		O	False

Table 5.1: Service Record for Headset

*. Indicating version 1.0

Item	Definition	Type	Value	AttrID	Status	Default
ServiceClassIDList					M	
ServiceClass0		UUID	Headset		M	
ServiceClass1		UUID	Generic Audio		M	
ProtocolDescriptorList					M	
Protocol0		UUID	L2CAP		M	
Protocol1		UUID	RFCOMM		M	
Protocol Specific Parameter0	Server Channel	Uint8	N=server channel #		M	
BluetoothProfile DescriptorList						
Profile0	Supported Profile	UUID	Headset		M	Headset
Param0	Profile Version	Uint16	0x0100*		M	0x0100
ServiceName	Display-able Text name	String	Service-provider defined		O	'Voice gateway'

Table 5.2: Service Record for the Audio Gateway

*. Indicating version 1.0

5.4 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

In addition to the requirements for the Link Manager as stated in the "Serial Port Profile" on page 165, this profile mandates support for SCO links, in both the HS and AG.

5.5 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

In the table below, changes to the support status as listed in the Serial Port Profile, Section 8, Table 8.1 on page 184 are listed.

	Capability	Support in baseband	Support in AG	Support in HS
1.	Inquiry	M		X
2.	Inquiry scan	M	X	
3.	Paging	M		
4.	Page scan			
A	Type R0	M		
B	Type R1	M		
C	Type R2	M		
7.	Voice codec			
C	CVSD	O	M	M

Table 5.3: LC capabilities

5.5.1 Class of Device

A device which is active in the HS role shall, in the Class of Device field:

1. Set the bit ‘Audio’ in the Service Class field
2. Indicate ‘Audio’ as Major Device class
3. Indicate “Headset” as the Minor Device class

An inquiring AG may use this to filter the inquiry responses.

6 GENERIC ACCESS PROFILE

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

6.1 MODES

The table shows the support status for Modes within this profile.

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

Table 6.1: Modes

6.2 SECURITY ASPECTS

No changes to the requirements as stated in the Generic Access Profile.

6.3 IDLE MODE PROCEDURES

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

	Procedure	Support in HS	Support in AG
1	General inquiry	N/A	M
2	Limited inquiry	N/A	O
3	Name discovery	N/A	O
4	Device discovery	N/A	O
5	Bonding	M (Note 1)	M (Note 1)
Note 1: Mandatory for the AG to support initiation of bonding, and for the HS to accept bonding.			

Table 6.2: Idle mode procedures

7 REFERENCES

[1] International Telecommunication Union, “ITU-T Recommendation V.250”

[2] ETS 300 916 (GSM 07.07) version 5.6.0

8 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles 193
Figure 1.2: Arrows used in signalling diagrams 195
Figure 2.1: Protocol model 196
Figure 2.2: Headset profile, example with cellular phone 197
Figure 2.3: Headset profile, example with personal computer 197
Figure 4.1: Incoming audio connection establishment 202
Figure 4.2: Outgoing audio connection establishment 203
Figure 4.3: Audio connection release – HS initiated 203
Figure 4.4: Audio connection release – AG initiated 204
Figure 4.5: Audio connection transfer from AG to HS 204
Figure 4.6: Audio connection transfer from HS to AG 205
Figure 4.7: Audio volume control – example flow 205
Figure 4.8: Volume level synchronization – example flow 206
Figure 4.9: Connection establishment – Unparking a parked device 209
Figure 4.10: Connection release – Parking 209

9 LIST OF TABLES

Table 3.1: Application layer procedures....200
Table 3.2: Application layer feature to procedure mapping....200
Table 4.1: Mandatory AT capabilities....207
Table 4.2: Optional AT capabilities....207
Table 5.1: Service Record for Headset....211
Table 5.2: Service Record for the Audio Gateway....212
Table 5.3: LC capabilities....213
Table 6.1: Modes....214
Table 6.2: Idle mode procedures....214

Part K:7

DIAL-UP NETWORKING PROFILE

This profile defines the requirements for Bluetooth devices necessary for the support of the Dial-up Networking use case. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the Dial-up Networking use case.

CONTENTS

1 Introduction 223
1.1 Scope 223
1.2 Bluetooth Profile Structure 223
1.3 Symbols and conventions 224
1.3.1 Requirement status symbols 224
1.3.2 Signalling diagram conventions 225
1.3.3 Notation for timers and counters 225
2 Profile overview 226
2.1 Profile stack 226
2.2 Configurations and roles 227
2.3 User requirements and scenarios 228
2.4 Profile fundamentals 229
2.5 Conformance 229
3 Application layer 230
3.1 Service overview 230
3.2 Data calls 230
3.3 Fax service 230
3.4 Voice calls 230
4 Dialling and Control Interoperability Requirements 231
4.1 AT command set used 231
4.1.1 Command syntax 231
4.1.2 Commands 231
4.1.3 Result codes 233
4.2 Call progress audio feedback 233
4.3 Escape sequence 234
5 Serial Port Profile Interoperability Requirements 235
5.1 RFCOMM Interoperability Requirements 235
5.2 L2CAP Interoperability Requirements 235
5.3 SDP Interoperability Requirements 235
5.4 Link Manager (LM) Interoperability Requirements 236
5.5 Link Control (LC) Interoperability Requirements 237
5.5.1 Class of Device usage 237
6 Generic Access Profile Interoperability Requirements 238
6.1 Modes 238
6.2 Security aspects 238
6.3 Idle mode procedures 239
6.3.1 Bonding 239

7 References 240

8 List of Figures 241

9 List of Tables 242

1 INTRODUCTION

1.1 SCOPE

The Dial-up Networking Profile defines the protocols and procedures that shall be used by devices implementing the usage model called ‘Internet Bridge’ (see Bluetooth SIG MRD). The most common examples of such devices are modems and cellular phones.

The scenarios covered by this profile are the following:

- Usage of a cellular phone or modem by a computer as a wireless modem for connecting to a dial-up internet access server, or using other dial-up services
- Usage of a cellular phone or modem by a computer to receive data calls

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.

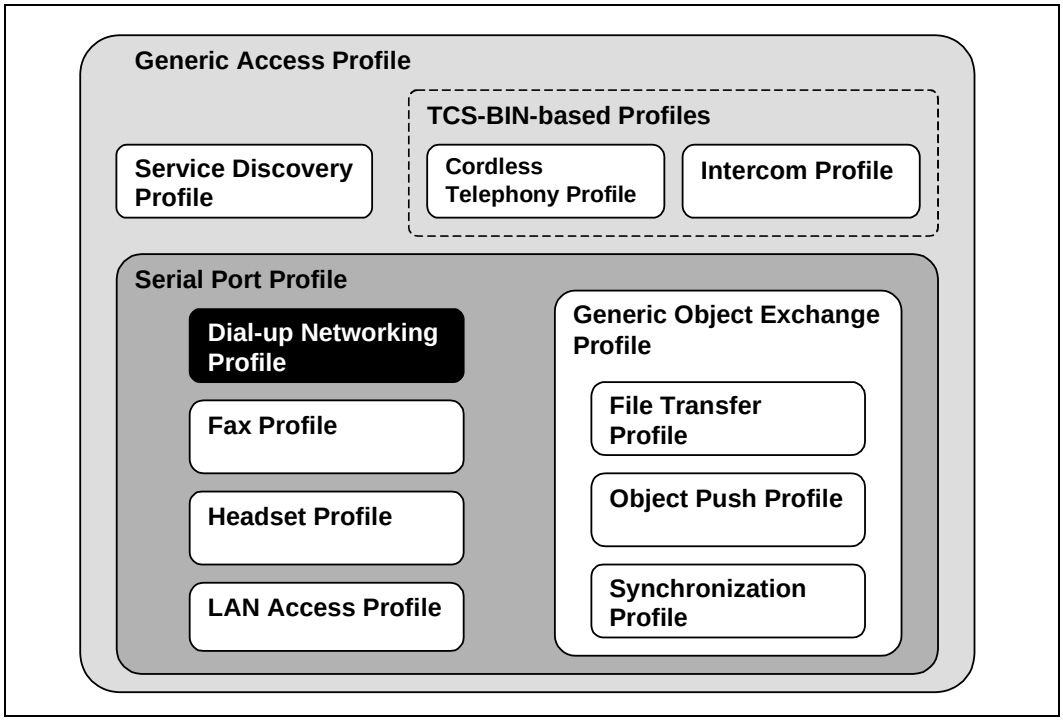


Figure 1.1: Bluetooth Profiles

1.3 SYMBOLS AND CONVENTIONS

1.3.1 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 will be used in case a certain other capability is supported);

‘X’ for excluded (used for capabilities that may be supported by the unit but which 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.

1.3.2 Signalling diagram conventions

The following arrows are used in diagrams describing procedures:

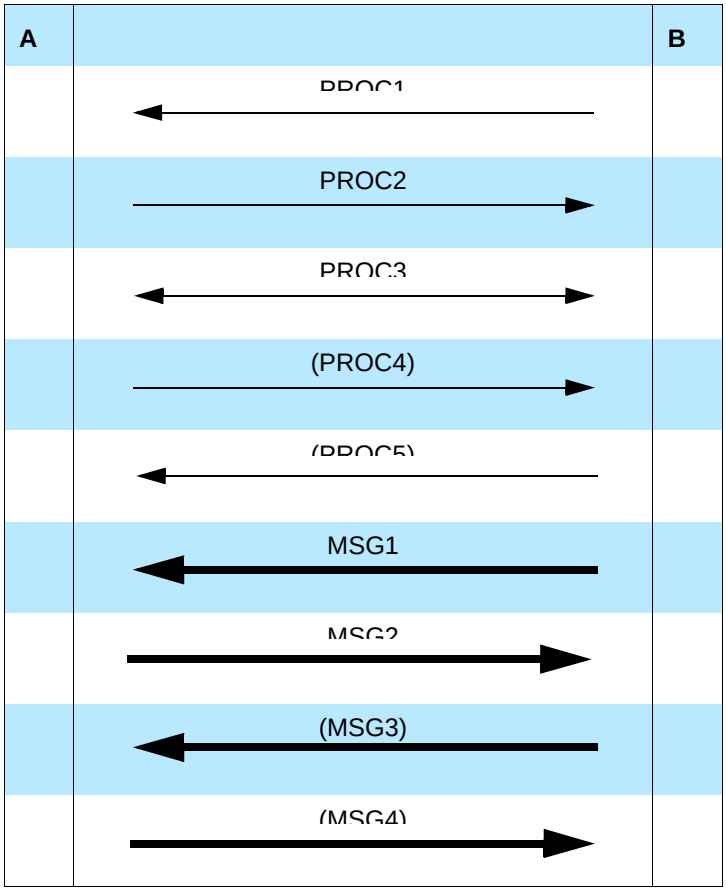


Table 1.1: Arrows used in signalling diagrams

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.

1.3.3 Notation for timers and counters

Timers and counters may be introduced specific to this profile. To distinguish them from timers (counters) used in the Bluetooth protocol specifications and other profiles, these timers (counters) are named in the following format: ‘T_{DNF}*nnn*’ (‘N_{DNF}*nnn*’).

2 PROFILE OVERVIEW

2.1 PROFILE STACK

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

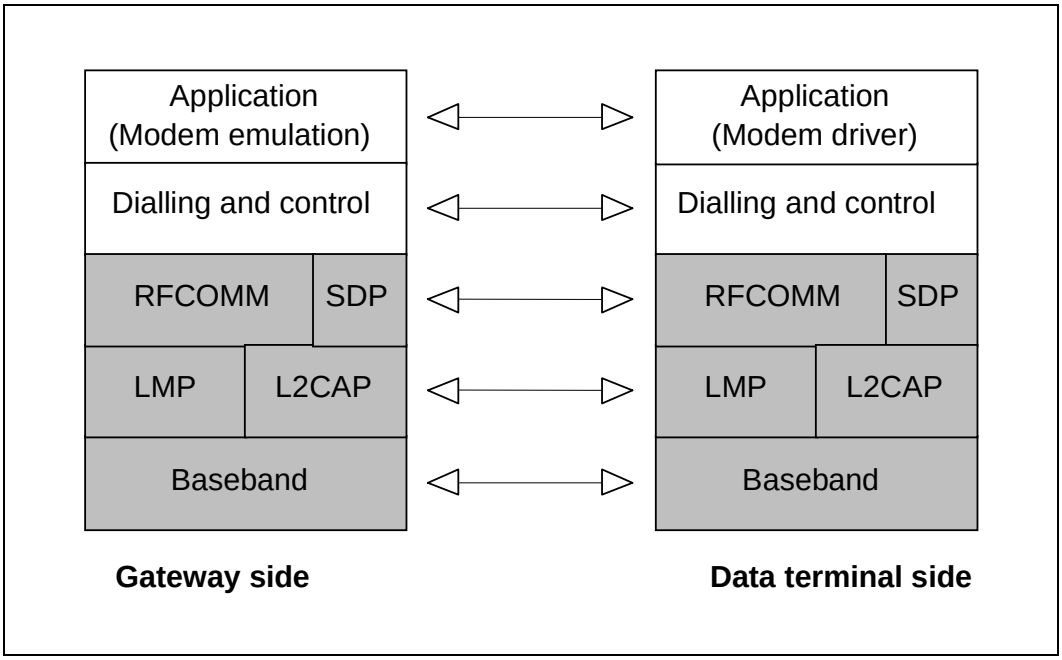


Figure 2.1: Protocol model

The Baseband, LMP and L2CAP are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM is the Bluetooth adaptation of GSM TS 07.10 [5], used for providing serial port emulation. SDP is the Bluetooth Service Discovery Protocol. Dialling and control (see Section 4) is the commands and procedures used for automatic dialling and control over the asynchronous serial link provided by the lower layers.

The modem emulation layer shown in Figure 2.1 is the entity emulating the modem, and the modem driver is the driver software in the data terminal.

For the shaded protocols/entities in Figure 2.1, The Serial Port Profile is used as base standard. For these protocols, all requirements stated in Serial Port Profile apply, except in those cases where this profile explicitly states deviations.

Note: Although not shown in the model above, it is assumed by this profile that the application layer has access to some lower layer procedures (for example SCO link establishment).

2.2 CONFIGURATIONS AND ROLES

The figures below show two typical configurations of devices for this profile:

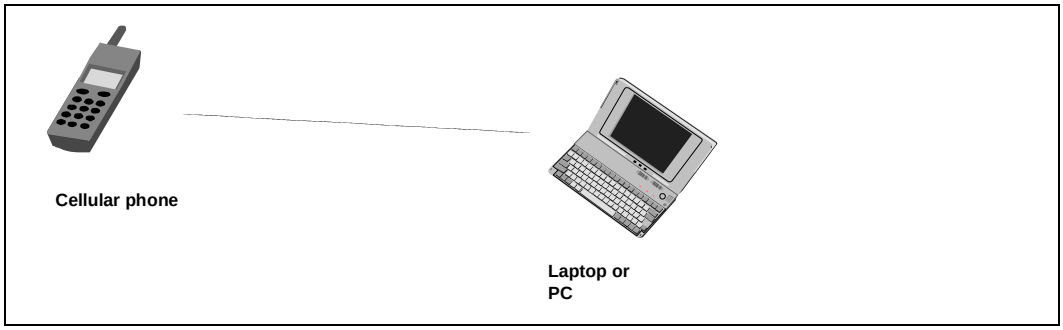


Figure 2.2: Dial-up Networking profile, example with cellular phone

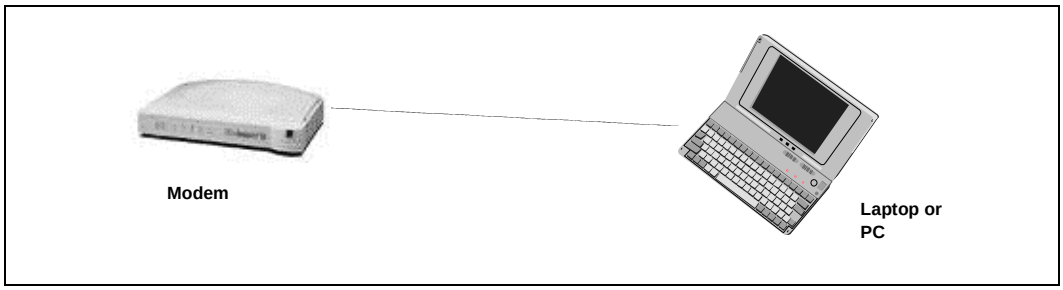


Figure 2.3: Dial-up Networking profile, example with modem

The following roles are defined for this profile:

Gateway (GW) – This is the device that provides access to the public network. Typical devices acting as gateways are cellular phones and modems.

Data Terminal (DT) – This is the device that uses the dial-up services of the gateway. Typical devices acting as data terminals are laptops and desktop PCs.

In the rest of this document, these terms are only used to designate these roles.

For purposes of mapping the Dial-up Networking profile to the conventional modem system architecture, the GW is considered Data Circuit Endpoint (DCE), and the DT is considered Data Terminal Endpoint (DTE).

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are the following:

- Usage of a GW by a DT as a wireless modem for connecting to a dial-up internet access server or using other dial-up services
- Usage of a GW by a DT to receive data calls

The following restrictions apply to this profile:

a) The modem is not required to be able to report and/or discriminate between different call types for incoming calls.

b) This profile requires support for one-slot packets only. This means that this profile ensures that data rates up to 128 kbps can be used. Support for higher rates are optional.

c) Only one call at a time is supported.

d) The profile only supports point-to-point configurations.

e) There is no way defined in this profile to discriminate between two SCO channels originating from the same device. It is therefore manufacturer-specific as to how to deal with the situation where there are multiple applications requiring the use of multiple SCO channels originating from the same device.

f) Before a cellphone or modem can be used with a PC/Laptop for the first time, an initialization procedure must be performed. This typically involves manually activating initialization support, and entering a PIN code on the PC/Laptop keyboard (see Generic Access Profile for more details). This procedure may have to be repeated under certain circumstances.

g) This profile does not support multiple instances of its implementation in the same device.

Security is ensured by authenticating the other party upon connection establishment, and by encrypting all user data. The baseband and LMP mechanisms for authentication and encryption are used.

2.4 PROFILE FUNDAMENTALS

Before a DT can use the services of a GW for the first time, the two devices have to initialize. Initialization includes exchanging a PIN code, creation of link keys and service discovery.

A link has to be established before calls can be initiated or received. This requires paging of the other device. Link establishment is always initiated by the DT.

There are no fixed master/slave roles.

The GW and DT provide serial port emulation. For the serial port emulation, the serial port profile (see Serial Port Profile) is used. The serial port emulation is used to transport the user data, modem control signals and AT commands between the GW and the DT. AT-commands are parsed by the GW and responses are sent to the DT.

An SCO link is used to transport audio.

For security purposes, authentication is used, and all user data is encrypted. For this, the baseband/LMP mechanisms are used.

2.5 CONFORMANCE

If conformance to this profile is claimed, all capabilities indicated 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 certification program.

3 APPLICATION LAYER

This section describes the service requirements on units active in the Dial-up Networking profile.

3.1 SERVICE OVERVIEW

Table 3.1 shows the required services:

	Services	Support in DT	Support in GW
1.	Data call without audio feedback	M	M
2.	Data call with audio feedback	O	O
3.	Fax services without audio feedback	N/A	N/A
4.	Fax services with audio feedback	N/A	N/A
5.	Voice call	N/A	N/A

Table 3.1: Application layer procedures

3.2 DATA CALLS

The support of data calls is mandatory for both GWs and DTs. Optionally, audio feedback may be provided (see Section 4.2).

The GW shall emulate a modem connected via a serial port. The Serial Port Profile is used for RS-232 emulation, and a modem emulation entity running on top of the serial port profile provides the modem emulation.

3.3 FAX SERVICE

The support of fax is not covered by this profile. Refer to Fax Profile.

3.4 VOICE CALLS

The support of voice calls is not covered by this profile.

4 DIALLING AND CONTROL INTEROPERABILITY REQUIREMENTS

4.1 AT COMMAND SET USED

To guarantee that basic functionality can always be provided, it is required that a GW device supports the commands and responses as defined in the following sub-clauses.

The commands are based on ITU-T V.250 and GSM 07.07.

4.1.1 Command syntax

For the exchange of the commands, responses and unsolicited results codes, the format, syntax and procedures of ITU-T V.250 [6] apply.

4.1.2 Commands

The table below lists all commands that shall be supported by the GW.

Name	Description	Reference
&C	Circuit 109 (Received line signal detector) Behavior	Shall be supported as defined in [6].
&D	Circuit 108 (Data terminal ready) Behavior	Shall be supported as defined in [6].
&F	Set to Factory-defined Configuration	Shall be supported as defined in [6].
+GCAP	Request Complete Capabilities List	Shall be supported as defined in [6].
+GMI	Request Manufacturer Identification	Shall be supported as defined in [6].
+GMM	Request Model Identification	Shall be supported as defined in [6].
+GMR	Request Revision Identification	Shall be supported as defined in [6].
A	Answer	Shall be supported as defined in [6].
D	Dial	Shall be supported *either* as defined in [6] *or* as defined in [10].
E	Command Echo	Shall be supported as defined in [6].
H	Hook Control	Shall be supported as defined in [6].
L	Monitor Speaker Loudness	Shall be supported as defined in [6].
M	Monitor Speaker Mode	Shall be supported as defined in [6].

Table 4.1: Required commands

Name	Description	Reference
O	Return to Online Data State	Shall be supported as defined in [6].
P	Select Pulse Dialling	Shall be supported as defined in [6].
Q	Result Code Suppression	Shall be supported as defined in [6].
S0	Automatic Answer	Shall be supported as defined in [6].
S10	Automatic Disconnect Delay	Shall be supported as defined in [6].
S3	Command Line Termination Character	Shall be supported as defined in [6].
S4	Response Formatting Character	Shall be supported as defined in [6].
S5	Command Line Editing Character	Shall be supported as defined in [6].
S6	Pause Before Blind Dialling	Shall be supported as defined in [6].
S7	Connection Completion Timeout	The setting of this parameter may be ignored. If not ignored, it shall be supported as defined in [6].
S8	Comma Dial Modifier Time	Shall be supported as defined in [6].
T	Select Tone Dialling	Shall be supported as defined in [6].
V	DCE Response Format	Shall be supported as defined in [6].
X	Result Code Selection and Call Progress Monitoring Control	Shall be supported as defined in [6].
Z	Reset To Default Configuration	Shall be supported as defined in [6].

Table 4.1: Required commands

4.1.3 Result codes

The table below lists all result codes that shall be supported by the GW.

Name	Description	Reference
OK	Acknowledges execution of a command.	Shall be supported as defined in [6].
CONNECT	Connection has been established.	Shall be supported as defined in [6].
RING	The DCE has detected an incoming call signal from the network.	Shall be supported as defined in [6].
NO CARRIER	The connection has been terminated, or the attempt to establish a connection failed.	Shall be supported as defined in [6].
ERROR	Error.	Shall be supported as defined in [6].
NO DIALTONE	No dial-tone detected.	Shall be supported as defined in [6].
BUSY	Busy signal detected.	Shall be supported as defined in[6].

Table 4.2: Required result codes

4.2 CALL PROGRESS AUDIO FEEDBACK

The GW or DT may optionally be able to provide audio feedback during call establishment. This clause applies only to gateways/data terminals that are able to provide audio feedback.

SCO links are used to transport the digitized audio over the Bluetooth link. The GW shall take all initiatives for SCO link establishment. The setting of the M parameter (see [6], Section 6.3.14) controls whether audio feedback is provided by the GW.

If a GW provides audio feedback for a call, the GW shall use the initiate SCO link procedure (see Link Manager protocol) to establish the audio link when the DCE goes off-hook.

Depending on the setting of the M parameter, the GW releases the audio link when the DCE has detected a carrier or when the DCE goes on-hook. The remove SCO link procedure (see [Link Manager protocol]) shall be used for audio link release.

If SCO link establishment fails, the call establishment shall proceed without the audio feedback.

This profile assumes that the DT is not active in any other profile which uses SCO links while it is operating in the Dial-up Networking profile. Therefore, the behavior in a situation where multiple SCO links are established simultaneously is undefined.

4.3 ESCAPE SEQUENCE

It is recommended that the GW supports an escape sequence (i.e. a sequence of characters which causes the GW to leave the online data state and go to the online command state). This profile does not mandate a particular escape sequence – it is up to the implementer of the profile if and how returning to command mode is supported.

5 SERIAL PORT PROFILE INTEROPERABILITY REQUIREMENTS

This profile requires compliance to the Serial Port Profile. For the purposes of reading the Serial Port Profile, the GW shall always be considered to be Device B and the DT shall always be considered to be Device A.

The following text together with the associated sub-clauses define the requirements with regards to this profile, in addition to the requirements defined in Serial Port Profile.

5.1 RFCOMM INTEROPERABILITY REQUIREMENTS

For RFCOMM, no additions to the requirements stated in Serial Port Profile apply.

5.2 L2CAP INTEROPERABILITY REQUIREMENTS

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

5.3 SDP INTEROPERABILITY REQUIREMENTS

Table 5.1 lists all entries in the SDP database of the GW defined by this profile. The ‘Status’ column indicates whether the presence of this field is mandatory or optional.

The codes assigned to the mnemonics used in the ‘Value’ column, and the codes assigned to the attribute identifiers, can be found in the Bluetooth Assigned Numbers section.

Item	Definition:	Type:	Value:	Status	Default
Service Class ID List				M	
Service Class #0		UUID	Generic Networking	O	
Service Class #1		UUID	Dial-up Networking	M	
Protocol Descriptor List				M	
Protocol #0		UUID	L2CAP	M	
Protocol #1		UUID	RFCOMM	M	

Table 5.1: Service Database Entries

Item	Definition:	Type:	Value:	Status	Default
Parameter for Protocol #1	Server Channel	UInt8	1,2,3,...,30	M	
Service Name	Displayable Text name	String	Service-provider defined	O	'Dial-up networking'
Audio Feedback Support		Boolean	True/False	O	False
BluetoothProfile-DescriptorList				M	
Profile #0		UUID	Dial-up Networking	M	
Parameter for Profile #0	Version	UInt16	0x0100*	O	0x100

Table 5.1: Service Database Entries

*. Indicating version 1.0

5.4 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

In addition to the requirements for the Link Manager as stated in the "Serial Port Profile" on page 165, this profile requires support for SCO links, in both the GW and DT. The support is conditional upon the ability to provide audio feedback."

5.5 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

In the table below, all LC capabilities required by this profile are listed.

	Capabilities	Support in baseband	Support in GW	Support in DT
5.	Packet types			
N	HV3 packet	O	C1	C2
7.	Voice codec			
C	CVSD	O	C1	C2
C1: The support for this capability is mandatory for gateways that are able to provide audio feedback to the DT. C2: The support for this capability is mandatory for data terminals that are able to provide audio feedback to the user.				

Table 5.2: Baseband/LC capabilities

5.5.1 Class of Device usage

A device which is active in the GW role of the Dial-up Networking profile shall, in the Class of Device field:

1. Set the bits ‘Telephony’ and ‘Networking’ in the Service Class field (see Bluetooth Assigned Numbers)
2. Indicate ‘Phone’ as Major Device class (see Bluetooth Assigned Numbers)

This may be used by an inquiring device to filter the inquiry responses.

6 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 Generic Access Profile.

6.1 MODES

The table shows the support status for Modes within this profile.

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

Table 6.1: Modes

6.2 SECURITY ASPECTS

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

	Procedure	Support in DT	Support in GW
1	Authentication	M	M
2	Security modes		
	Security mode 1	N/A	X
	Security mode 2	C1	C1
	Security mode 3	C1	C1
C1: Support for at least one of the security modes 2 and 3 is mandatory.			

Table 6.2: Security aspects

6.3 IDLE MODE PROCEDURES

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

	Procedure	Support in DT	Support in GW
1	General inquiry	M	N/A
2	Limited inquiry	O	N/A
3	Name discovery	O	N/A
4	Device discovery	O	N/A
5	Bonding	M (Note 1)	M (Note 1)
Note 1: See section 6.3.1			

Table 6.3: Idle mode procedures

6.3.1 Bonding

It is mandatory for the DT to support initiation of bonding, and for the GW to accept bonding.

7 REFERENCES

[1] Bluetooth Baseband specificationt

[2] Bluetooth Link Manager Protocol

[3] Bluetooth Logical Link Control and Adaptation Protocol Specification

[4] RFCOMM with TS 07.10

[5] TS 101 369 (GSM 07.10) version 6.1.0

[6] International Telecommunication Union, “ITU-T Recommendation V.250”

[7] Bluetooth Service Discovery Protocol

[8] John Webb, “Bluetooth SIG MRD”, version 1.0 Draft

[9] Bluetooth Serial Port Profile

[10] ETS 300 916 (GSM 07.07) version 5.6.0

[11] Bluetooth Fax Profile

[12] Bluetooth Assigned Numbers

8 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles 223
Figure 2.1: Protocol model 226
Figure 2.2: Dial-up Networking profile, example with cellular phone....227
Figure 2.3: Dial-up Networking profile, example with modem 227

9 LIST OF TABLES

Table 1.1: Arrows used in signalling diagrams 225
Table 3.1: Application layer procedures 230
Table 4.1: Required commands 231
Table 4.2: Required result codes 233
Table 5.1: Service Database Entries 235
Table 5.2: Baseband/LC capabilities 237
Table 6.1: Modes 238
Table 6.2: Security aspects 238
Table 6.3: Idle mode procedures 239

Part K:8

FAX PROFILE

This profile defines the requirements for Bluetooth devices necessary for the support of the Fax use case. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the Fax use case.

CONTENTS

1 Introduction 246
1.1 Scope 246
1.2 Profile Dependencies 246
1.3 Symbols and conventions 247
1.3.1 Requirement status symbols 247
1.3.2 Signalling diagram conventions 248

2 Profile overview 249
2.1 Profile stack 249
2.2 Configurations and roles 250
2.3 User requirements and scenarios 251
2.4 Profile fundamentals 251
2.5 Conformance 252

3 Application layer 253
3.1 Service overview 253
3.2 Data calls 253
3.3 Fax service 253
3.4 Voice calls 253

4 Dialling and Control Interoperability Requirements 254
4.1 AT command set used 254
4.1.1 Command syntax, Protocols and Result Codes 254
4.1.2 Fax Service Class selection procedure 254
4.2 Call progress audio feedback 255

5 Serial Port Profile 256
5.1 RFCOMM Interoperability Requirements 256
5.2 L2CAP Interoperability Requirements 256
5.3 SDP Interoperability Requirements 256
5.4 Link Manager (LM) Interoperability Requirements 257
5.5 Link Control (LC) Interoperability Requirements 258
5.5.1 Class of Device usage 258

6 Generic Access Profile Interoperability Requirements 259
6.1 Modes 259
6.2 Security aspects 260
6.3 Idle mode procedures 260
6.3.1 Bonding 260

7 References 261

8 List of Figures 262

9 List of Tables 263

1 INTRODUCTION

1.1 SCOPE

The Fax profile defines the protocols and procedures that shall be used by devices implementing the fax part of the usage model called ‘Data Access Points, Wide Area Networks’ (see Bluetooth SIG MRD).

A Bluetooth cellular phone or modem may be by a computer as a wireless fax modem to send or receive a fax message.

1.2 PROFILE DEPENDENCIES

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.

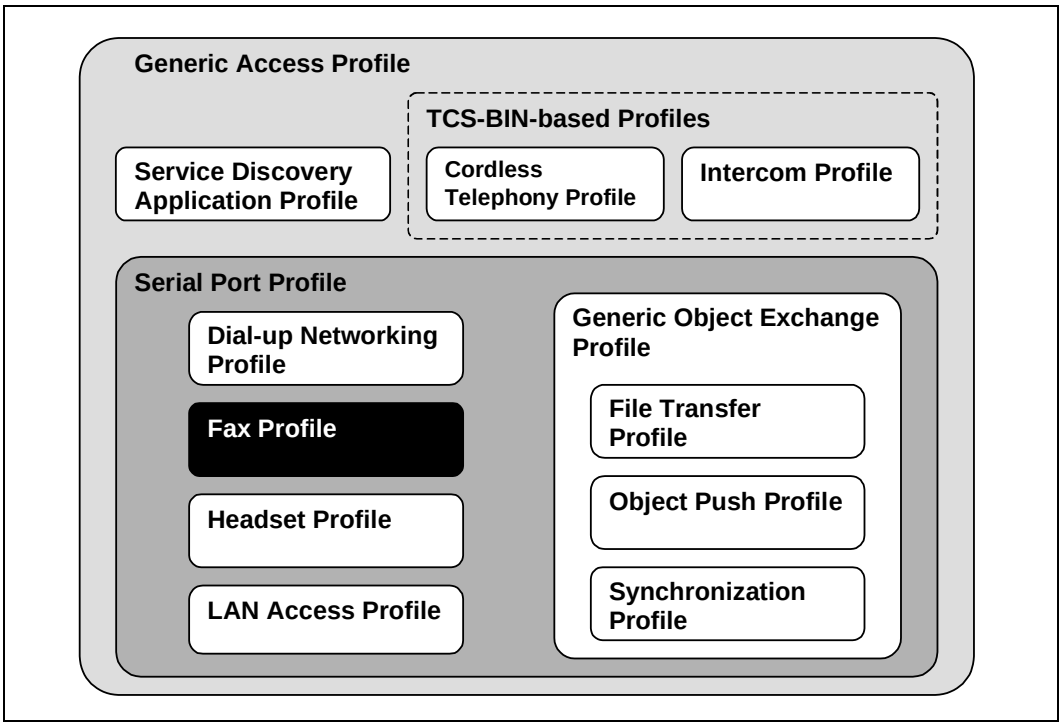


Figure 1.1: Bluetooth Profiles

As indicated in the figure, the Fax profile is dependent upon both the Serial Port Profile and the Generic access profile – details are provided in Section 5 Serial Port Profile on page 256 and Section 6 Generic Access Profile Interoperability Requirements on page 259.

1.3 SYMBOLS AND CONVENTIONS

1.3.1 Requirement status symbols

In this document, the following symbols are used:

- ‘M’ for mandatory to support
- ‘O’ for optional to support
- ‘X’ for excluded (used for capabilities that may be supported by the unit but which shall never be used in the 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.

Within the scope of this Fax profile, the expression ‘Fax class” is used as a shortcut to ‘facsimile service class” as defined by [2], [3], [4] and [4]. This is not to be confused with Bluetooth service class.

1.3.2 Signalling diagram conventions

The following arrows are used in diagrams describing procedures:

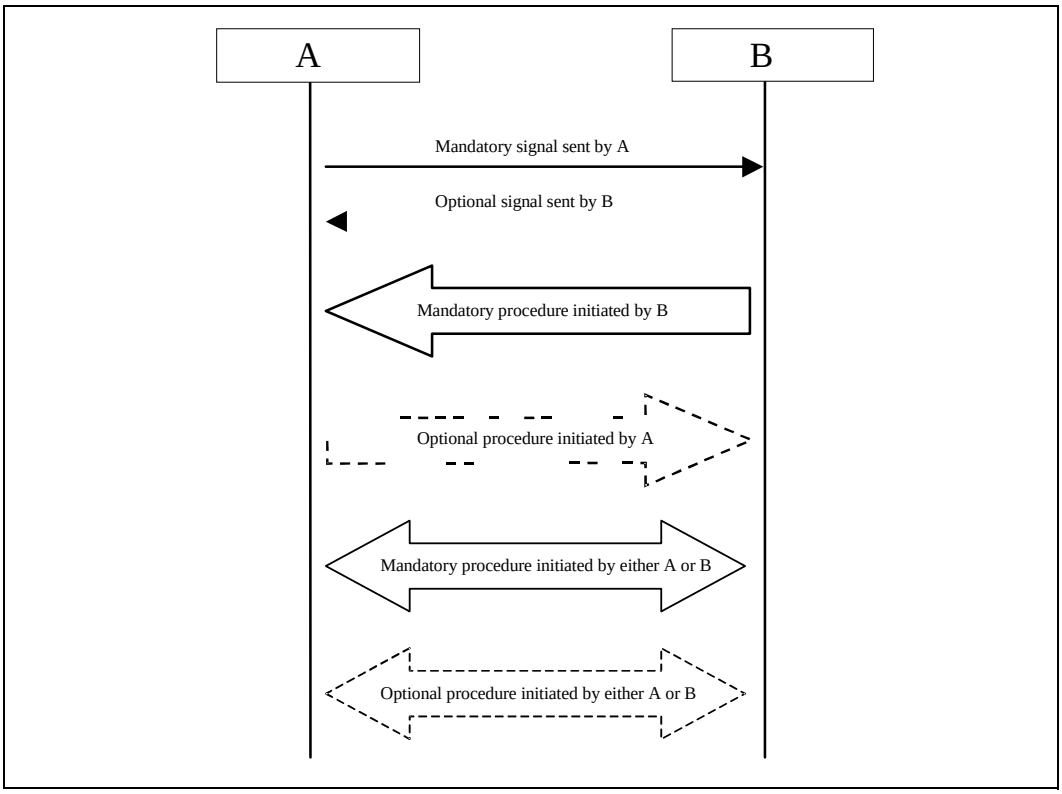


Figure 1-2 Arrows used in signalling diagrams

2 PROFILE OVERVIEW

2.1 PROFILE STACK

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

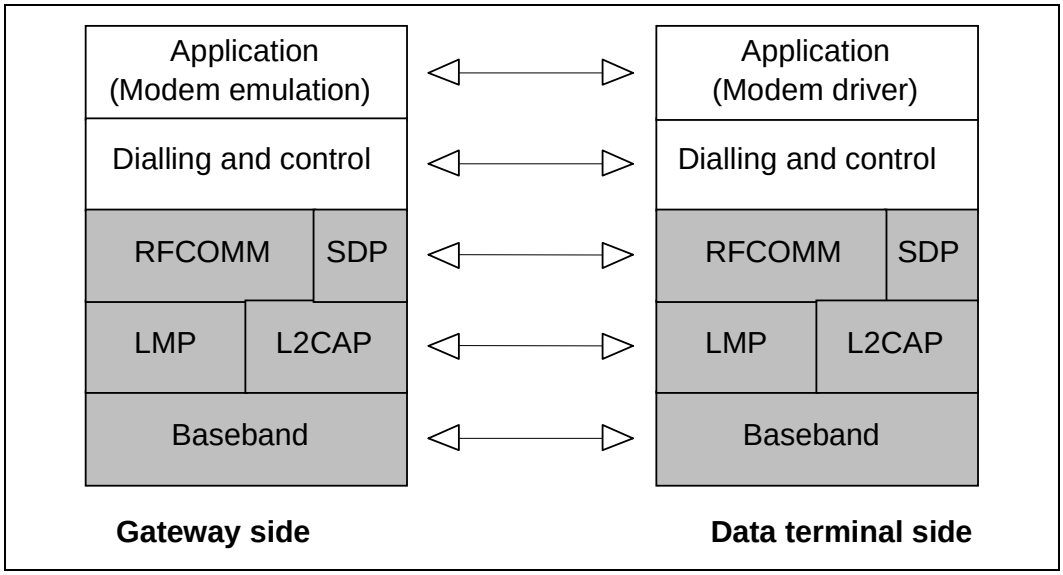


Figure 2.1: Protocol model

The Baseband, LMP and L2CAP are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM is the Bluetooth adaptation of GSM TS 07.10 [1], used for providing serial port emulation. SDP is the Bluetooth Service Discovery Protocol. Dialling and control (see Section 4) defines the commands and procedures used for automatic dialling and control over the asynchronous serial link provided by the lower layers.

The modem emulation layer shown in Figure 2.1 is the entity emulating the modem, and the modem driver is the driver software in the data terminal.

For the shaded protocols/entities in Figure 2.1, The Serial Port Profile is used as base standard. For these protocols, all requirements stated in Serial Port Profile apply, except in those cases where this profile explicitly states deviations.

Note: Although not shown in the model above, it is assumed by this profile that the application layer has access to some lower layer procedures (for example SCO link establishment).

2.2 CONFIGURATIONS AND ROLES

The figures below show two typical configurations of devices for this profile:

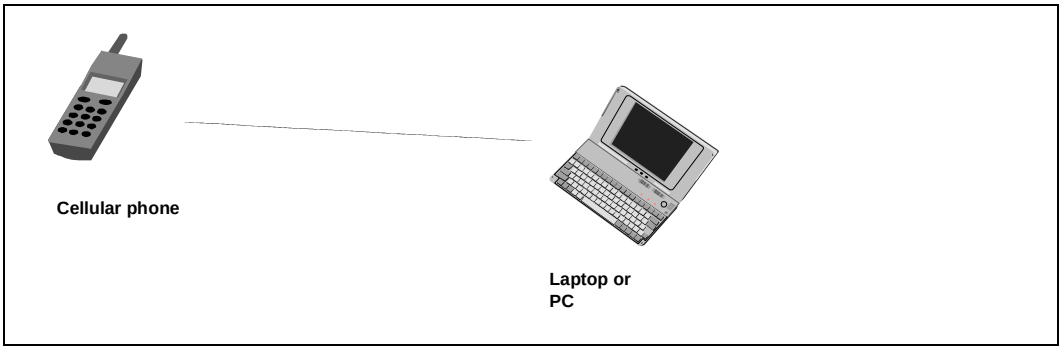


Figure 2.2: Fax profile, example with cellular phone

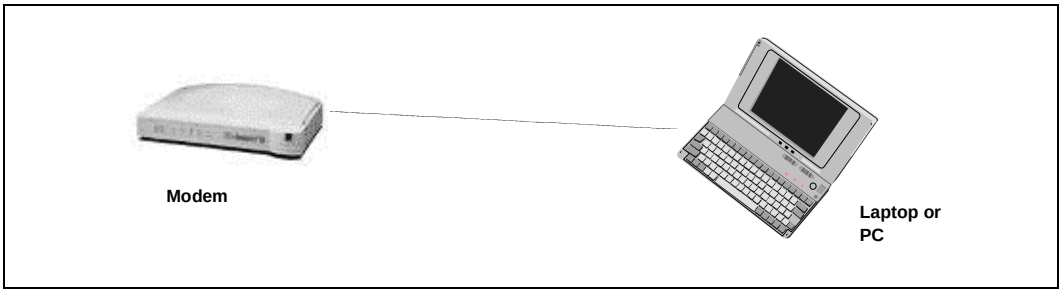


Figure 2.3: Fax profile, example with modem

The following roles are defined for this profile:

Gateway (GW) – This is the device that provides facsimile services. Typical devices acting as gateway are cellular phones and modems.

Data Terminal (DT) – This is the device that uses the facsimile services of the gateway. Typical devices acting as data terminals are laptops and desktop PCs.

In the rest of this document, these terms are only used to designate these roles.

For purposes of mapping the Fax profile to the conventional modem system architecture, the GW is considered Data Circuit Endpoint (DCE), and the DT is considered Data Terminal Endpoint (DTE).

2.3 USER REQUIREMENTS AND SCENARIOS

The Fax profile defines the usage of a GW by a DT as a wireless modem to send or receive fax messages

The following restrictions apply to this profile:

a) The GW (cellphone or modem) is not required to be able to report and/or discriminate between different call types for incoming calls.

b) This profile requires support for one-slot packets only. This means that this profile ensures that data rates up to 128 kbps can be used. Support for higher rates are optional.

c) Only one call at a time is supported.

d) The profile only supports point-to-point configurations.

e) Since in this profile there is no way defined to discriminate between 2 SCO channels originating from the same device, it is manufacturer specific as to deal with the situation where there are multiple applications requiring the use of multiple SCO channels originating from the same device.

f) This profile does not support multiple instances of its implementation in the same device.

2.4 PROFILE FUNDAMENTALS

Here is a brief summary of the interactions that take place when a DT wants to use the facsimile services of a GW.

1. If the DT does not have the Bluetooth Address of the GW, the DT has to obtain the address; e.g. using the Device discovery procedure, see Section 6.4 of Generic Access profile.
2. The Fax profile mandates the use of a secure connection through the authentication procedure (see Section 5.1 of Generic Access profile), and encryption of all user data through the baseband / LMP encryption mechanisms (see Section 8 of the Generic Access profile).
3. Link establishment is always initiated by the DT.
4. There are no fixed master / slave roles.
5. The fax call is established.
6. The GW and DT provide serial port emulation. For the serial port emulation, the serial port profile (see Serial Port Profile) is used. The serial port emulation is used to transport the user data, modem control signals and AT commands between the GW and the DT. AT-commands are parsed by the GW and responses are sent to the DT.
7. An optional SCO link may be used to transport fax audio feedback.
8. After the fax call has been cleared, the channel and link will be released as well.

2.5 CONFORMANCE

When conformance to this profile is claimed, all capabilities indicated 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 certification program.

3 APPLICATION LAYER

This section describes the service requirements on units active in the Fax profile.

3.1 SERVICE OVERVIEW

Table 3.1 shows the required services:

	Services	Support in DT	Support in GW
1.	Data call without audio feedback	N/A	N/A
2.	Data call with audio feedback	N/A	N/A
3.	Fax services without audio feedback	M	M
4.	Fax services with audio feedback	O	O
5.	Voice call	N/A	N/A

Table 3.1: Application layer procedures

3.2 DATA CALLS

The support of data calls is not covered by this profile.
Refer to Dial-up Networking Profile.

3.3 FAX SERVICE

At least one of the following fax classes of service is mandatory for both the GW and the paired DT (see Section 4.1.2):

Fax Class 1 TIA-578-A [2] and ITU T.31 [4]

Fax Class 2.0 TIA-592 [3]and ITU T.32 [5]

Fax Service Class 2 – No industry standard exists (manufacturer specific).

Optionally, audio feedback may be provided (see Section 4.2).

The GW shall emulate a modem connected via a serial port. The Serial Port Profile is used for RS-232 emulation, and RFCOMM running on top of the serial port profile provides the modem emulation.

3.4 VOICE CALLS

The support of voice calls is not covered by this profile. Refer to Cordless Telephony Profile.

4 DIALLING AND CONTROL INTEROPERABILITY REQUIREMENTS

4.1 AT COMMAND SET USED

To guarantee that basic functionality can always be provided, it is required that a GW device supports the commands and responses as defined in the supported fax class of service(s):

Fax Class 1 TIA-578-A [2] and ITU T.31 [4]

Fax Class 2.0 TIA-592 [3]] and ITU T.32 [5]

Fax Service Class 2 – No industry standard exists (manufacturer specific).

4.1.1 Command syntax, Protocols and Result Codes

Refer to each specific implemented fax service class document for a description of the required commands, protocols and result codes.

4.1.2 Fax Service Class selection procedure

This profile does not require a specific service class of fax. This profile supports 2 standards-based fax ‘classes’ – fax class 1 [2], [4] and fax class 2.0 [3], [5] – and a third manufacturer-specific pseudo-standard, fax class 2 (no industry reference standard exists).

The DT shall check the GW SDP or perform an ‘AT+FCLASS” command to discover the fax class of service(s) supported by the GW.

Bluetooth devices implementing this profile must support a minimum of one fax service class, but may support any or all fax services classes.

4.2 CALL PROGRESS AUDIO FEEDBACK

The GW or DT may optionally be able to provide audio feedback during call establishment. This clause applies only to gateways/data terminals that are able to provide audio feedback.

SCO links are used to transport the digitized audio over the Bluetooth link. The GW shall take all initiatives for SCO link establishment. The setting of the M parameter (see [6], Section 6.3.14) controls whether the GW provides audio feedback.

If a GW provides audio feedback for a call, the GW shall use the ‘initiate SCO link’ procedure (see Link Manager protocol) to establish the audio link when the DCE goes off-hook.

Depending on the setting of the M parameter, the GW releases the audio link when the DCE has detected a carrier or when the DCE goes on-hook. The ‘remove SCO link’ procedure (see [Link Manager protocol]) shall be used for audio link release.

If SCO link establishment fails, the call establishment shall proceed without the audio feedback.

This profile assumes that the DT is not active in any other profile that uses SCO links while it is operating in the Fax profile. Therefore, behavior is not defined for a situation where multiple SCO links are established simultaneously.

5 SERIAL PORT PROFILE

This profile requires compliance to the Serial Port Profile. For the purposes of reading the Serial Port Profile, the GW shall always be considered to be Device B and the DT shall always be considered to be Device A.

The following text together with the associated sub-clauses define the requirements with regards to this profile in addition to the requirements defined in the Serial Port Profile.

5.1 RFCOMM INTEROPERABILITY REQUIREMENTS

For RFCOMM, no additions to the requirements stated in Serial Port Profile apply.

5.2 L2CAP INTEROPERABILITY REQUIREMENTS

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

5.3 SDP INTEROPERABILITY REQUIREMENTS

Table 5.1 lists all entries in the SDP database of the GW defined by this profile. The ‘Status’ column indicates whether the presence of this field is mandatory or optional.

The codes assigned to the mnemonics used in the ‘Value’ column and the codes assigned to the attribute identifiers can be found in Bluetooth Assigned Numbers.

Item	Definition:	Type:	Value:	Status	Default
Service Class ID List				M	
Service Class #0		UUID	Generic Telephony	O	
Service Class #1		UUID	Fax	M	
Protocol Descriptor List				M	
Protocol #0		UUID	L2CAP	M	
Protocol #1		UUID	RFCOMM	M	
Parameter for Protocol #1	Server Channel	UInt8	N = server channel #	M	

Table 5.1: Service Database Entries

Item	Definition:	Type:	Value:	Status	Default
Service Name	Displayable Text name	String	Service-provider defined	O	'Fax'
Audio Feedback Support		Boolean	True/False	O	False
Fax Class 1 Support		Boolean	True/False	O	False
Fax Class 2.0 Support		Boolean	True/False	O	False
Fax Class 2 Support		Boolean	True/False	O	False
BluetoothProfile Descrip-torList				M	
Profile #0		UUID	Fax	M	
Parameter for Profile #0	Version	UInt16	0x0100*	O	0x100

Table 5.1: Service Database Entries

*. Indicating version 1.0

5.4 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

In addition to the requirements for the Link Manager as stated in the "Serial Port Profile" on page 165, this profile requires support for SCO links, in both the GW and DT. The support is conditional upon the ability to provide audio feedback."

5.5 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

In the table below, all LC capabilities required by this profile are listed.

	Capabilities	Support in baseband	Support in GW	Support in DT
5.	Packet types			
N	HV3 packet	O	C1	C2
7.	Voice codec			
C	CVSD	O	C1	C2
C1: The support for this capability is mandatory for gateways that are able to provide audio feedback to the DT. C2: The support for this capability is mandatory for data terminals that are able to provide audio feedback to the user.				

Table 5.2: Baseband/LC capabilities

5.5.1 Class of Device usage

A device which is active in the GW role of the Fax profile shall, in the Class of Device field:

1. Set the ‘Telephony’ bit in the Service Class field (see Bluetooth Assigned Numbers)
2. Indicate ‘Phone’ as Major Device class (see Bluetooth Assigned Numbers)

This may be used by an inquiring device to filter the inquiry responses.

6 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 Generic Access Profile.

6.1 MODES

The table shows the support status for Modes within this profile.

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

Table 6.1: Modes

6.2 SECURITY ASPECTS

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

	Procedure	Support in DT	Support in GW
1	Authentication	M	M
2	Security modes		
	Security mode 1	N/A	X
	Security mode 2	C1	C1
	Security mode 3	C1	C1
C1: Support for at least one of the security modes 2 and 3 is mandatory			

Table 6.2: Security aspects

6.3 IDLE MODE PROCEDURES

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

	Procedure	Support in DT	Support in GW
1	General inquiry	M	N/A
2	Limited inquiry	O	N/A
3	Name discovery	O	N/A
4	Device discovery	O	N/A
5	Bonding	M (Note 1)	M (Note 1)
Note 1: See section 6.3.1			

Table 6.3: Idle mode procedures

6.3.1 Bonding

It is mandatory for the DT to support initiation of bonding, and for the GW to accept bonding.

7 REFERENCES

[1] TS 101 369 (GSM 07.10) version 6.1.0

[2] TIA-578-A Facsimile Digital Interface. Asynchronous Facsimile DCE Control Standard, Service Class 1

[3] TIA-592 Facsimile Digital Interface. Asynchronous Facsimile DCE Control Standard, Service Class 2

[4] ITU T.31 Asynchronous Facsimile DCE Control – Service Class 1

[5] ITU T.32 Asynchronous Facsimile DCE Control – Service Class 2

[6] International Telecommunication Union, “ITU-T Recommendation V.250”

8 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles 246
Figure 2.1: Protocol model 249
Figure 2.2: Fax profile, example with cellular phone 250
Figure 2.3: Fax profile, example with modem 250

9 LIST OF TABLES

Table 3.1: Application layer procedures....253
Table 5.1: Service Database Entries....256
Table 5.2: Baseband/LC capabilities....258
Table 6.1: Modes....259
Table 6.2: Security aspects....260
Table 6.3: Idle mode procedures....260

Part K:9

LAN ACCESS PROFILE

This document is a LAN Access Profile for Bluetooth devices. Firstly, this profile defines how Bluetooth-enabled devices can access the services of a LAN using PPP. Secondly, this profile shows how the same PPP mechanisms are used to form a network consisting of two Bluetooth-enabled devices.

CONTENTS

1 Introduction 269
1.1 Scope 269
1.2 Profile Dependencies 270
1.3 Symbols and conventions 270
2 Profile overview 271
2.1 Protocol stack 271
2.2 Configurations and roles 271
2.3 User requirements and scenarios 272
2.4 Profile fundamentals 274
2.5 Conformance 274
3 User interface aspects 275
3.1 Security 275
3.2 Generic Modes 276
3.3 Additional Parameters 277
4 Application layer 278
4.1 Initialization of LAN Access Point service 278
4.2 Shutdown of LAN Access Point service 279
4.3 Establish LAN Connection 279
4.4 Lost LAN Connection 280
4.5 Disconnect LAN Connection 280
5 PPP 281
5.1 Initialize PPP 281
5.2 Shutdown PPP 282
5.3 Establish PPP Connection 282
5.4 Disconnect PPP Connection 282
5.5 PPP Authentication Protocols 283
6 RFCOMM 284
7 Service Discovery 285
7.1 SDP service records 285
8 L2CAP 287
9 Link Manager 288
9.1 Profile Errors 289
10 Link control 290

11 Management Entity Procedures 291
11.1 Link Establishment 291
11.1.1 No responses to inquiry 291
11.1.2 No response to paging 292
11.1.3 Pairing 292
11.1.4 Errors 292
11.2 Maximum Number of users 292
12 APPENDIX A (Normative): Timers and counters 293
13 APPENDIX B (Normative): Microsoft Windows 294
13.1 PC-2-PC configuration 294
14 APPENDIX C (Informative): Internet Protocol (IP) 295
14.1 IP Interfaces 295
14.1.1 Interface Enabled 295
14.1.2 Interface Disabled 295
14.2 The IPCP Protocol 295
14.2.1 IPCP Connection 296
14.2.2 IP Address Allocation 296
14.2.3 DNS and NBNS addresses 296
14.2.4 NetBIOS over IP 296
15 List of Figures 297
16 List of Tables 298
17 References 299
17.1 Normative references 299
17.2 Informative references 299

1 INTRODUCTION

1.1 SCOPE

This profile defines LAN Access using PPP over RFCOMM. There may be other means of LAN Access in the future.

- PPP is a widely deployed means of allowing access to networks. PPP provides authentication, encryption, data compression and multi-protocol facilities. PPP over RFCOMM has been chosen as a means of providing LAN Access for Bluetooth devices because of the large installed base of devices equipped with PPP software.
- It is recognized that PPP is capable of supporting various networking protocols (e.g. IP, IPX, etc.). This profile does not mandate the use of any particular protocol. However, since IP is recognized as the most important protocol used in today's networks, additional IP-related information is provided in an appendix. The use of these other PPP protocols is not discussed.
- This profile does not deal with conferencing, LAN emulation, ad-hoc networking or any other means of providing LAN Access. These functions are, or may be, dealt with in other Bluetooth profiles.

This profile defines how PPP networking is supported in the following situations.

a) LAN Access for a single Bluetooth device.

b) LAN Access for multiple Bluetooth devices.

c) PC to PC (using PPP networking over serial cable emulation).

1.2 PROFILE DEPENDENCIES

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile does have dependencies – direct and indirect – on the profile(s) within which it is contained, as illustrated in the figure.
In particular, this LAN Access profile is dependent on the Serial Port and Generic Access profiles.

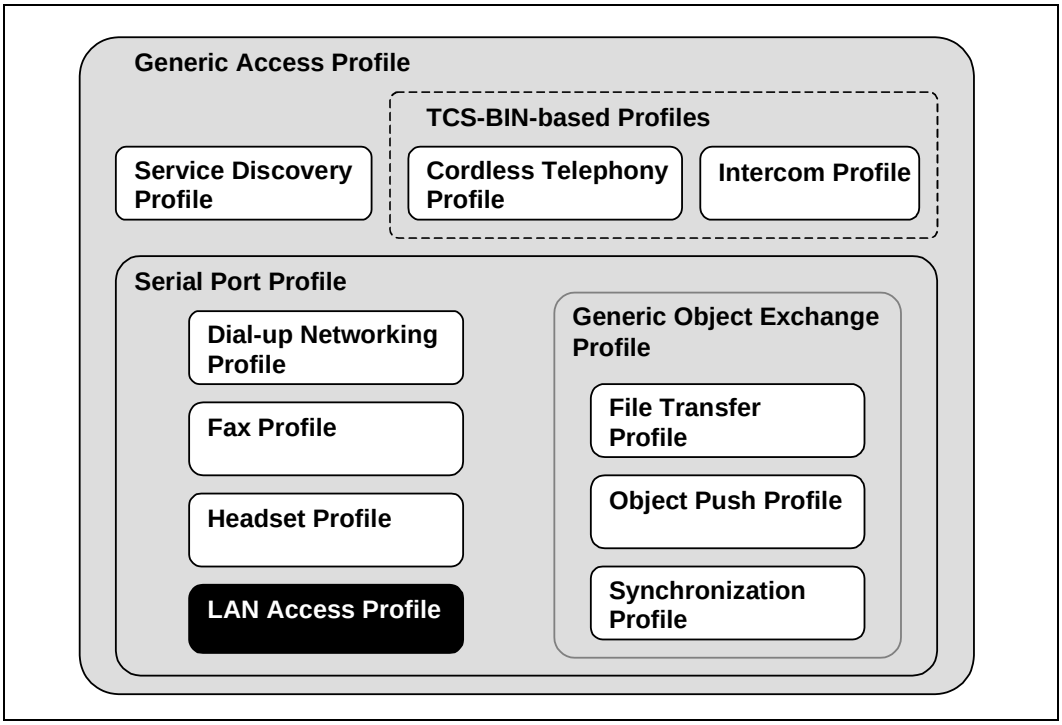


Figure 1.1: Bluetooth Profiles

1.3 SYMBOLS AND CONVENTIONS

This profile uses the symbols and conventions specified in Section 1.2 of the Generic Access Profile [13].

2 PROFILE OVERVIEW

2.1 PROTOCOL STACK

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

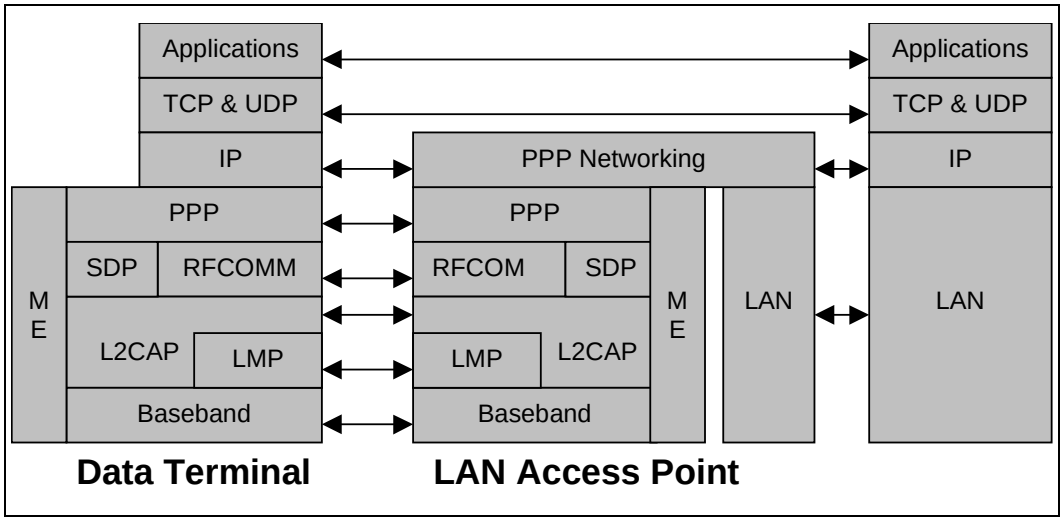


Figure 2.1: Protocol Stack

PPP is the IETF Point-to-Point Protocol [8]. PPP-Networking is the means of taking IP packets to/from the PPP layer and placing them onto the LAN. This mechanism is not defined by this profile but is a well-understood feature of Remote Access Server products.

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

ME is the Management Entity which coordinates procedures during initialization, configuration and connection management.

2.2 CONFIGURATIONS AND ROLES

The following roles are defined for this profile.

LAN Access Point **(LAP) –** This is the Bluetooth device that provides access to a LAN (e.g. Ethernet, Token Ring, Fiber Channel, Cable Modem, Firewire, USB, Home Networking). The LAP provides the services of a PPP Server. The PPP connection is carried over RFCOMM. RFCOMM is used to transport the PPP packets and it can also be used for flow control of the PPP data stream.

Data Terminal **(DT)** – This is the device that uses the services of the LAP. Typical devices acting as data terminals are laptops, notebooks, desktop PCs and PDAs. The DT is a PPP client. It forms a PPP connection with a LAP in order to gain access to a LAN.

This profile assumes that the LAP and the DT each have a single Bluetooth radio.[1]

2.3 USER REQUIREMENTS AND SCENARIOS

The following scenarios are covered by this profile.

1. A single DT uses a LAP as a wireless means for connecting to a Local Area Network (LAN). Once connected, the DT will operate as if it were connected to the LAN via dial-up networking. The DT can access all of the services provided by the LAN.

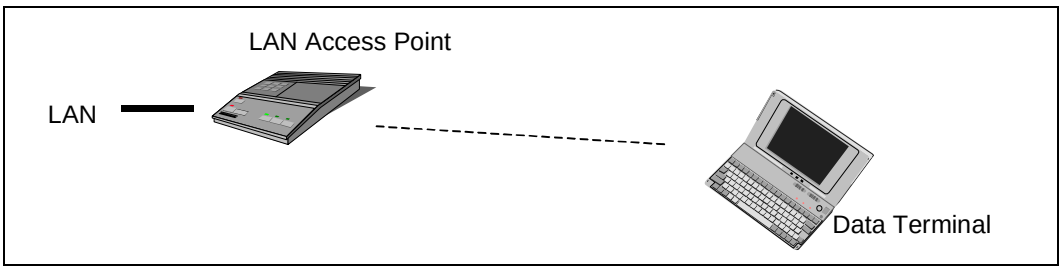


Figure 2.2: LAN Access by a single DT.

1. Products with multiple radios can still conform to this profile. The LAP and DT roles can be adopted independently by each radio.

2. Multiple DTs use a LAP as a wireless means for connecting to a Local Area Network (LAN). Once connected, the DTs will operate as if they were connected to the LAN via dial-up networking. The DTs can access all of the services provided by the LAN. The DTs can also communicate with each other via the LAP.[2]

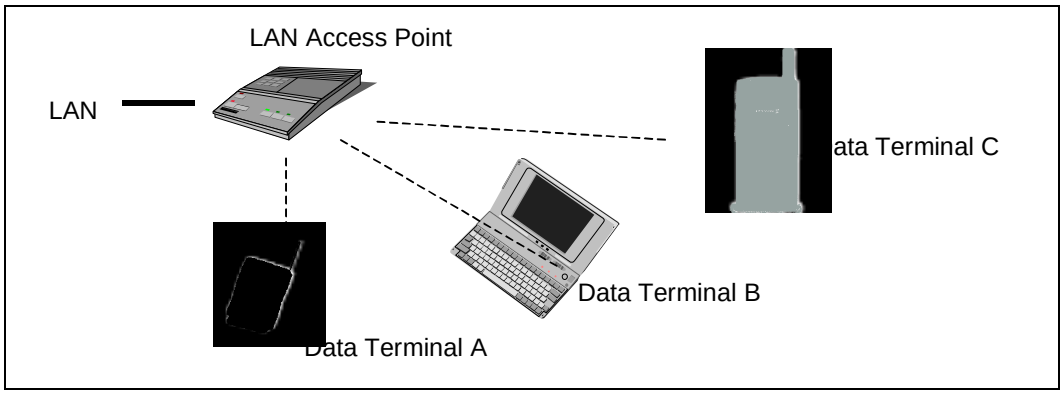


Figure 2.3: LAN Access by multiple DTs.

3. PC to PC connection. This is where two Bluetooth devices can form a single connection with each other. This scenario is similar to a direct cable connection commonly used to connect two PCs. In this scenario, one of the devices will take the role of a LAP, the other will be a DT. See Appendix 13.1 for more details of how this can be configured.

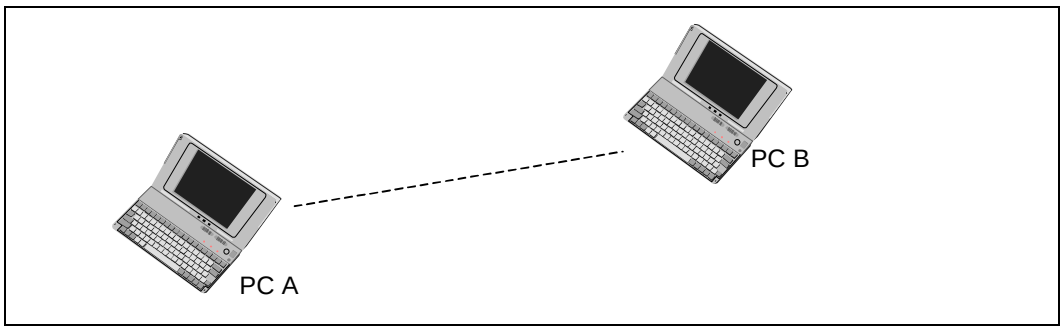


Figure 2.4: PC to PC connection.

Some LAP products may have an internal LAN or use the PSTN to access the Internet or corporate networks. The dial-up mechanisms to achieve the Internet connection are specific to the LAP. The DTs are totally unaware of these activities – except maybe in the event of longer connection times and traffic delays.

2. The DTs will be able to communicate with each other only if the required services (e.g. DNS) are available on the LAN.

2.4 PROFILE FUNDAMENTALS

Here is a brief summary of the interactions between a DT and a LAP. Subsequent sections in this profile provide more detail for each of the following steps.

1. The first step is to find a LAP that is within radio range and is providing a PPP/RFCOMM/L2CAP service. For example, the DT user could use some application to find and select a suitable LAP.
2. If there is no existing baseband physical link, then the DT requests a baseband physical link with the selected LAP. At some point after the physical link establishment, the devices perform mutual authentication. Each device insists that encryption is used on the link – see Section 3.1.
3. The DT establishes a PPP/RFCOMM/L2CAP connection.
4. Optionally, the LAP may use some appropriate PPP authentication mechanism (e.g. CHAP [21]). For example, the LAP may challenge the DT's user to authenticate himself or herself; the DT must then supply a username and password. If these mechanisms are used and the DT fails to authenticate itself, then the PPP link will be dropped.
5. Using the appropriate PPP mechanisms, a suitable IP address is negotiated between the LAP and the DT.
6. IP traffic can now flow across the PPP connection.
7. At any time the DT or the LAP may terminate the PPP connection.

2.5 CONFORMANCE

If conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies to 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 certification program.

3 USER INTERFACE ASPECTS

This profile is built upon the Generic Access Profile.

- When reading Generic Access Profile, DevA (the connection initiator) is equivalent to DT, and DevB is equivalent to the LAP.
- All the mandatory requirements defined in Generic Access Profile are mandatory for this profile.
- Unless otherwise stated below, all the optional requirements defined in Generic Access Profile are optional for this profile.

3.1 SECURITY

It is recognized that security in a wireless environment is of paramount importance.

Both the LAP and the DT must enforce that encryption is operating on the baseband physical link while any PPP traffic is being sent or received. The LAP and the DT will refuse any request to disable encryption. Therefore, Bluetooth pairing must occur as a means of authenticating the users. A PIN or link key must be supplied, even if the default PIN is used. The default PIN for LAN access is a zero length PIN. Failure to complete the pairing process will prevent access to the LAN Access service.

A more sophisticated product may require further authentication, encryption and/or authorization.

3.2 GENERIC MODES

The following modes are defined in Section 4 of Generic Access Profile [13]. This profile requires the following support.

Modes	Support in LAP	Support in DT
Discoverability modes		
Non-discoverable mode	O	X
Limited discoverable mode	X	X
General discoverable mode	M	X
Connectability modes		
Non-connectable mode	O	X
Connectable mode	M	X
Pairing modes		
Non-pairable mode	O	X
Pairable mode	M	X

Table 3.1: Generic Mode requirements table

Notes

1. A typical use for the Non-discoverable mode is where the LAP is intended for personal use only. The DT would remember the identity of the LAP and never need to use the Bluetooth inquiry mechanism.
2. A typical use for the General discoverable mode is where the LAP is intended for general use. The DT would not be expected to remember the identity of all the LAPs that it uses. The DT is expected to use the Bluetooth inquiry mechanism to discover the LAPs in range.

3.3 ADDITIONAL PARAMETERS

The following parameter is mandatory for the LAP. Optionally it may be configurable by the LAP administrator.

Maximum number of users. Different products have different capabilities and resource limitations that will limit the number of simultaneous users that they can support. The administrator of the LAP may choose to further limit the number of simultaneous users.[3]

- **Single-user mode** is when the maximum number of users is configured to allow only a single user. In this mode, either the DT or the LAP may be the master of the piconet. Single-user mode means that a single DT has exclusive access to a LAP.[4]
- **Multi-user mode** is when the maximum number of users is configured to allow more than one user. In this mode, the LAP must always become the master of the piconet. If the DT refuses to allow the LAP to become master, then the DT cannot gain access to the LAN.

3. The fewer simultaneous users there are using a Bluetooth radio, the more bandwidth will be available to each. A LAP can be restricted to a single user.
4. There are situations where a DT may wish to connect to a LAP and still remain the master of an existing piconet. For example, a PC is the master of a piconet with connections to a Bluetooth mouse and a Bluetooth video projector. The PC then requires a connection to the LAP, but must remain master of the existing piconet. If, for some reason, the PC can only be a member of one piconet, then the LAP must be a piconet slave. This situation is only possible if the LAP's 'maximum number of users' parameter has been configured to 1; i.e. single-user mode.

4 APPLICATION LAYER

Section	Feature	Support in LAP	Support in DT
4.1	Initialization of LAN Access service	M	X
4.2	Shutdown of LAN Access service	M	X
4.3	Establish LAN Connection	C1	M
4.4	Lost LAN Connection	X	M
4.5	Disconnect LAN Connection	M	M

Table 4.1: Application-layer requirements table

C1: Currently the LAP is not required to initiate a LAN connection establishment. In the future, a LAP may initiate a connection (e.g. as part of some form of LAP-initiated hand-off).

4.1 INITIALIZATION OF LAN ACCESS POINT SERVICE

This procedure initiates the configuration of the device as a LAP. This operation involves setting the following parameters:

- All the configurable parameters defined in Section 3.2. (For example, maximum number of users, discoverability mode, etc.)
- The required Bluetooth PINs and/or link keys.
- Any appropriate PPP configuration options (e.g. authentication, compression) can be configured. In order to ensure interoperability, a LAP shall not require connecting DTs to perform any PPP authentication, until the LAP administrator has configured PPP authentication.

When initialization is complete, the device will be able to accept PPP connections.

For products whose main role is that of a LAP, this initialization procedure is typically run automatically when the device is powered up.

For other products (e.g. PCs, Notebooks, etc.), this initialization procedure allows the user to configure the product as an access point, so that a DT can connect to it.

4.2 SHUTDOWN OF LAN ACCESS POINT SERVICE

This procedure stops the device from acting as a LAP.

- The PPP Server is shutdown – as defined in Section 5.2.
- Optionally, a product may take steps to prevent un-authorized Bluetooth access at a later time by deleting some of the stored link keys.

4.3 ESTABLISH LAN CONNECTION

Normally the DT will initiate the establishment of a connection to the LAN.

1. The first step is to select a LAP and a suitable PPP/RFCOMM service that it provides. This selection may be done in one of the following ways:

- The DT user is presented with a list of LAPs that are within radio range, and the services that they provide. The user can then select a LAP/service from the list provided.
- The DT user is presented with a list of services that are being provided by the LAPs that are within radio range. Where the same service is provided by multiple LAPs (i.e. identical ServiceClass-IDs), the application may choose to show the service only once. The user can then select a service from the list provided. The DT will automatically select a suitable LAP that provides the selected service.
- The DT user enters the name of the service that is required, e.g. ‘network’, or ‘Meeting #1’ (see Section 7.1 for more information on service names). The DT will automatically select a suitable LAP that provides the required service.
- Some application on the DT automatically searches for and selects a suitable LAP/service.

Whatever means is used, the result of the selection process must be a LAP that is within radio range and a PPP/RFCOMM service that it provides.

In all cases, the Bluetooth Service Discovery mechanisms are used to retrieve service information. This service information provides all the information required to create the RFCOMM connection in step 4.

2. Optionally, the DT user (or application) is allowed to enter a Bluetooth Authentication PIN or link key supplied by the application. If no PIN is entered, then a zero-length PIN is used.
3. Optionally, the DT user (or application) is allowed to enter a username and password for PPP authentication. If some PPP authentication mechanism is used and the user does not initially supply the username and password, then he/she may be prompted for them later in the connection establishment.

4. When the user (or application) activates the connection, then a PPP application is started, to attempt to establish a connection to the selected access point/service using the procedures in 12.1.

More complex situations (e.g. hand-off of a DT between LAPs) may require the LAP to initiate the establishment of a connection. These situations are possible, but are outside the scope of this document.

4.4 LOST LAN CONNECTION

If the LAN connection is lost for any reason, then the DT user (or application) must be notified of connection failure.

Optionally, the application may allow re-establishment of the connection to the same (or similar) LAP/service. The application could remember the previous LAP, service, PIN, link key, username and password and use them to allow speedy or automatic re-establishment of the LAN connection. The procedures described in Section 5.1 will be used.

4.5 DISCONNECT LAN CONNECTION

Either the LAP or the DT may terminate the LAN Connection at any time – using the procedures in Section 5.4.

5 PPP

PPP/RFCOMM operation in this profile is similar to PPP operation in normal dial-up networking, except that this profile omits the use of AT commands; PPP starts as soon as the RFCOMM link is established. By contrast, in dial-up networking, AT commands are used to establish the link, then PPP starts communicating.

The LAP exports a PPP Server interface [8]. This specification does not require any particular means of achieving the ‘appearance’ of a PPP Server. One implementation of a LAP could contain a PPP Server. Alternatively, the LAP could be some kind of PPP proxy, where PPP packets are transferred to/from a PPP server somewhere else on the network.

The following text, together with the associated sub-clauses, defines the mandatory requirements with regard to this profile.

Section	Procedure	Support in LAP	Support in DT
5.1	Initialize PPP	M	X
5.2	Shutdown PPP	M	X
5.3	Establish PPP Connection	M	M
5.4	Disconnect PPP Connection	M	M
5.5	PPP Authentication Protocols	O	O

Table 5.1: PPP capabilities

5.1 INITIALIZE PPP

On the LAP, the existence of a PPP Server shall be registered in the Service Discovery Database. The service attributes are defined in 7.1.

A device in the DT role does not register PPP in the Service Discovery Database. However, it is possible for a device to be both a LAP and a DT; therefore the device could register PPP in the Service Discovery Database as defined above.

PPP is a packet-oriented protocol, whereas RFCOMM expects serial data streams. Therefore, the PPP layer must use the serialization mechanisms described in [9].

5.2 SHUTDOWN PPP

All existing PPP connections are disconnected.

The PPP layer disables or removes the PPP service entry from the Service Discovery Database.

5.3 ESTABLISH PPP CONNECTION

If there is no existing RFCOMM session between the LAP and the DT, then the device initiating the PPP connection shall first initialize RFCOMM (see Section 6). The device obtains the RFCOMM Server channel to use from the service information it discovered earlier.

Using the Link Control Protocol (LCP) [8], the LAP and DT negotiate a PPP link.

Part of the LCP is the negotiation of the Maximum Transmission Unit (MTU) to be used on the PPP link – see [8] for details. This profile places no requirements on the negotiated MTU.[5]

Depending upon its capabilities and configuration (see 3.2), a LAP may have multiple PPP sessions in operation simultaneously.

5.4 DISCONNECT PPP CONNECTION

The following reasons will cause PPP to terminate the connection:

1. User intervention.
2. Failure of the RFCOMM/L2CAP connection. The RFCOMM/L2CAP connection may fail for several reasons. For example, when the radio link has failed or the device has been out of range for an excessive amount of time; see [3].
3. Termination by the LAP, if the access point can no longer provide the appropriate service. The reasons that would cause this are very dependent on the implementation of the LAP, but they could include (a) detection of duplicate IP addresses, (b) loss of connection to the LAN, (c) loss of connectivity to the PPP Server, or (d) loss of connection to the required IP subnet.
4. Some implementation-specific policy decision made by an application that is running on the LAP or the DT.

PPP handles each of the above situations differently. Reasons 1, 3 and 4 above result in a controlled disconnection at each protocol layer. Reason 2 above requires different processing.

5. Some products may use the LCP negotiation process to insist on specific values for the MTU. For example, a simple LAP with an Ethernet connection may wish to have a suitable MTU, so that IP packets do not require fragmentation when transmitted from Bluetooth to Ethernet.

When the PPP connection is terminated, either by user intervention or automatically by the LAP, then the PPP layer takes the following steps:

1. Gracefully terminate the IPCP connections (as defined in [24]). This will cause the IP interface to be disabled.
2. Gracefully terminate the LCP connections (as defined in [8]),
3. Disconnect the RFCOMM connection (as defined in Serial Port Profile)

When the RFCOMM/L2CAP connection suddenly terminates, then the PPP layer takes the following steps:

1. Terminate the IPCP connections (as defined in [24]). This will cause the IP interface to be disabled.
2. Terminate the LCP connections (as defined in [8]).

5.5 PPP AUTHENTICATION PROTOCOLS

Optionally, a LAP may be configured to use one or more of the PPP authentication protocols. These protocols allow a network administrator to control access to the network. The use of these PPP protocols does not form part of this profile. They are mentioned here for information only.

PPP supports a number of authentication protocols including the following:

- PPP Challenge Handshake Authentication Protocol (CHAP) [21]
- Microsoft PPP CHAP Extensions [22]
- PPP Authentication [23]
- PPP Extensible Authentication Protocol (EAP) [23]

Typically, the user needs to supply a username and password in order to gain authorization to use the PPP connection. If the authentication fails, then the PPP connection is normally dropped.

The PPP authentication protocols are independent of the Bluetooth authentication mechanisms. A network administrator may choose to use any combination of the PPP and Bluetooth mechanisms.

6 RFCOMM

This section describes the requirements on RFCOMM in units complying with the LAN Access Profile.

This profile is built upon the Serial Port Profile [10]. The requirements defined in the Serial Port Profile Section 4, “RFCOMM Interoperability Requirements,” on page 177, apply to this profile.

- When reading [10], DevA (the connection initiator) is equivalent to DT and DevB is equivalent to the LAP.
- All the mandatory requirements defined in the Serial Port Profile Section 4 on page 177 are mandatory for this profile.
- All the optional requirements defined in the Serial Port Profile Section 4 on page 177 are optional for this profile.

In addition:

1. In order to maximize packet throughput, it is recommended that RFCOMM should make use of the 3 and 5 slot baseband packets.
2. As defined in [4] section 2, the speed of RFCOMM connections is not configurable by the user. RFCOMM will transfer the data as fast as it can. The actual transfer rate will vary, depending upon the other Bluetooth traffic on the baseband link. In particular, the connection speed will not be artificially held at some typical serial port value; e.g. 19200.

7 SERVICE DISCOVERY

A LAP will be capable of providing one or more services for connecting to a LAN. For example, different services could provide access to different IP subnets on the LAN. The DT's user must be able to choose which of the LAN access services he or she requires.

7.1 SDP SERVICE RECORDS

Each LAP will provide one Service Class for PPP/RFCOMM services. A LAP may contain multiple instances of this Service Class; e.g. access to multiple subnets. Where the access point provides more than one PPP/RFCOMM service, the service selection is based on service attributes. These services are made public via the SDP.

The service record will have the following attributes. The syntax and usage of these attributes is defined in [6].

Item	Definition	Mand. /Opt.	Type	Value	Default Value
ServiceClassIDList		Mand.			
ServiceClass0	UUID for "LAN Access using PPP"	Mand.	UUID	See [11]	See [11]
ProtocolDescriptorList		Mand.			
Protocol0	UUID for L2CAP protocol	Mand.	UUID	See [11]	See [11]
Protocol1	UUID for RFCOMM protocol	Mand.	UUID	See [11]	See [11]
Parameter0	Server channel	Mand.	UInt8	varies	varies
ProfileDescriptorList		Opt.			
Profile #0	Uuid for "LAN Access using PPP"		UUID	see [11]	see [11]
Parameter0	Version "1.00"		Uint16	0x0100	0x0100
ServiceName	Displayable name	Opt.	String	Configurable	'LAN Access using PPP'
ServiceDescription	Displayable Information	Opt.	String	Configurable	'LAN Access using PPP'
ServiceAvailability	Load Factor	Opt.	Uint8	Dynamic	Dynamic
IpSubnet	Displayable Information	Opt.	String	Configurable	Configurable

The actual values of universal attribute IDs are defined in the Assigned Numbers specification [11] section 4. Values that are of the type UUID are defined in the Assigned Numbers specification [11] section 4.

- The ServiceName attribute is a short user-friendly name for the service; e.g. ‘Corporate Network’, ‘Conference#1’, etc.
- The ServiceDescription attribute is a longer description of the service. For example. “This network is provided for our guests. It provides free Internet Access and printing services. No username or password are required.“
- The ServiceAvailability attribute may be used in conjunction with the LoadFactor field of the CoD defined for LAN Access Points – see [11] section 1.2.6.
- The IpSubnet attributeID is (0x0200). This attribute is a displayable string containing subnet definition of the network, e.g. “191.34.12.0/24”. The first 4 numbers define the IP subnet in dotted-decimal notation. The fifth number, after the “/” character, is the number of subnet bits to use in the subnet mask; e.g. 24 means a subnet mask of 255.255.255.0.

8 L2CAP

This section describes the requirements on L2CAP in units complying with the LAN Access Profile.

This profile is built upon the Serial Port Profile [10]. The requirements defined in the Serial Port Profile Section 5, “L2CAP Interoperability Requirements,” on page 179 apply to this profile.

- When reading [10], DevA (the connection initiator) is equivalent to DT and DevB is equivalent to the LAP.
- All the mandatory requirements defined in the Serial Port Profile section 5 on page 179 are mandatory for this profile.
- All the optional requirements defined in the Serial Port Profile Section 5 on page 179 are optional for this profile.

In addition:

1. The MTU used at the L2CAP layer is determined by the RFCOMM parameter ‘maximum frame size’ – see Section 6 on page 284.

9 LINK MANAGER

This section describes the requirements on Link Manager in units complying with the LAN Access Profile.

This profile is built upon the Serial Port Profile [10]. The requirements defined in the Serial Port Profile Section 7, “Link Manager (LM) Interoperability Requirements,” on page 183 apply to this profile.

- When reading [10], DevA (the connection initiator) is equivalent to DT and DevB is equivalent to the LAP.
- All the mandatory requirements defined in the Serial Port Profile Section 7 on page 183 are mandatory for this profile.
- The following optional requirements defined in the Serial Port Profile Section 7 on page 183 are mandatory for this profile.

Procedure	Support in LAP	Support in DT
Authentication	M	M
Pairing	M	M
Encryption	M	M
Request master/slave switch	M	X
Perform master/slave switch	M	M

Table 9.1: LMP procedures

- All the remaining optional requirements defined in the Serial Port Profile Section 7 on page 183 are optional for this profile.

In addition:

- For bandwidth reasons, it is advisable but not mandatory for both devices to use multi-slot packets.
- When the LAP is configured in single-user mode (i.e. maximum number of users is 1), then the LAP may be either the master or the slave of the piconet.
- When the LAP is configured in multi-user mode (i.e. maximum number of users is more than 1), then the LAP must be the master of the piconet.

9.1 PROFILE ERRORS

The LAP must deny access to the PPP service if the DT fails to comply with the requirements of this profile, as follows:

1. Failure to complete the pairing process.
2. Failure to comply with a request to enable encryption on the baseband connection.
3. Failure by the DT to comply with a request to perform a master/slave switch. The LAP only requests a master/slave switch when it is configured in multi-user mode. In this mode the LAP must be the master of the piconet.

The LAP must reject all attempts by the DT to perform the following operations (see [2] section 5.1.2 for the appropriate LMP rejection reasons):

4. Requesting that encryption be disabled. The error code “Host Rejected due to security reasons” is used.
5. Requesting that the LAP switch to be a slave when the LAP is configured to be in multi-user mode. The error code “Unspecified Error” is used.
6. Requesting that a new connection be made when the LAP already has its configured maximum number of users. The error code “Other End Terminated Connection: Low Resources” is used.

10 LINK CONTROL

This section describes the requirements on Link Control in units complying with the LAN Access Profile.

This profile is built upon the Serial Port Profile [10]. The requirements defined in the Serial Port Profile, Section 8, “Link Control (LC) Interoperability Requirements,” on page 184, apply to this profile.

- When reading [10], DevA (the connection initiator) is equivalent to DT and DevB is equivalent to the LAP.
- All the mandatory requirements defined in the Serial Port Profile, Section 8 on page 184, are mandatory for this profile.
- All the optional requirements defined in the Serial Port Profile, Section 8 on page 184, are optional for this profile.
- The timer definitions defined in the Serial Port Profile, Section 8 on page 184, are not used in this profile.

In addition:

1. The Non-discoverable and General Discoverable Modes of the LAP (i.e. how InquiryScan is used) are defined in the Generic Access Profile [13], Section 4 on page 29.
2. In order to discover the nearby LAPs, a DT must use the General Inquiry procedure defined in the Generic Access Profile [13], Section 6 on page 37.

11 MANAGEMENT ENTITY PROCEDURES

The following text together with the associated sub-clauses defines the mandatory requirements with regard to this profile.

Section	Procedure	Support in LAP	Support in DT
11.1	Link Establishment	M	M
11.2	Single/Multi-user mode	M	N/A

Table 11.1: Management Entity Procedures

11.1 LINK ESTABLISHMENT

Link Establishment is required for communication between a LAP and a DT. The Link Establishment procedure is started as a direct consequence of the user operations described in "Establish LAN Connection" Section 4.3.

1. The DT first performs a General Inquiry to discover what LAPs are within radio range, see Generic Access Profile, Section 6 on page 37. Having performed the inquiry, the DT will have gathered a list of responses from nearby LAPs.
2. The DT sorts the list according to some product-specific criteria. The LAN Access Point CoD contains a field called 'Load Factor', see [11] section 1.2.6. It is recommended (but not mandated) that this field is used to sort the list.
3. The DT shall start with the LAP at the top of the list and try to establish a link with it, see Generic Access Profile, Section 7.1 on page 45. Any error or failure to establish a link shall cause the DT to skip this LAP. The DT will attempt to establish a link the next LAP in the list.
4. If there are no more LAPs in the list, the DT shall not proceed with further link establishment procedures. Link establishment has to be re-initiated.

The following subsections apply.

11.1.1 No responses to inquiry

If the DT did not get any response during inquiry, the DT shall not proceed with further link establishment procedures. Link establishment has to be re-initiated by the user or an application.

11.1.2 No response to paging

If a LAP does not respond to paging attempts, the DT shall skip this LAP.

11.1.3 Pairing

During link establishment, the LAP and DT are paired, which means that the DT and LAP build a security wall towards other devices.

11.1.4 Errors

If any LM procedure or Service Discovery procedure fails, or if link is lost at any time during link establishment, then the DT shall skip this LAP.

11.2 MAXIMUM NUMBER OF USERS

When the LAP is configured to allow multiple users, then the LAP must be the master of the piconet, see 3.2. In this mode, the Management Entity on the LAP ensures that the LAP remains the master of the Bluetooth piconet.

While in multi-user mode, the LAP shall request that it become the master of any new baseband physical link. If, for any reason, the LAP cannot remain the master, then the baseband physical link shall fail. The LMP [2] allows a device to (a) request a master/slave switch, and also (b) to refuse to comply with a request to perform a master/slave switch, see [1] section 10.9.3.

12 APPENDIX A (NORMATIVE): TIMERS AND COUNTERS

No specific timers are required by this profile.

Timer name	Recommended value	Description	Comment

Table 12.1: Defined timers

No specific counters are required by this profile.

Counter name	Proposed value	Description	Comment

Table 12.2: Defined Counters

The following parameters are required by this profile.

Parameter	Description
Discoverability mode	Controls whether the DT can discover the LAP.
Connectability mode	Controls whether the DT can be connected to the LAP.
Pairing mode	Controls whether the DT can be pair with the LAP.
Maximum users	The maximum the number of simultaneous users/connections.

Table 12.3: Defined parameters

13 APPENDIX B (NORMATIVE): MICROSOFT WINDOWS

This section contains various bits of information relating to Microsoft Windows and how it can be used in this profile.

13.1 PC-2-PC CONFIGURATION

This section contains information for configuring two PCs to form a connection. This configuration is independent of Bluetooth. This configuration is the same whether a serial cable or Bluetooth is used to make the connection.

- It is known that Windows '98 comes with a PPP server and that this PPP Server can be used to achieve the PC-to-PC feature. Detailed configuration information is available at the following Web sites.

  Microsoft Direct Cable Connection & Dial-up networking:

  http://support.microsoft.com/support/windows/ServiceWare/

  Win95/33BKKC22.ASP

  http://www.wown.com/

  http://www.tecno.demon.co.uk/dcc.html

  http://www.cs.purdue.edu/homes/kime/directcc/directcc95.htm

- This application requires some exchange of text strings before the PPP connection will become operational. The client PC sends the string ‘CLIENT’ and the server must reply with ‘CLIENTSERVER’.
- The tools provided by Windows ’98 configure one PC as the server and the other as the client. The PC configured as the server can share its resources with the client, but not vice versa.

14 APPENDIX C (INFORMATIVE): INTERNET PROTOCOL (IP)

The use of IP in this profile is optional. This section is provided for information only.

This section contains various bits of IP information that relate to various parts of this profile.

14.1 IP INTERFACES

14.1.1 Interface Enabled

The PPP layer in the DT will enable an IP interface when the IPCP link has been established and a suitable IP address has been negotiated. Typically, the DT will only have one PPP session in operation and only need one IP interface.

The DT may also need to configure its default gateway – WINS, DNS, etc. This profile does not define how this configuration is achieved. Mechanisms exist within PPP for supplying this information, see [24]. Other mechanisms may be used as appropriate.

In the event a DT has multiple IP interfaces, we rely on the IP protocol layer within the DT to select the correct interface to use for transmitting packets.

14.1.2 Interface Disabled

When the PPP connection is terminated or aborted, then the IP interface is disabled. The IP protocol stack will then remove that IP address from its routing tables.

14.2 THE IPCP PROTOCOL

Optionally, a LAP may be configured to support the IP protocol. The use of this PPP protocol does not form part of this profile. It is mentioned here for information only.

If supported, the IPCP protocol must be fully supported as defined in [24].

The following sub-sections concerning IPCP are informational only. They briefly describe certain aspects of IPCP. See [24] for full details.

14.2.1 IPCP Connection

IPCP only starts to operate after (a) the PPP connection has been established using LCP and optionally (b) the user has been authenticated.

The IPCP protocol negotiates certain parameters between the LAP and the DT.

Once the IPCP connection is established, and a suitable IP address has been negotiated, then IP interface is enabled.

14.2.2 IP Address Allocation

The DT will require an IP address in order to operate on the LAN. Current PPP implementations allow only three possibilities:

1. The IPCP option is used to specify a pre-configured IP address. If this IP address is not suitable on the LAN, then the IPCP link will not be established.
2. The IPCP option is used to request a suitable IP configuration from a PPP Server.
3. The IPCP Mobile-IP options are used to request a specified IP configuration. When moving between access points on the same LAN, it may be advantageous for the DT to continue using the same IP configuration.

14.2.3 DNS and NBNS addresses

Optionally, the LAP support could include the IPCP extensions defined in RFC1877 (defined by Microsoft). These extensions define the negotiation of primary and secondary Domain Name System (DNS) and NetBIOS Name Server (NBNS) addresses.

14.2.4 NetBIOS over IP

The NetBIOS protocol is used by Microsoft Windows to implement many of its networking features. The NetBIOS protocol can be carried over IP packets as defined in [29] and [30].

15 LIST OF FIGURES

Figure 1.1: Bluetooth Profiles 270
Figure 2.1: Protocol Stack 271
Figure 2.2: LAN Access by a single DT. 272
Figure 2.3: LAN Access by multiple DTs. 273
Figure 2.4: PC to PC connection. 273

16 LIST OF TABLES

Table 3.1: Generic Mode requirements table 276
Table 4.1: Application-layer requirements table 278
Table 5.1: PPP capabilities 281
Table 9.1: LMP procedures 288
Table 11.1: Management Entity Procedures 291
Table 12.1: Defined timers 293
Table 12.2: Defined Counters 293
Table 12.3: Defined parameters 293

17 REFERENCES

17.1 NORMATIVE REFERENCES

[1] Bluetooth Baseband specification (See Volume 1, Part B)

[2] Bluetooth Link Manager Protocol (See Volume 1, Part C)

[3] Bluetooth Logical Link Control and Adaptation Protocol Specification (See Volume 1, Part D)

[4] RFCOMM with TS 07.10 (See Volume 1, Part F:1)

[5] TS 101 369 (GSM 07.10) version 6.2.0.

[6] Bluetooth Service Discovery Protocol (SDP) (See Volume 1, Part E)

[7] John Webb, "Bluetooth SIG MRD", version 1.0.

[8] Simpson, W., Editor, "The Point-to-Point Protocol (PPP)", STD 50, RFC 1661, Daydreamer, July 1994.

[9] Simpson, W., Editor, "PPP in HDLC Framing", STD 51, RFC 1662, Daydreamer, July 1994.

[10] Serial Port Profile

[11] Bluetooth Assigned Numbers (See Volume 1,Appendix VIII)

[12] Thomas Muller, “Bluetooth Security Architecture”. Version 1.0.

[13] Generic Access Profile

17.2 INFORMATIVE REFERENCES

[20] Lloyd, B., and W. Simpson, "PPP Authentication Protocols", RFC 1334, Lloyd Internetworking, Daydreamer, October 1992.

[21] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)", RFC 1994, August 1996.

[22] Zorn, G., " Microsoft PPP CHAP Extensions", RFC 2433, October 1998.

[23] L. Blunk., " PPP Extensible Authentication Protocol (EAP)", RFC 2433, March 1998.

[24] McGregor, G., " The PPP Internet Protocol Control Protocol (IPCP)", RFC 1332, May 1992.

[25] Simpson, W., " The PPP Internetwork Packet Exchange Control Protocol (IPXCP)", RFC 1552, December 1993.

[26] Pall, G., " The PPP NetBIOS Frames Control Protocol (NBFCP)”, RFC 2097, January 1997.

[27] " Mobile-IPv4 Configuration Option for PPP IPCP ", RFC 2290.

[28] Cobb, S., " PPP Internet Protocol Control Protocol Extensions for Name Server Addresses”, RFC 1877, December 1995.

[29] NetBIOS Working Group, “PROTOCOL STANDARD FOR A NetBIOS SERVICE ON A TCP/UDP TRANSPORT: CONCEPTS AND METHODS”, RFC 1001, March 1987.

[30] NetBIOS Working Group, “PROTOCOL STANDARD FOR A NetBIOS SERVICE ON A TCP/UDP TRANSPORT: DETAILED SPECIFICATIONS”, RFC 1002, March 1987.

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

1 Introduction 306
1.1 Scope 306
1.2 Bluetooth Profile Structure 306
1.3 Bluetooth OBEX-Related Specifications 307
1.4 Symbols and conventions 308
1.4.1 Requirement status symbols 308
1.4.2 Signaling diagram conventions 309

2 Profile overview 310
2.1 Profile stack 310
2.2 Configurations and roles 310
2.3 User requirements and scenarios 311
2.4 Profile fundamentals 311

3 User interface aspects 312

4 Application layer 313
4.1 Feature Overview 313
4.2 Establishing an Object Exchange Session 313
4.3 Pushing a Data Object 313
4.4 Pulling a Data Object 313

5 OBEX Interoperability Requirements 314
5.1 OBEX Operations Used 314
5.2 OBEX Headers 314
5.3 Initialization of OBEX 315
5.4 Establishment of OBEX session 315
5.4.1 OBEX Session without Authentication 316
5.4.2 OBEX Session with Authentication 318
5.5 Pushing Data to Server 321
5.6 Pulling Data from Server 322
5.7 Disconnection 323

6 Serial Port Profile Interoperability Requirements 324
6.1 RFCOMM Interoperability Requirements 324
6.2 L2CAP Interoperability Requirements 324
6.3 SDP Interoperability Requirements 324
6.4 Link Manager (LM) Interoperability Requirements 324
6.5 Link Control (LC) Interoperability Requirements 324
6.5.1 Inquiry and Inquiry Scan 325

7 Generic Access Profile Interoperability Requirements 326

7.1 Modes 326

7.2 Security aspects 326

7.3 Idle mode procedures 327

7.3.1 Bonding 327

8 References 328

8.1 Normative references 328

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.

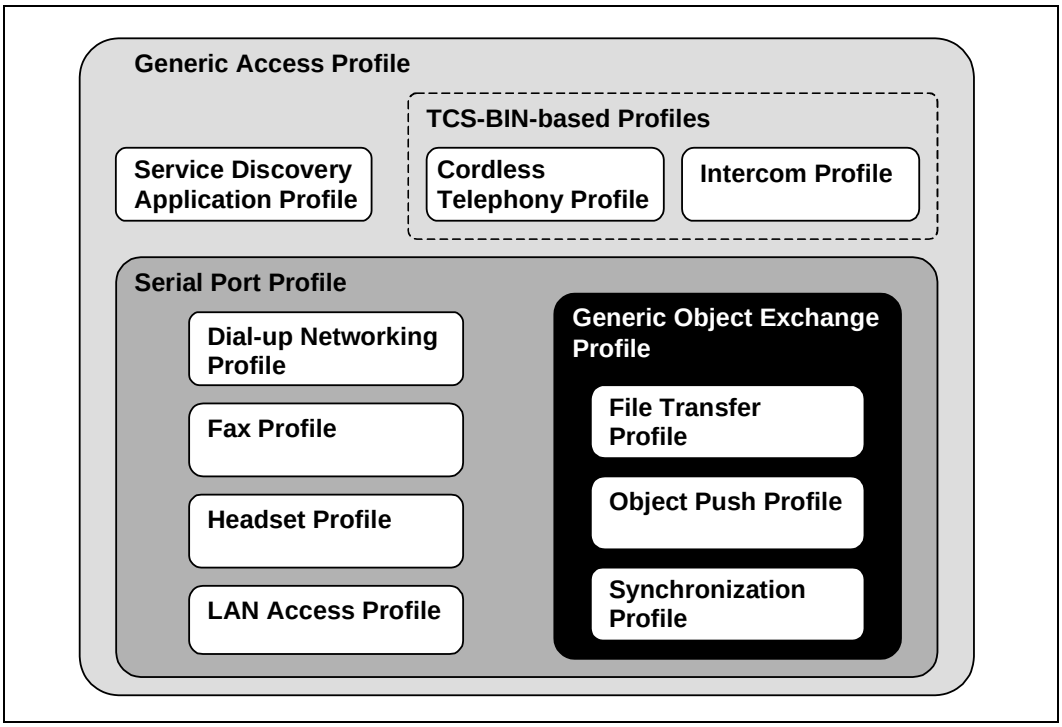


Figure 1.1: Bluetooth Profiles

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:

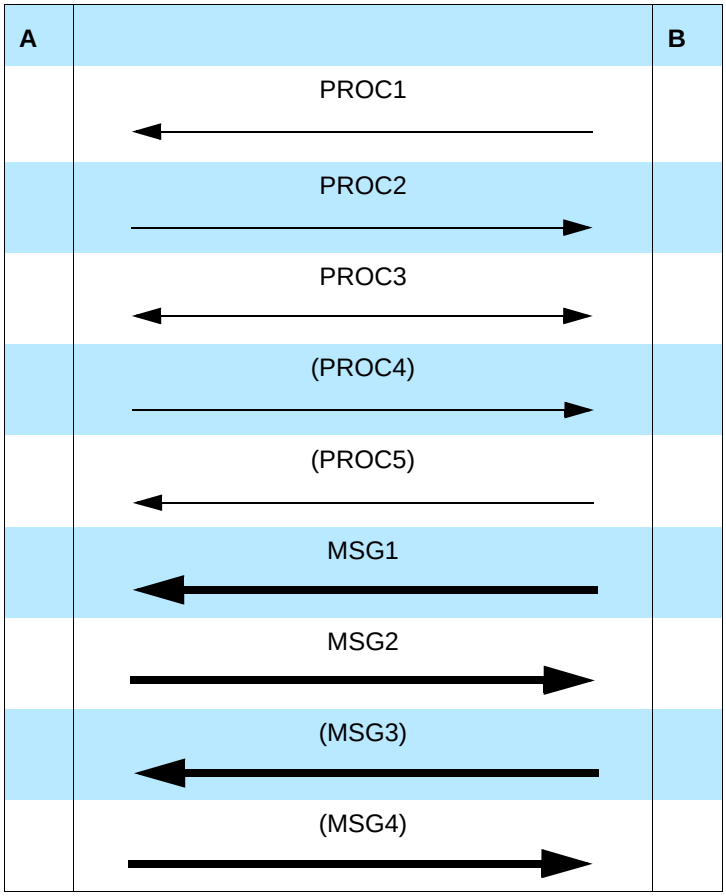


Table 1.1: Arrows used in signaling diagrams

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.

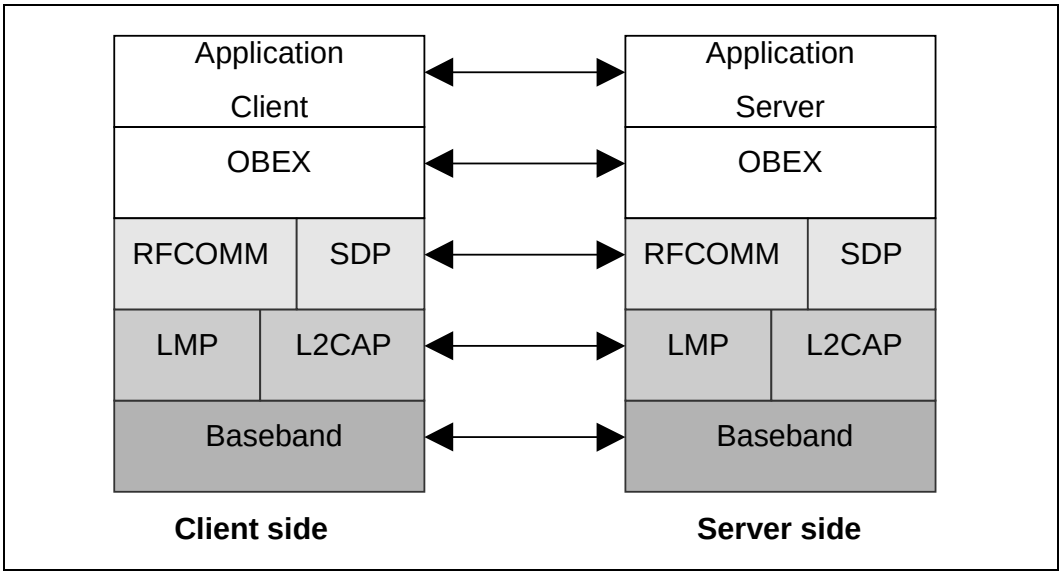


Figure 2.1: Protocol model

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 adata 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.

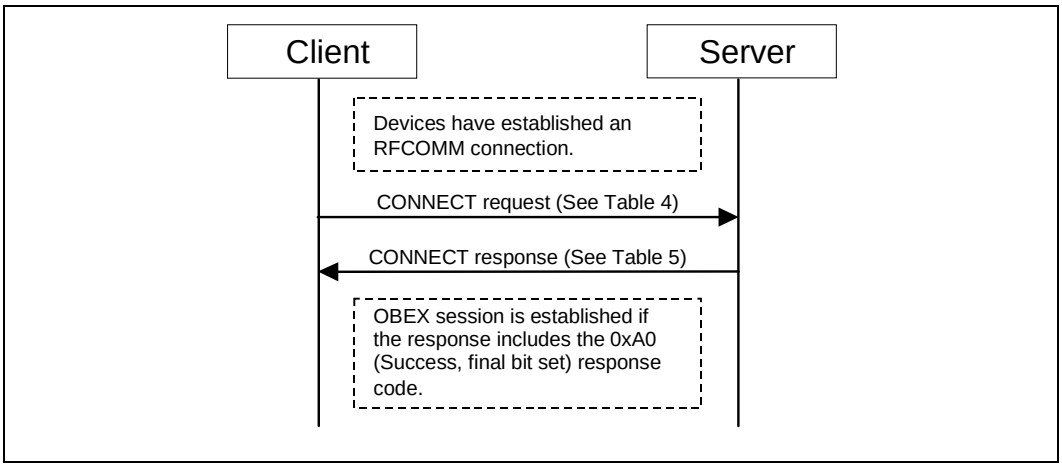


Figure 5.1: Establishment of OBEX Session without Authentication

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.

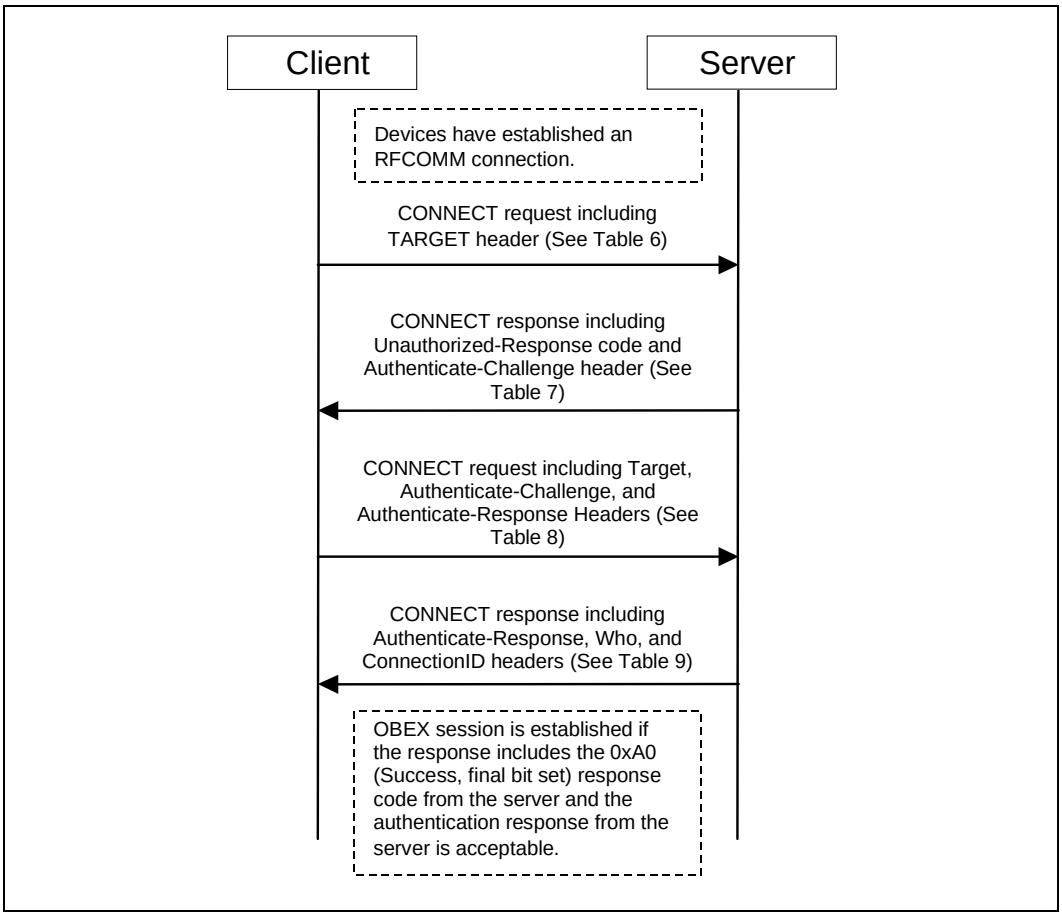


Figure 5.2: Establishment of OBEX Session with Authentication

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

Part K:11

OBJECT PUSH PROFILE

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

CONTENTS

1 Introduction 334
1.1 Scope 334
1.2 Bluetooth Profile Structure 334
1.3 Bluetooth OBEX-Related Specifications 335
1.4 Symbols and conventions 336
1.4.1 Requirement status symbols 336
1.4.2 Signaling diagram conventions 337

2 Profile overview 338
2.1 Profile stack 338
2.2 Configurations and roles 338
2.3 User requirements and scenarios 339
2.4 Profile fundamentals 339

3 User interface aspects 340
3.1 Mode selection, Push Servers 340
3.2 Function Selection, Push Clients 340
3.3 Application Usage Events 341
3.3.1 Object Push 341
3.3.2 Business Card Pull 341
3.3.3 Business Card Exchange 342

4 Application layer 344
4.1 Feature overview 344
4.2 Object Push Feature 344
4.2.1 Content Formats 344
4.2.2 Application Procedure 345
4.3 Business Card Pull Feature 345
4.3.1 Owner’s Business Card 346
4.3.2 Application Procedure Business Card Pull 346
4.4 Business Card Exchange Feature 346
4.4.1 Owner’s Business Card 346
4.4.2 Application Procedure Business Card Exchange 346

5 OBEX 348
5.1 OBEX Operations Used 348
5.2 OBEX Headers 348
5.2.1 OBEX Headers for the Object Push Feature 348
5.2.2 OBEX Headers for the Business Card Pull and Exchange Features349
5.3 Initialization of OBEX 350
5.4 Establishment of OBEX session 350
5.5 Pushing Data 350
5.6 Pulling Data 350
5.7 Disconnection 350
6 Service Discovery 351
6.1 SD Service Records 351
6.2 SDP Protocol Data Units 352
7 References 353
7.1 Normative references 353

FOREWORD

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

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

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

1 INTRODUCTION

1.1 SCOPE

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

The scenarios covered by this profile are the following:

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

1.2 BLUETOOTH PROFILE STRUCTURE

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

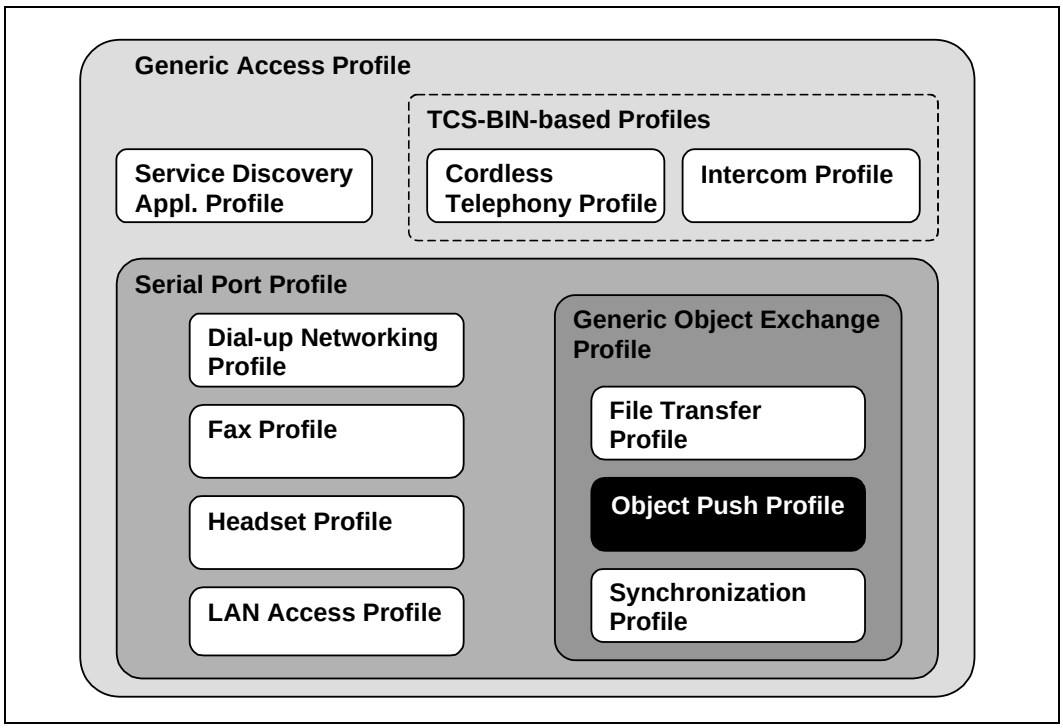


Figure 1.1: Bluetooth Profiles

1.3 BLUETOOTH OBEX-RELATED SPECIFICATIONS

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

1. Bluetooth IrDA Interoperability Specification [7].

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

2. Bluetooth Generic Object Exchange Profile Specification [10]

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

3. Bluetooth Synchronization Profile Specification [15]

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

4. Bluetooth File Transfer Profile Specification [14]

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

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

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

1.4 SYMBOLS AND CONVENTIONS

1.4.1 Requirement status symbols

In this document, the following symbols are used:

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

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

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

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

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

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

1.4.2 Signaling diagram conventions

The following arrows are used in diagrams describing procedures:

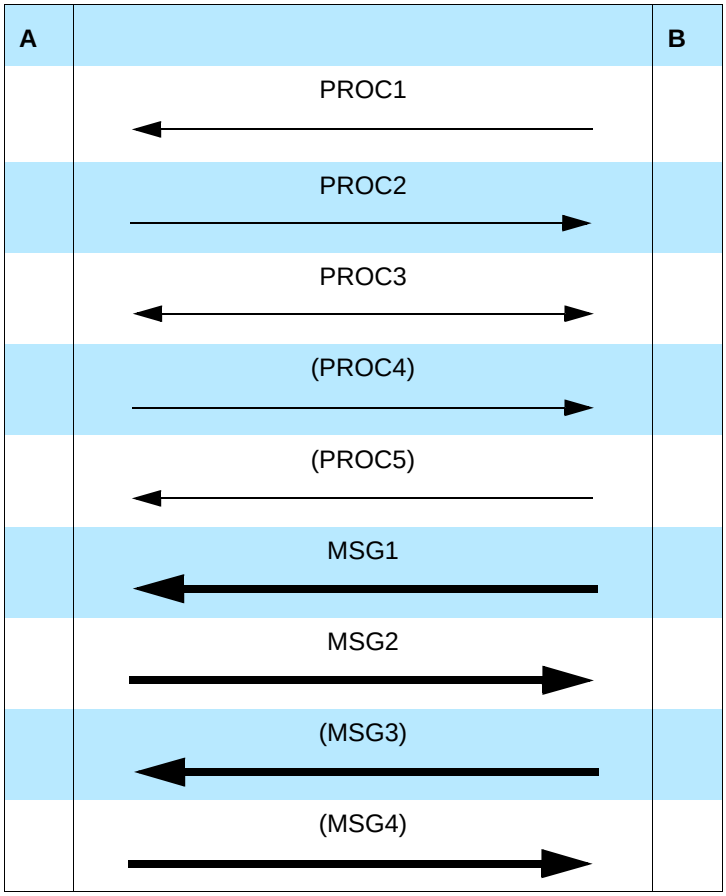


Table 1.1: Arrows used in signaling diagrams

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

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

2 PROFILE OVERVIEW

2.1 PROFILE STACK

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

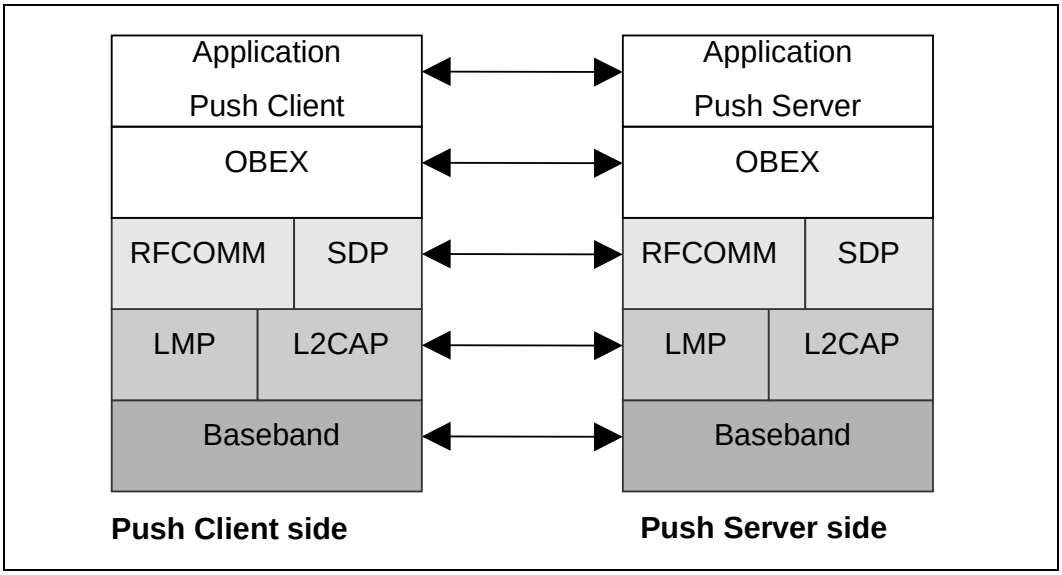


Figure 2.1: Protocol model

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

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

2.2 CONFIGURATIONS AND ROLES

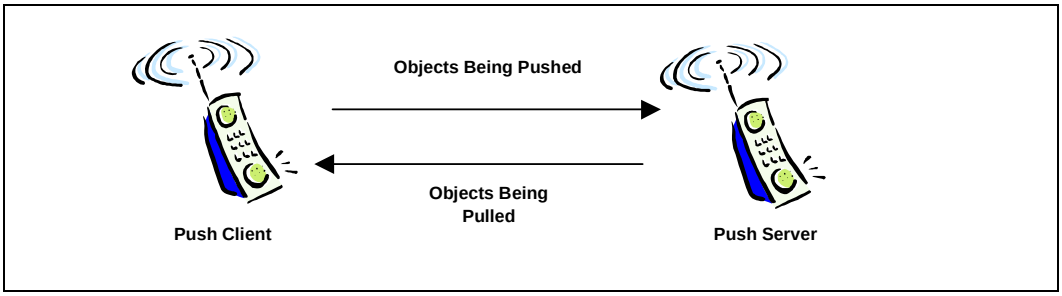


Figure 2.2: Push and Pull Example between two Mobile Phones

The following roles are defined for this profile:

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

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

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are:

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

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

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

2.4 PROFILE FUNDAMENTALS

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

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

Bonding is mandatory to support and optional to use.

OBEX authentication is not used.

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

3 USER INTERFACE ASPECTS

3.1 MODE SELECTION, PUSH SERVERS

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

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

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

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

3.2 FUNCTION SELECTION, PUSH CLIENTS

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

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

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

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

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

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

3.3 APPLICATION USAGE EVENTS

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

3.3.1 Object Push

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

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

3.3.2 Business Card Pull

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

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

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

3.3.3 Business Card Exchange

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

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

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

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

4 APPLICATION LAYER

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

4.1 FEATURE OVERVIEW

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

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

Table 4.1: Application layer features

*. Optional, but the server must be able to respond with an error code on a pull request, even if it doesn't support this feature

4.2 OBJECT PUSH FEATURE

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

4.2.1 Content Formats

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

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

mat as specified in Chapter 9 of [9]. If a device does not have a messaging application it does not have to support the vMessage content format.

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

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

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

4.2.2 Application Procedure

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

Table 4.2 shows the application procedure required by the Push Client to push one or more objects to a Push Server.

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

Table 4.2: Application layer procedure for Object Push

For a detailed description of OBEX operations see Section 5.

4.3 BUSINESS CARD PULL FEATURE

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

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

4.3.1 Owner's Business Card

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

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

See [8] for a discussion on the Default Get Object.

4.3.2 Application Procedure Business Card Pull

Table 4.3 shows the application procedure required by the Push Client to perform a Business Card Pull from a Push Server.

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

Table 4.3: Application layer procedure for Business Card Pull

For a detailed description of OBEX operations see Section 5.

4.4 BUSINESS CARD EXCHANGE FEATURE

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

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

4.4.1 Owner's Business Card

See Business Card Pull feature.

4.4.2 Application Procedure Business Card Exchange

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

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

Table 4.4: Application layer procedure for Business Card Exchange

For a detailed description of OBEX operations see Section 5.

5 OBEX

5.1 OBEX OPERATIONS USED

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

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

Table 5.1: OBEX Operations

5.2 OBEX HEADERS

5.2.1 OBEX Headers for the Object Push Feature

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

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

Table 5.2: OBEX Headers used for the Object Push feature

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

Table 5.2: OBEX Headers used for the Object Push feature

5.2.2 OBEX Headers for the Business Card Pull and Exchange Features

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

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

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

5.3 INITIALIZATION OF OBEX

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

5.4 ESTABLISHMENT OF OBEX SESSION

See Section 5.4.1, in GOEP for a description of OBEX connection establishment without authentication.

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

5.5 PUSHING DATA

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

See Section 5.5 in GOEP.

5.6 PULLING DATA

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

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

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

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

See Section 5.6 in GOEP.

5.7 DISCONNECTION

See Section 5.7 in GOEP.

6 SERVICE DISCOVERY

6.1 SD SERVICE RECORDS

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

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

Table 6.1: Object Push Service Record

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

6.2 SDP PROTOCOL DATA UNITS

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

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

Table 6.2: SDP PDUs

7 REFERENCES

7.1 NORMATIVE REFERENCES

[1] Bluetooth Special Interest Group, Baseband Specification

[2] Bluetooth Special Interest Group, LMP Specification

[3] Bluetooth Special Interest Group, L2CAP Specification

[4] Bluetooth Special Interest Group, RFCOMM with TS 07.10

[5] ETSI, TS 07.10, Version 6.3

[6] Bluetooth Special Interest Group, SDP Specification

[7] Bluetooth Special Interest Group, IrDA Interoperability

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

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

[10] Bluetooth Special Interest Group, Generic Object Exchange Profile

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

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

[13] Bluetooth Special Interest Group, Generic Access Profile Specification

[14] Bluetooth Special Interest Group, File Transfer Profile Specification

[15] Bluetooth Special Interest Group, Synchronization Profile Specification

[16] Bluetooth Special Interest Group, Assigned Numbers specification

Part K:12

FILE TRANSFER PROFILE

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

CONTENTS

1 Introduction 360
1.1 Scope 360
1.2 bluetooth profile structure 360
1.3 Bluetooth OBEX-Related Specifications 361
1.4 Symbols and conventions 362
1.4.1 Requirement status symbols 362
1.4.2 Signaling diagram conventions 363

2 Profile overview 364
2.1 Profile stack 364
2.2 Configurations and roles 364
2.3 User requirements and scenarios 365
2.4 Profile fundamentals 365

3 User interface aspects 367
3.1 File Transfer Mode selection, Servers 367
3.2 Function Selection, Clients 367
3.3 Application usage 368

4 Application layer 370
4.1 Feature overview 370
4.2 Folder Browsing 370
4.3 Object Transfer 372
4.4 Object Manipulation 373

5 OBEX 374
5.1 OBEX Operations Used 374
5.2 OBEX Headers 375
5.3 Initialization of OBEX 375
5.4 Establishment of OBEX session 375
5.5 Browsing Folders 376
5.5.1 Pulling a Folder Listing Object 376
5.5.2 Setting the Current Folder (Forward) 376
5.5.3 Setting the Current Folder (Backward) 377
5.5.4 Setting the Current Folder (Root) 378
5.6 Pushing Objects 379
5.6.1 Pushing Files 379
5.6.2 Pushing Folders 379
5.6.2.1 Creating New Folders 380
5.7 Pulling Objects 381
5.7.1 Pulling Files 381
5.7.2 Pulling Folders 381

5.8 Manipulating Objects 381
5.8.1 Deleting Files 381
5.8.2 Deleting Folders 382
5.9 Disconnection 382

6 Service Discovery 383
6.1 SD service records 383
6.2 SDP Protocol Data Units 384

7 References 385
7.1 Normative references 385

FOREWORD

This document, together with the Generic Object Exchange profile and the Generic Access profile form the File Transfer usage model.

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

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

1 INTRODUCTION

1.1 SCOPE

The File Transfer profile defines the requirements for the protocols and procedures that shall be used by the applications providing the File Transfer usage model. This profile uses the Generic Object Exchange profile (GOEP) as a base profile to define the interoperability requirements for the protocols needed by the applications. The most common devices using these usage models can be (but are not limited to) PCs, notebooks, and PDAs.

The scenarios covered by this profile are the following:

- Usage of a Bluetooth device (e.g. a notebook PC) to browse an object store (file system) of another Bluetooth device. Browsing involves viewing objects (files and folders) and navigating the folder hierarchy of another Bluetooth device. For example, one PC browsing the file system of another PC.
- A second usage is to transfer objects (files and folders) between two Bluetooth devices. For example, copying files from one PC to another PC.
- A third usage is for a Bluetooth device to manipulate objects (files and folders) on another Bluetooth device. This includes deleting objects, and creating new folders.

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.

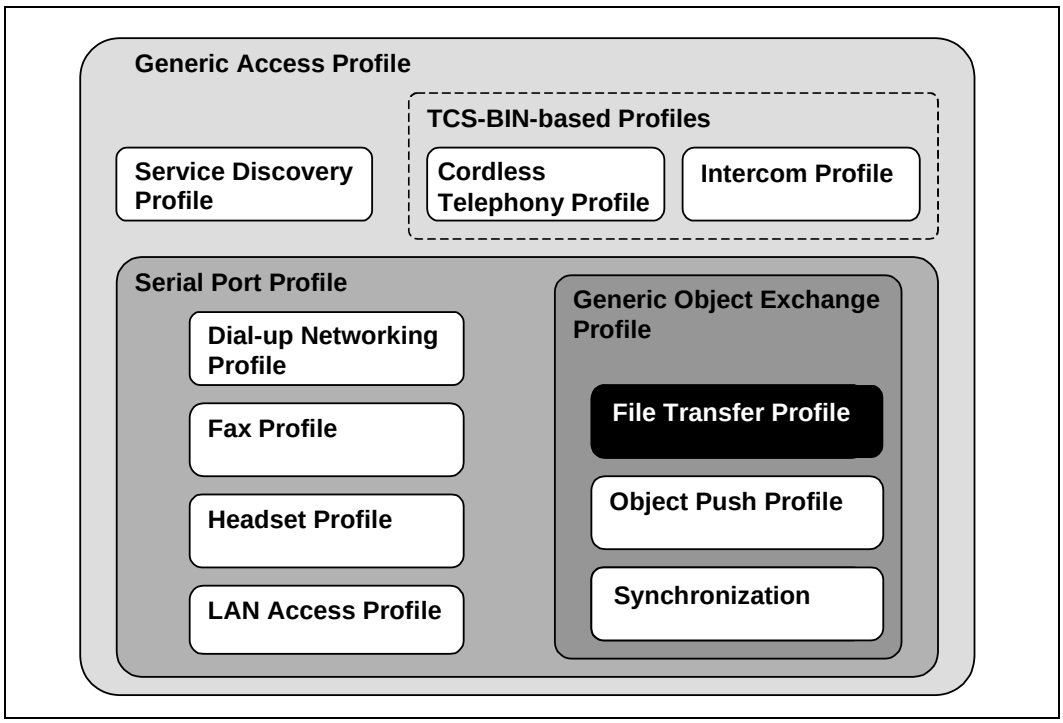


Figure 1.1: Bluetooth Profiles

1.3 BLUETOOTH OBEX-RELATED SPECIFICATIONS

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

1. Bluetooth IrDA Interoperability Specification [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 [2]

- 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 [3]

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

4. Bluetooth File Transfer Profile Specification (This Specification)

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

5. Bluetooth Object Push Profile Specification [4]

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

1.4 SYMBOLS AND CONVENTIONS

1.4.1 Requirement status symbols

In this document (especially in the profile requirements tables in Annex A), 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:

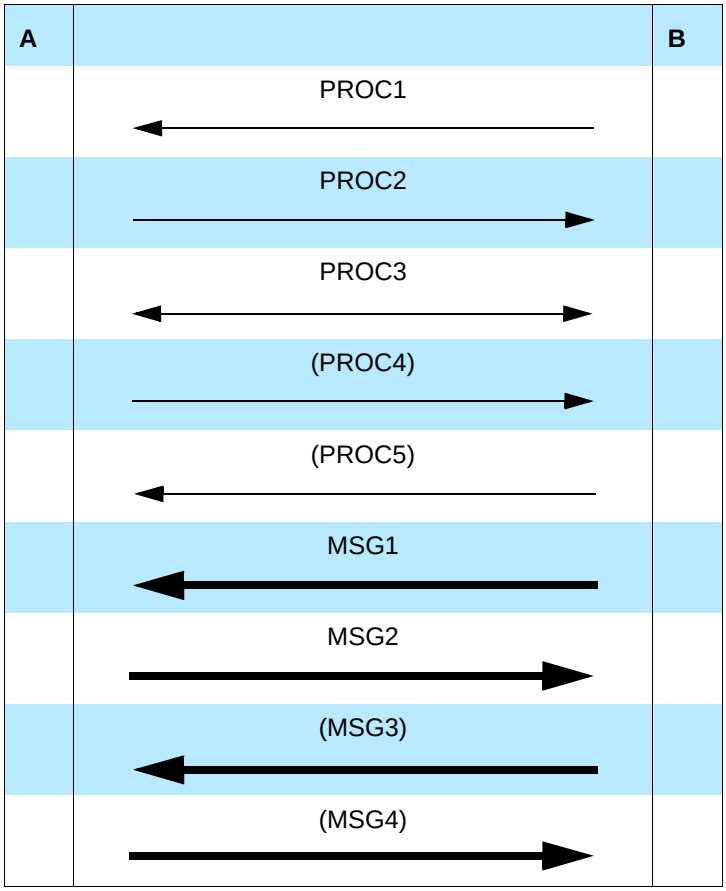


Table 1.1: Arrows used in signaling diagrams

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.

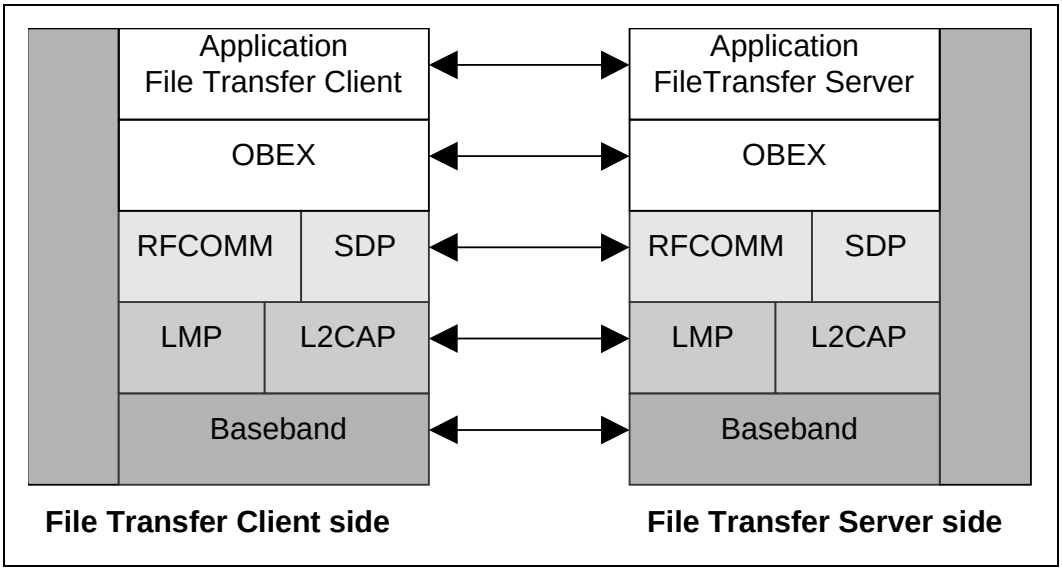


Figure 2.1: Protocol model

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 RFCOMM, L2CAP, LMP, and Baseband interoperability requirements are defined in GOEP.

2.2 CONFIGURATIONS AND ROLES

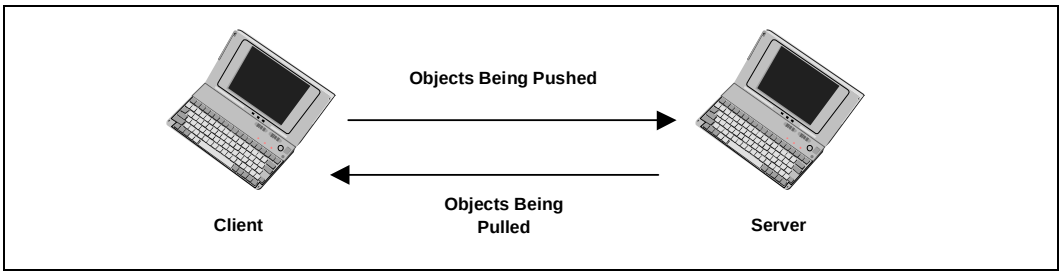


Figure 2.2: Bi-directional File Transfer Example between two Personal Computers

The following roles are defined for this profile:

Client – The Client device initiates the operation, which pushes and pulls objects to and from the *Server*. In addition to the interoperability requirements defined in this profile, the Client must also comply with the interoperability requirements for the Client of the GOEP if not defined in the contrary. The Client must be able to interpret the OBEX Folder Listing format and may display this information for the user.

Server – The Server device is the target remote Bluetooth device that provides an object exchange server and folder browsing capability using the OBEX Folder Listing format. In addition to the interoperability requirements defined in this profile, the Server must comply with the interoperability requirements for the Server of the GOEP if not defined in the contrary.

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are the following:

- Usage of the Client to browse the object store of the Server. Clients are required to pull and understand Folder Listing Objects. Servers are required to respond to requests for Folder Listing Objects. Servers are required to have a root folder. Servers are not required to have a folder hierarchy below the root folder.
- Usage of the Client to transfer objects to and from the Server. The transfer of objects includes folders and files. Clients must support the ability to push or pull files from the Server. Clients are not required to push or pull folders. Servers are required to support file push, pull, or both. Servers are allowed to have read-only folders and files, which means they can restrict object pushes. Thus, Servers are not required to support folder push or pull.
- Usage of the Client to create folders and delete objects (folders and files) on the Server. Clients are not required to support folder/file deletion or folder creation. Servers are allowed to support read-only folders and files, which means they can restrict folder/file deletion and creation.

A device adhering to this profile must support Client capability, Server capability or both. The restrictions applying to this profile are the same as in the GOEP.

2.4 PROFILE FUNDAMENTALS

The profile fundamentals are the same as defined in Section 2.4 in GOEP [2]. Support for link level authentication and encryption is required but their use is optional.

Support for OBEX authentication is required but its use is optional.

This profile does not mandate the server or client to enter any discoverable or

connectable modes automatically, even if they are able to do so.

On the Client side, end-user intervention is always needed to initiate file transfer (see Chapter 3).

Support of bonding is required but its use is optional.

3 USER INTERFACE ASPECTS

3.1 FILE TRANSFER MODE SELECTION, SERVERS

Servers must be placed in File Transfer mode. This mode enables a Client to perform file transfer operations with the Server. When entering this mode, File Transfer Servers should set the device in *Limited Discoverable* mode (see Generic Access Profile), ensure that the Object Transfer Bit is set in the CoD (see [15]), and register a service record in the SDDB (see section 6 on page 383).

It is recommended that this mode be set and unset by user interaction, when possible. Public devices, devices that want to be visible at all times, or devices that can not supply a user interface to enable File Transfer mode shall use *General Discoverable* mode (see Generic Access Profile) instead of *Limited Discoverable* mode.

3.2 FUNCTION SELECTION, CLIENTS

Clients provide file transfer functions to the user via a user interface. An example of a file transfer user interface is a file-tree viewer to browse folders and files. Using such a system file-tree viewer, the user can browse and manipulate files on another PC, which appears in the network view.

File Transfer Applications provide the following functions.

Select Server	Selecting the Server from a list of possible Servers, and setting up a connection to it.
Navigate Folders	Displaying the Server's folder hierarchy, including the files in the folders, and moving through the Server's folder hierarchy to select the current folder. The current folder is where items are pulled and/or pushed.
Pull Object	Copying a file or a folder from the Server to the Client.
Push Object	Copying a file or folder from the Client to the Server.
Delete Object	Deleting a file or folder on the Server.
Create Folder	Creating a new folder on the Server.

When the user selects the Select Server function, an inquiry procedure will be performed to produce a list of available devices in the vicinity. Requirements on inquiry procedures are discussed in Section 6.5.1 of the GOEP [2].

3.3 APPLICATION USAGE

In this section, the presented scenarios work as examples. Variations in the actual implementations are possible and allowed.

When the Client wants to select a Server the following user interaction can be followed:

Client	Server
	The user sets the device **into File Transfer mode**. A Server typically does not need to provide any other user interaction.
The user of the Client selects the **File Transfer Application** on the device.	
A list of Servers that may support the File Transfer service is displayed to the user.	
The user selects a Server in which to connect. The connection may require the user to enter a password for authentication. If both link level authentication and OBEX authentication is required, then the user will need to be prompted for two passwords.	If the Client requires authentication of the Server, then the Server will need to prompt the user for a password. If both link level authentication and OBEX authentication are required, then the user will need to be prompted for two passwords.
After the connection is complete, including any authentication, the contents of the Server's root folder are displayed.	

The following user interaction shows how the user of the Client performs file transfer functions. The operations assume a Server has already been selected as described above.

Client	**Server**
The user is presented with the folder hierarchy of the Server. The first presentation has the root folder selected as the current folder.	
The user chooses a folder to be the current folder. The contents of this folder are displayed.	
To push a file from the Client to the Server, the user selects a file on the Client and activates the **Push Object** function. The object is transferred to the current folder on the Server.	
To pull a file from the Server, the user selects a file in the current folder of the Server and activates the **Pull Object** function. The user is notified of the result of the operation.	
To delete a file on the Server, the user selects the file in the Server's current folder and activates the **Delete Object** function. The user is notified of the result of the operation.	
To create a new folder on the Server, the user activates the **Create Folder** function. This function requests a name from the user for the folder. When complete, a new folder is created in the Server's current folder.	

4 APPLICATION LAYER

This section describes the feature requirements on units active in the File Transfer use case.

4.1 FEATURE OVERVIEW

The File Transfer application is divided into three main features, as shown in the Table 4.1 below.

	Features	**Support in File Transfer Client**	**Support in File Transfer Server**
1.	Folder Browsing	M	M
2.	Object Transfer: File Transfer Folder Transfer	 M O	 M O*
3.	Object Manipulation	O	O*

Table 4.1: Application layer procedures

*. Optional, but the server must be able to respond with an appropriate error code, even if it doesn't support these capabilities.

4.2 FOLDER BROWSING

A file transfer session begins with the Client connecting to the Server and pulling the contents of the Server's root folder. When an OBEX connection is made, the Server starts out with its current folder set to the root folder. The contents of folders must be transferred in the Folder Listing format specified in [11].

Table 4.2 shows the application procedure required by the Client to connect to the Server and pull the contents of the root folder.

Client	Details
OBEX CONNECT.	Target Header must be set to the Folder Browsing UUID: F9EC7BC4-953C-11D2-984E-525400DC9E09. This UUID is sent in binary (16 bytes) with most significant byte sent first (0xF9 is sent first).
Pull the contents of the Server's root folder using GET.	The Type Header must be set to the MIME-type of the Folder Listing Object (x-obex/folder-listing). The Connect ID header must be set to the value returned in the Connect operation. A Name header is not used.

Table 4.2: Application layer procedure for File Transfer Connect

Browsing an object store involves displaying folder contents and setting the 'current folder'. The OBEX SETPATH command is used to set the current folder. To display a folder hierarchy starting with the root folder, the Client must read the root folder contents using GET. It must then retrieve the contents of all sub-folders using GET. If the sub-folders contain folders, then the Client must retrieve the contents of these folders and so on. To retrieve the contents of a folder, the Client must set the current folder to the sub-folder using SETPATH, then pull the sub-folder contents using GET. Table 4.3 shows the application procedure required for retrieving the contents of a sub-folder.

Client	Details
Set the current folder to the sub-folder using OBEX SETPATH.	Name header is set to the name of the sub-folder. Connect ID header is required.
Pull the contents of the sub-folder using GET.	No Name is sent, since the sub-folder is the current folder. The Type Header must be set to the MIME-type of the Folder Listing Object (x-obex/folder-listing). Connect ID header is required.
Set the current folder back to the root folder using OBEX SETPATH.	Name header is empty. Connect ID header is required.
If the parent of the sub-folder is not the root folder, then set the current folder to the parent folder using SETPATH.	The Backup flag is set and no Name header is sent. Connect ID header is required.

Table 4.3: Application layer procedure for Folder Browsing

4.3 OBJECT TRANSFER

Objects are transferred from the Client to the Server using OBEX PUT, and objects are transferred from the Server to the Client using OBEX GET. Transferring files requires a single PUT or GET operation per file. Transferring folders requires transferring all the items stored in a folder, including other folders. The process of transferring a folder may require that new folders be created. The SETPATH command is used to create folders.

Table 4.4 shows the application procedure for transferring a folder from the Client to the Server. If the folder contains other folders, then these other folders are transferred using the same method. The folder is transferred to the current folder on the Server.

Client	Details
Create a new folder (if it does not already exist) in the Server's current folder using SETPATH. The current folder is changed to this new folder.	Name header is set to the name of the new folder. Connect ID header is required.
Push all files to the new folder using a PUT command for each file.	The Name header is set to the name of the file. Connect ID header is required.
Folders are created using SETPATH.	Name header is set to folder name. This application procedure is applied recursively to each folder. Connect ID header is required.
Set the current folder back to the parent folder using SETPATH.	The Backup flag is set and no Name header is sent. Connect ID header is required.

Table 4.4: Application layer procedure for Pushing a Folder

Table 4.5 shows the application procedure for transferring a folder from the Server to the Client.

Client	Details
Set the current folder to the folder which is to be transferred using SETPATH.	The Name header is set to the name of the folder. Connect ID header is required.
Pull the contents of the folder using GET.	A Name header is not sent, and the Type Header must be set to the MIME-type of the Folder Listing Object (x-obex/folder-listing).
Pull all files to the new folder using a GET command for each file.	The Name header is set to the name of the file. Connect ID header is required.
Pull all Folders to the new folder using this application procedure.	This application procedure is applied recursively to each folder.
Set the current folder back to the parent folder, using SETPATH.	The Backup flag is set and no Name header is sent. Connect ID header is required.

Table 4.5: Application layer procedure for Pulling a Folder

4.4 OBJECT MANIPULATION

A Client can delete folders and files on a Server. It can also create new folders on a Server. A brief summary of these functions is shown below.

- A file is deleted by using a PUT command with the name of the file in a Name header and no Body header.
- An empty folder is deleted by using a PUT command with the name of the folder in a Name header and no Body header.
- A non-empty folder can be deleted in the same way as an empty folder but Servers may not allow this operation. If a Server refuses to delete a non-empty folder it must return the “Precondition Failed” (0xCC) response code. This response code tells the Client that it must first delete all the elements of the folder individually before deleting the folder.
- A new folder is created in the Server’s current folder by using the SETPATH command with the name of the folder in a Name header. If a folder with that name already exists, then a new folder is not created. In both cases the current folder is set to the new folder.

5 OBEX

5.1 OBEX OPERATIONS USED

Table 5.1 shows the OBEX operations, which are required in the File Transfer profile.

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

Table 5.1: OBEX Operations

5.2 OBEX HEADERS

Table 5.2 shows the specified OBEX headers, which are required in the File Transfer profile.

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

Table 5.2: OBEX Headers

5.3 INITIALIZATION OF OBEX

Devices implementing the File Transfer profile can optionally use OBEX authentication. The initialization procedure is defined in Section 5.3 of GOEP [2].

5.4 ESTABLISHMENT OF OBEX SESSION

The OBEX connection must use a Target header set to the File Browsing UUID, F9EC7BC4-953C-11D2-984E-525400DC9E09. This UUID is sent in binary (16 bytes) with 0xF9 sent first. OBEX authentication can optionally be used. This profile follows the procedures described in Section 5.4 of GOEP [2] with the Target, Connection ID, and Who headers being mandatory.

5.5 BROWSING FOLDERS

Browsing folders involves pulling Folder Listing objects and setting the current folder. Navigating a folder hierarchy requires moving forward and backward by changing the current folder. Upon completion of the OBEX Connect operation the Server's current folder is the root folder.

5.5.1 Pulling a Folder Listing Object

Pulling a Folder Listing object uses a GET operation and follows the procedure described in Section 5.6 of GOEP [2]. The Connection ID and Type headers are mandatory. A Name header containing the name of the folder is used to pull the listing of a folder. Sending the GET command without a name header is used to pull the contents of the current folder. Typically, a folder browsing application will pull the contents of the current folder, so a Name header is not used. The Type header must be set to 'x-obex/folder-listing'.

5.5.2 Setting the Current Folder (Forward)

Setting the current folder requires the SETPATH operation. The SETPATH request must include the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for SETPATH	0x82	M	-
Field	Packet Length	Varies	M	-
Field	Flags	0x02	M	'Backup level' flag is set to 0 and 'Don't Create' flag is set to 1.
Field	Constants	0x00	M	Constants are not used, and the field must be set to 0.
Header	Connection ID	Varies	M	Connection ID is set to the value returned by the Server during the OBEX Connect operation. This must be the first header.
Header	Name	Varies	M	Name of the folder.

Table 5.3: Fields and Headers in SETPATH Request for Setting Current Folder (Forward)

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

Field/ Header	Name	Value	Status	Explanation
Field	Response code for SETPATH	0xA0 or 0xC4	M	0xA0 for success or 0xC4 if the folder does not exist.
Field	Packet Length	Varies	M	-

Table 5.4: Fields and Headers in SETPATH Response for Setting Current Folder (Forward)

Other headers such as Description can optionally be used.

5.5.3 Setting the Current Folder (Backward)

Setting the current folder back to the parent folder requires the SETPATH operation. The SETPATH request must include the following fields and headers (note that a Name header is not used):

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for SET-PATH	0x82	M	-
Field	Packet Length	Varies	M	-
Field	Flags	0x03	M	‘Backup level’ flag is set to 1 and ‘Don’t Create’ flag is set to 1.
Field	Constants	0x00	M	Constants are not used, and the field must be set to 0.
Header	Connection ID	Varies	M	Connection ID is set to the value returned by the Server during the OBEX Connect operation. This must be the first header.

Table 5.5: Fields and Headers in SETPATH Request for Setting Current Folder (Backward)

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

Field/ Header	Name	Value	Status	Explanation
Field	Response code for SETPATH	0xA0 or 0xC4	M	0xA0 for success, or 0xC4 if the current folder is the root.
Field	Packet Length	Varies	M	-

Table 5.6: Fields and Headers in SETPATH Response for Setting Current Folder (Backward)

Other headers, such as Description, can optionally be used.

5.5.4 Setting the Current Folder (Root)

Setting the current folder to the root requires the SETPATH operation. The SETPATH request must include the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for SET-PATH	0x82	M	-
Field	Packet Length	Varies	M	-
Field	Flags	0x02	M	‘Backup level’ flag is set to 0 and ‘Don’t Create’ flag is set to 1.
Field	Constants	0x00	M	Constants are not used, and the field must be set to 0.
Header	Connection ID	Varies	M	Connection ID is set to the value returned by the Server during the OBEX Connect operation. This must be the first header.
Header	Name	Empty	M	Name header is empty.

Table 5.7: Fields and Headers in SETPATH Request for Setting Current Folder (Root)

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

Field/ Header	Name	Value	Status	Explanation
Field	Response code for SETPATH	0xA0	M	0xA0 for success.
Field	Packet Length	Varies	M	-

Table 5.8: Fields and Headers in SETPATH Response for Setting Current Folder (Root)

Other headers, such as Description, can optionally be used.

5.6 PUSHING OBJECTS

Pushing object involves pushing files and folders.

5.6.1 Pushing Files

Pushing files follows the procedure described in Section 5.5 of GOEP [2]. The Connection ID header is mandatory.

5.6.2 Pushing Folders

Pushing folders involves creating new folders and pushing files. It may also involve navigating through the folder hierarchy. Navigation is described in Section 5.5 on page 376. Pushing files is described in Section 5.6.1 on page 379.

5.6.2.1 Creating New Folders

Creating a new folder requires the SETPATH operation. The SETPATH request must include the following fields and headers:

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for SET-PATH	0x82	M	-
Field	Packet Length	Varies	M	-
Field	Flags	0x00	M	'Backup level' flag is set to 0 and 'Don't Create' flag is set to 0.
Field	Constants	0x00	M	Constants are not used, and the field must be set to 0.
Header	Connection ID	Varies	M	Connection ID is set to the value returned by the Server during the OBEX Connect operation. This must be the first header.
Header	Name	Varies	M	Name of the folder.

Table 5.9: Fields and Headers in SETPATH Request for Creating a Folder.

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

Field/ Header	Name	Value	Status	Explanation
Field	Response code for SETPATH	0xA0 or 0xC1	M	0xA0 for success or 0xC1 if the current folder is read only and creation of a new folder is unauthorized.
Field	Packet Length	Varies	M	-

Table 5.10: Fields and Headers in SETPATH Response for Creating a Folder

Other headers such as Description can optionally be used.

5.7 PULLING OBJECTS

Pulling objects involves pulling files and folders.

5.7.1 Pulling Files

Pulling files follows the procedure described in Section 5.6 of GOEP [2]. The Connect ID header is mandatory.

5.7.2 Pulling Folders

Pulling folders involves navigating the folder hierarchy, pulling folder listing objects and pulling files. Navigating the folder hierarchy and pulling folder listing-objects is described in Section 5.5 on page 376. Pulling files is described in Section 5.7.1 on page 381.

5.8 MANIPULATING OBJECTS

Manipulating objects includes deleting objects and creating new folders. Creating new folders is described in Section 5.6.2.1 on page 380, Creating New Folders. Deleting objects involves deleting files and folders.

5.8.1 Deleting Files

Deleting a file requires the PUT operation. The PUT request must include the following fields and headers (note that no Body or End Body headers are sent):

Field/ Header	Name	Value	Status	Explanation
Field	Opcode for PUT	0x82	M	-
Field	Packet Length	Varies	M	-
Header	ConnectionID	Varies	M	Connection ID is set to the value returned by the Server during the OBEX Connect operation. This must be the first header.
Header	Name	Varies	M	The header value is the name of the object to delete.

Table 5.11: Fields and Headers in PUT Request for Delete

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

Field/ Header	Name	Value	Status	Explanation
Field	Response code for PUT	0xA0, 0xC1 or 0xC4	M	0xA0 for success, 0xC1 for unauthorized (e.g. read only) or 0xC4 if the file does not exist.
Field	Packet Length	Varies	M	-

Table 5.12: Fields and Headers in PUT Response for Delete

Other headers such as Description can optionally be used.

5.8.2 Deleting Folders

A folder can be deleted using the same procedure used to delete a file (see Section 5.8.1 on page 381). Deleting a non-empty folder will delete all its contents, including other folders. Some Servers may not allow this operation and will return the “Precondition Failed” (0xCC) response code, indicating that the folder is not empty. In this case the Client will need to delete the contents before deleting the folder.

5.9 DISCONNECTION

See Section 5.7 in GOEP [2].

6 SERVICE DISCOVERY

6.1 SD SERVICE RECORDS

The service belonging to the File Transfer profile is a server, which enables bi-directional generic file transfer. OBEX is used as a session protocol for this service. The following service records must be put into the SDDB.

Item	Definition:	Type/ Size:	Value:*	AttrID	Status	Default Value
Service Class ID List				See [15]	M	
Service Class #0		UUID	OBEX-File Transfer		M	
Protocol Descriptor list				See [15]	M	
Protocol ID #0		UUID	L2CAP		M	
Protocol ID #1		UUID	RFCOMM		M	
Param #0	CHANNEL	Uint8	Varies		M	
Protocol ID #2		UUID	OBEX		M	
Service name	Display-able Text name	String	Varies	See [15]	O	“OBEX File Trans-fer”
BluetoothProfileDe-scriptorList				See [15]	O	
Profile ID #0	Supported profile	UUID	OBEX File-Transfer			OBEX File Transfer [15]
Param #0	Profile ver-sion	uint16	0x100			0x100

Table 6.1: File Transfer Service Record

* UUID values are defined in the *Assigned Numbers* document.

6.2 SDP PROTOCOL DATA UNITS

Table 19 shows the specified SDP PDUs (Protocol Data Units) which are required in the File Transfer profile.

PDU no.	SDP PDU	Server	Client
1	SdpErrorResponse	M	M
2	SdpServiceSearch AttributeRequest	M	M
3	SdpServiceSearch AttributeResponse	M	M

Table 6.2: SDP PDUs Minimal Requirements

7 REFERENCES

7.1 NORMATIVE REFERENCES

[1] Bluetooth Special Interest Group, IrDA Interoperability

[2] Bluetooth Special Interest Group, Generic Object Exchange Profile

[3] Bluetooth Special Interest Group, Synchronization 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

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

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

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

[15] Bluetooth Special Interest Group, Assigned Numbers specification

[16] Bluetooth Special Interest Group, Bluetooth Generic Access Profile Specification

Part K:13

SYNCHRONIZATION PROFILE

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

CONTENTS

1 Introduction 392
1.1 Scope 392
1.2 Bluetooth Profile Structure 392
1.3 Bluetooth OBEX Related Specifications 393
1.4 Symbols and conventions 394
1.4.1 Requirement status symbols 394
1.4.2 Signaling diagram conventions 395

2 Profile overview 396
2.1 Profile stack 396
2.2 Configurations and roles 396
2.3 User requirements and scenarios 397
2.4 Profile fundamentals 398

3 User interface aspects 399
3.1 Mode selection 399
3.2 Application Usage Events 399
3.2.1 Synchronization Scenario 400
3.2.2 Sync Command Scenario 401
3.2.3 Automatic Synchronization Scenario 401

4 Application layer 402
4.1 Feature overview 402
4.2 Synchronization Feature 402
4.3 Sync Command Feature 403
4.4 Automatic Synchronization Feature 403

5 IrMC Synchronization Requirements 404

6 OBEX 406
6.1 OBEX Operations Used 406
6.2 OBEX Headers 406
6.3 Initialization of OBEX 407
6.4 Establishment of OBEX session 407
6.5 Pushing Data 407
6.6 Pulling Data 407
6.7 Disconnection 407

7 Service Discovery 408

7.1 SD Service Records 408

7.1.1 Synchronization Service 408

7.1.2 Sync Command Service 409

7.2 SDP Protocol Data Units 410

8 References 411

8.1 Normative references 411

FOREWORD

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

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

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

1 INTRODUCTION

1.1 SCOPE

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

The scenarios covered by this profile are:

- Usage of a mobile phone or PDA by a computer to exchange PIM (Personal Information Management) data, including a necessary log information to ensure that the data contained within their respective Object Stores is made identical. Types of the PIM data are, for example, phonebook and calendar items.
- Use of a computer by a mobile phone or PDA to initiate the previous scenario (Sync Command Feature).
- Use of a mobile phone or PDA by a computer to automatically start synchronization when a mobile phone or PDA enters the RF proximity of the computer

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.

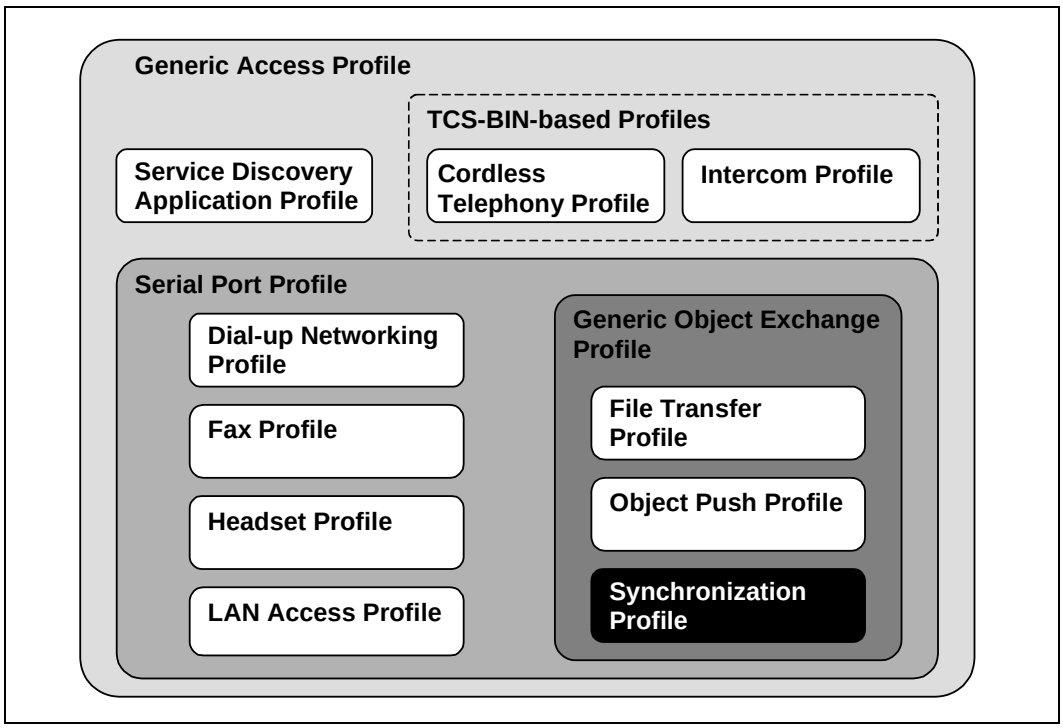


Figure 1.1: Bluetooth Profiles

1.3 BLUETOOTH OBEX RELATED SPECIFICATIONS

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

1. Bluetooth IrDA Interoperability Specification [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 [2]

- 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 (This Specification)

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

4. Bluetooth File Transfer Profile Specification [3]

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

5. Bluetooth Object Push Profile Specification [4]

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

1.4 SYMBOLS AND CONVENTIONS

1.4.1 Requirement status symbols

In this document, the following symbols are used:

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

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

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

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

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

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

1.4.2 Signaling diagram conventions

The following arrows are used in diagrams describing procedures:

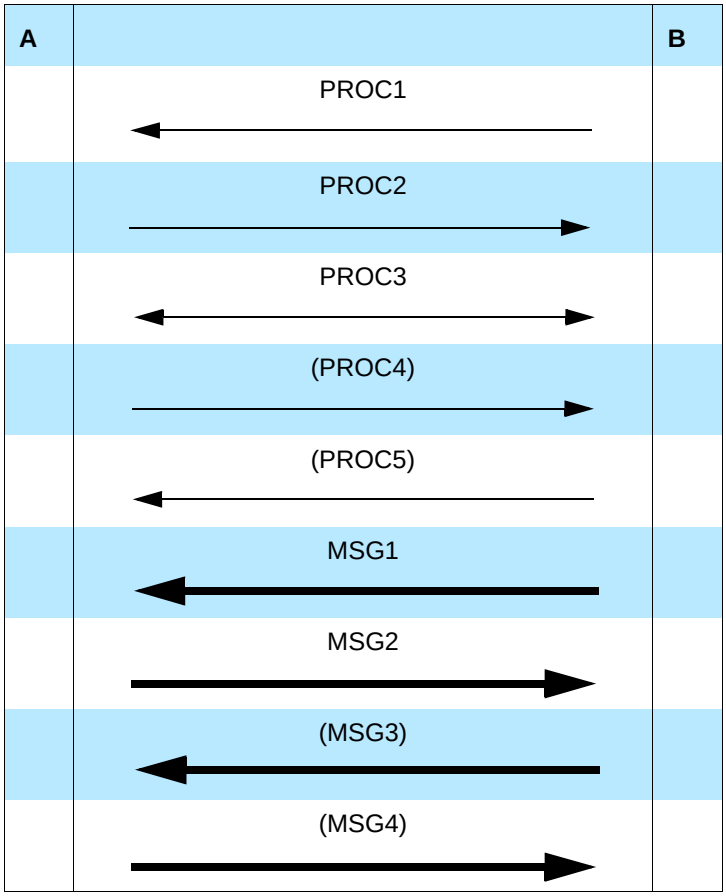


Table 1.1: Arrows used in signaling diagrams

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

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

2 PROFILE OVERVIEW

2.1 PROFILE STACK

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

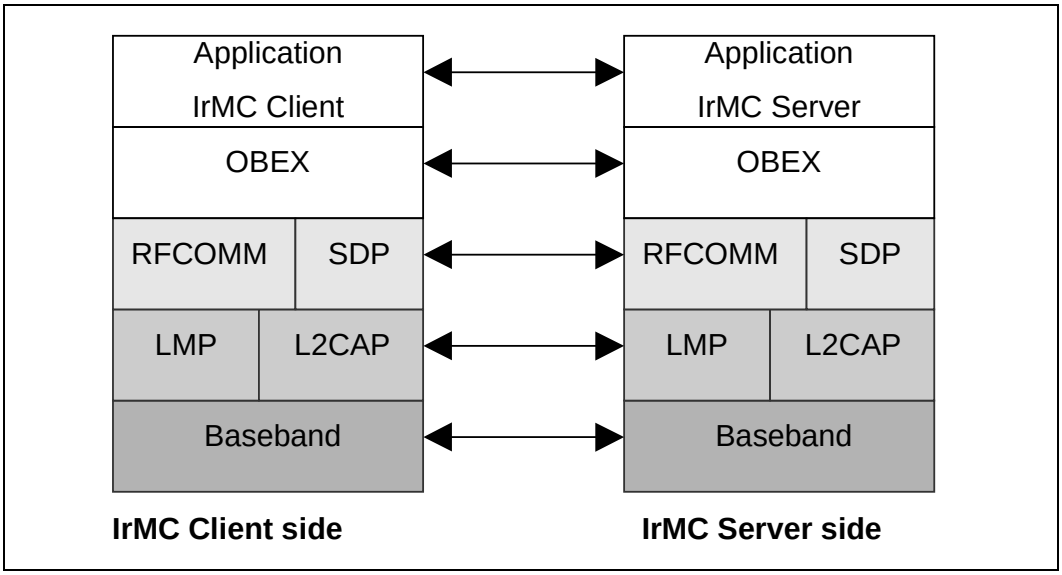


Figure 2.1: Protocol model

The Baseband [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 IrMC Client layer shown in Figure 2.1 is the entity processing the synchronization according to the IrMC specification [12], and the IrMC server is the server software compliant to the IrMC specification.

The RFCOMM, L2CAP, LMP, and Baseband interoperability requirements are defined in Section 6 in GOEP[2].

2.2 CONFIGURATIONS AND ROLES

Figure 2.2 depicts a synchronization example in which a mobile phone acts as an IrMC server and a PC notebook as an IrMC Client. The IrMC Client (PC) pulls the PIM data from the IrMC server and synchronizes this data with data stored in the IrMC client. After that, the IrMC client puts this synchronized data back to the IrMC server.

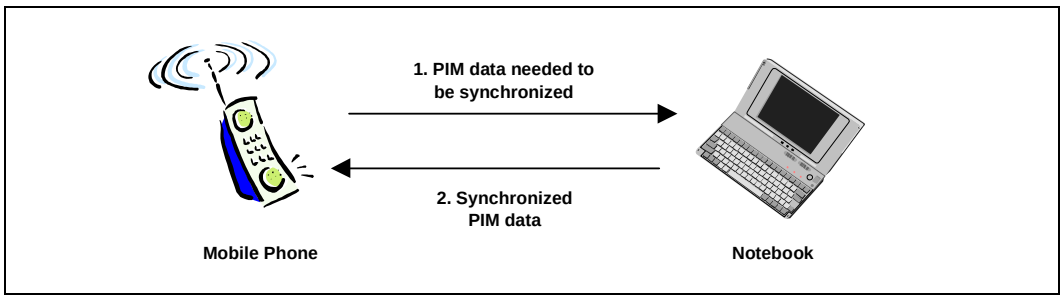


Figure 2.2: Synchronization Example with Mobile Phone and Computer

The following roles are defined for this profile:

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

If the IrMC Server also provides the functionality to initiate the synchronization, then it must act as a client temporarily. In this case, it must also comply with the requirements with the client of the GOEP if not defined in the contrary.

IrMC Client – This is the IrMC client device, which contains a sync engine and pulls and pushes the PIM data from and to the IrMC Server. Usually, the IrMC Client device is a PC. Because the IrMC Client must also provide functionality to receive the initialization command for synchronization, sometimes it must temporarily act as a server. In addition to the interoperability requirements defined in this profile, the IrMC server must also comply with the interoperability requirements for the server and client of the GOEP if not defined to the contrary.

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are:

- Usage of an IrMC Server by an IrMC Client to pull the PIM data needed to be synchronized from the IrMC Server, to synchronize this data with the data on the IrMC Client, and to push this synchronized data back to the IrMC Server.
- Usage of an IrMC Client by an IrMC Server to initiate the previous scenario by sending a sync command to the IrMC Client.
- Automatic synchronization initiated by the IrMC client.

The restrictions applying to this profile are the same as in the GOEP. In addition to these restrictions, the peer-to-peer synchronization is not supported by the BT synchronization.

2.4 PROFILE FUNDAMENTALS

The profile fundamentals are the same as defined in Section 2.4 in GOEP [2], with the addition of the requirements that bonding, link level authentication, and encryption (Fundamentals 1 and 3 in GOEP) must always be used for this profile. The OBEX authentication (Fundamental 2 in GOEP) as an application-level security mechanism must be supported by the devices providing this profile, but this profile does not mandate that it must be used.

In this profile, because both the IrMC Client and IrMC Server can act as a client (IrMC Server temporarily), both can initiate link and channel establishments; i.e. create a physical link between these two devices.

This profile does not mandate the IrMC server or client to enter any discoverable or connectable modes automatically, even if they are able to do so. This means that the end-user intervention may be needed on both the devices when, for example, the synchronization is initiated on the IrMC client device.

3 USER INTERFACE ASPECTS

3.1 MODE SELECTION

There are two modes associated with the Synchronization profile.

- Initialization Sync mode
- General Sync mode

In the **Initialization Sync** mode, the IrMC Server is in the Limited discoverable (or the General discoverable mode, see Section 6.5.1 in GOEP [2]), Connectable, and Pairable modes (See Section 4 in GAP [16]). The IrMC Client does not enter this mode in this profile. It is recommended that the Limited Inquiry procedure (Section 6.2 in GAP[16]) is used by the IrMC Client when discovering the IrMC server. Requirements on inquiry procedures are discussed in Section 6.5.1 of the GOEP [2].

In the **General Sync** mode, the device is in the Connectable mode. Both the IrMC Client and Server can enter this mode. For the IrMC Server, this mode is used when the IrMC Client connects the server and starts the synchronization at the subsequent times after pairing. For the IrMC Client, the mode is used when the synchronization is initiated by the IrMC server.

The devices are not required to enter these modes automatically without user intervention, even if they can do so. When entering either of these modes, IrMC Server and Client must ensure that the Object Transfer bit is set in the CoD (See [15]), and register a service record in the SDDB (See Section 7).

3.2 APPLICATION USAGE EVENTS

In the following sections (Section 3.2.1-3.2.3), the presented scenarios work as examples and variations in the actual implementations are possible and allowed.

3.2.1 Synchronization Scenario

When an IrMC Client wants to synchronize with an IrMC Server for the first time, the following scenario (Table 3.1) can be followed:

Step	IrMC Client	IrMC Server
1		The IrMC server device must be in the General Sync mode. If the device is not in this mode, the user must activate this mode on the device.
2	The user activates an application for synchronization.	
3	A list of devices in the RF proximity of the IrMC client is displayed to the user.	
4	The user selects a device to be connected and synchronized.	
5	The user is alerted if the device does not support the Synchronization feature, and the user may select another possible device (Step 4).	
6	The Bluetooth PIN code is requested from the user and entered on both devices.	
7	If OBEX authentication is used, the user enters the password for the OBEX authentication on both devices.	
8	The first synchronization is processed.	
9	The user may be notified of the result of the operation.	

Table 3.1: Usage Events for First Time Synchronization

At subsequent times, when the bonding is done, the scenario below (Table 3.2) can be followed.:

Step	IrMC Client	IrMC Server
1		The IrMC server device must be in the General Sync mode. If the device is not in this mode, the user must activate this mode on the device.
2	The user of the IrMC Client selects the Synchronization feature on the device, or another event triggers the synchronization to start in the IrMC client.	
3	The synchronization is processed.	
4	The User may be notified of the result of the operation.	

Table 3.2: Usage Events after First Time Synchronization

3.2.2 Sync Command Scenario

When an IrMC Server wants to initiate synchronization, and when the bonding and the possible OBEX initialization are done, the scenario below (Table 3.3) can be followed:

Step	IrMC Client	IrMC Server
1	The IrMC Client should be in the General Sync mode, without user intervention. Otherwise this operation is not applicable.	
2		The user selects the Sync Command feature in the IrMC Server, and the synchronization is initiated with the IrMC client. On the IrMC Server device, the user has earlier configured the IrMC Client to which the sync command is sent.
3	The synchronization is processed.	
4		The User may be notified of the result of the operation.

Table 3.3: Usage Events of Sync Command Scenario

3.2.3 Automatic Synchronization Scenario

When it is desired that an IrMC Server and Client synchronize automatically, and when the bonding and (possible) OBEX initialization are done, the scenario below (Table 3.4) can be followed.

Step	IrMC Client	IrMC Server
1	The IrMC Server enters the RF proximity of the IrMC Client. The Client notices it, and starts the synchronization without any notification to the User. The IrMC Server must be constantly in the General Sync mode so that the IrMC Client can notice the presence of the server in its RF vicinity.	
2	The synchronization is processed.	
3	The User may be notified of the result of the operation on both the devices.	

Table 3.4: Usage Events of Automatic Synchronization Scenario

4 APPLICATION LAYER

This section describes the feature requirements on units active in the Synchronization use case.

4.1 FEATURE OVERVIEW

Table 4.1 shows the required services:

	Features	Support in IrMC Client	Support in IrMC Server
1.	Synchronization of one or more of the following cases:	M	M
	Synchronization of phonebooks		
	Synchronization of calendars		
	Synchronization of emails		
	Synchronization of notes		
2.	Sync Command	M	O
3.	Automatic Synchronization	O	M

Table 4.1: Application layer features

4.2 SYNCHRONIZATION FEATURE

The support of Synchronization with IrMC level 4 functionality is mandatory for both IrMC Clients and IrMC Servers. The requirements for IrMC Synchronization are defined in the IrMC spec (See also Section 5). Bluetooth Synchronization must support at least one of the following cases (i.e. the application classes):

1. Synchronization of phonebooks
2. Synchronization of calendars
3. Synchronization of messages
4. Synchronization of notes

To achieve application level interoperability, the content formats are defined for Bluetooth Synchronization. The content formats are dependent on the application classes, which are designed for the different purposes. The supported application classes must be identified in terms of the data stores in the SDDB of the IrMC Server (See Section 7.1.1). For the application classes the content format requirements are:

- Phone Book applications must support data exchange using the vCard 2.1 content format specified in [13]. Section 7 of IrMC Specification [12] includes extensions to vCard2.1, which must also be supported by the actual implementations.
- Calendar applications must support data exchange using the vCalendar 1.0 content format specified in [14].
- Messaging applications must support data exchange using the vMessage content format in Section 9 of [12].
- Notes applications must support data exchange using the vNote content format specified in Section 10 of [12].

The above requirements are the minimal requirements, and the application utilizing any of these classes may store its objects in any internal content format the implementer chooses.

The support for the various mandatory and optional fields of the content formats listed above shall be in accordance with the IrMC Specification [12].

4.3 SYNC COMMAND FEATURE

This feature means that the IrMC client device works temporarily as a server and is able to receive a Sync Command from the IrMC server, which in this case acts temporarily as a client. This Sync Command orders the IrMC client to start synchronization with the IrMC Server.

After sending the sync command and getting the response for it, the IrMC Server must terminate the OBEX session and the RFCOMM data link connection.

This feature must be supported by the IrMC Client and it can optionally be supported by the IrMC Server. The formal requirements for this feature are defined in Section 5.8 in [12].

4.4 AUTOMATIC SYNCHRONIZATION FEATURE

In this feature, the IrMC Client can start the synchronization when the IrMC Server enters the RF proximity of the IrMC Client. Basically, this means that, on the Baseband level, the IrMC Client pages the IrMC Server at intervals and, when it finds that the IrMC Server is in the range, the IrMC Client can begin synchronization.

The support of this feature is optional for the IrMC Client but mandatory for the IrMC Server. This means that the IrMC Server must offer a capability to put the server device into the General Sync mode so that it does not leave this mode automatically.

5 IRMC SYNCHRONIZATION REQUIREMENTS

The IrMC specification [12] specifies IrMC Synchronization, which is utilized by this profile. The sections of the IrMC specification, with which this profile complies, are defined in Table 5.1.

Chapter	Name	Informative Sections	Mandatory Sections	Optional Sections	Not Applicable Sections
1	Introduction	All	-	-	-
2	IrMC Framework	2.1-3, 2.5.1, and 2.6-7	2.8.1-2, 2.8.4, and 2.9 (except 2.9.2)	2.8.3, and 2.9.2	2.4 and 2.5.2-3
3	Data Transmis-sions Services	3.3	3.1	-	3.2
4	OBEX Informa-tion Access and Indexing	4.1, 4.4.2, and 4.7	4.1.2, 4.2-3, 4.6, and 4.8	4.1.1 and 4.5	4.4.1
5	Synchronization	5.1 and 5.7	5.2-6 (except 5.5.3), and 5.8	5.5.3	-
6	Device Informa-tion	-	6.1-2	-	
7	Phone Book	7.1	7.3, 7.5, 7.7.1, 7.7.3, 7.7.5, 7.8.1, and 7.8.2	7.4, 7.6, 7.7.4, 7.7.6, and 7.8.3-5	7.2 and 7.7.2
8	Calendar	8.1	8.3, 8.5, 8.6.1, 8.6.3, 8.6.5, and 8.7	8.4 and 8.6.4	8.2, and 8.6.2
9	Messaging	9.1	9.3, 9.5, 9.8.1, 9.8.3, 9.8.6, and 9.9-10	9.4, 9.6-7, 9.8.4, and 9.8.5	9.2, and 9.8.2
10	Notes	10.1	10.3, 10.5, 10.6.1, 10.6.3, 10.6.5, and 10.7	10.4 and 10.6.4	10.2, and 10.6.2

Table 5.1: IrMC Specification Dependencies

Chapter	Name	Informative Sections	Mandatory Sections*	Optional Sections	Not Applicable Sections
11	Call Control	-	-	-	ALL
12	Audio	-	-	-	ALL
13	IrMC Applications IAS Entry and Service Hint Bit	-	-	-	ALL

Table 5.1: IrMC Specification Dependencies

*. Some of these sections may not be mandatory if the applications do not support all of the applications classes

This profile does not mandate that the functionality of IrMC level 1 must be supported for the different personal data objects (vcard, vcal, vmessage and vnote), although the IrMC specification requires its support. However, it is worth mentioning that the Push command of IrMC requires the level1 functionality for a text message. Thus, the IrMC client must be able to receive this command into its Inbox and the IrMC server must be able to send this command, if support for the Sync Command feature is claimed. For Bluetooth, the object push functionality and requirements are defined in the Object Push profile.

6 OBEX

6.1 OBEX OPERATIONS USED

Table 6.1 shows the OBEX operations which are required in the Synchronization profile.

Operation no.	OBEX Operation	Ability to Send		Ability to Respond	
		IrMC Client	IrMC Server*	IrMC Client*	IrMC Server
1	Connect	M	O	M	M
2	Disconnect	M	O	M	M
3	Put	M	O	M	M
4	Get	M	X	X	M
5	Abort	M	O	M	M
6	SetPath	X	X	X	X

Table 6.1: OBEX Operations

The columns marked with '*' refer to the Sync Command feature for which support in the IrMC Server is optional.

6.2 OBEX HEADERS

Table 6.2 shows the specified OBEX headers which are required in the Synchronization profile.

Header No.	OBEX Headers	IrMC Client	IrMC Server
1	Count	X	X
2	Name	M	M
3	Type	X	X
4	Length	M	M
5	Time	O	O
6	Description	O	O
7	Target	M	M

Table 6.2: OBEX Headers

Header No.	OBEX Headers	IrMC Client	IrMC Server
8	HTTP	O	O
9	Body	M	M
10	End of Body	M	M
11	Who	M	M
12	Connection ID	M	M
13	Authenticate Challenge	M	M
14	Authenticate Response	M	M
15	Application Parameters	M	M
16	Object Class	X	X

Table 6.2: OBEX Headers

6.3 INITIALIZATION OF OBEX

OBEX authentication must be supported by the devices implementing the Synchronization profile. The initialization procedure for OBEX is defined in Section 5.3 in GOEP [2].

6.4 ESTABLISHMENT OF OBEX SESSION

The Target header must be used when the IrMC client establishes the connection (See Section 5.4 in GOEP [2]). The Target header value is 'IRMC-SYNC'.

6.5 PUSHING DATA

See Section 5.5 in GOEP [2].

6.6 PULLING DATA

See Section 5.6 in GOEP [2].

6.7 DISCONNECTION

See Section 5.7 in GOEP [2].

7 SERVICE DISCOVERY

7.1 SD SERVICE RECORDS

There are two separate services related to the Synchronization profile. The first is the actual synchronization server (i.e. IrMC server), and the second is the sync command server (i.e. IrMC Client).

7.1.1 Synchronization Service

In this case, the service is the IrMC server. The following information (i.e. service records) must be put into the SDDB.

Item	Definition:	Type/ Size:	Value:*	AttrID:	Status:	Default Value:
Service Class ID List				See [15]	M	
Service Class #0		UUID	IrMCSync		M	
Protocol Descriptor list				See [15]	M	
Protocol ID #0		UUID	L2CAP		M	
Protocol ID #1		UUID	RFCOMM		M	
Param #0	CHANNEL	Uint8	Varies		M	
Protocol ID #2		UUID	OBEX		M	
Service name	Displayable Text name	String	Varies	See [15]	O	'IrMC Synchro-nization'
BluetoothProfileDe-scriptorList	Supported profiles and versions			See [15]	O	
Profile #0		UUID	IrMCSync			IrMC-Sync
Version #0		Uint16	Varies			0x0100
Supported Data Stores List	Data stores may be phonebook, calendar, notes, and messages.	Data Ele-ment Sequence of UInt8	Data stores: **0x01** = Phonebook **0x03** = Cal-endar **0x05** = Notes **0x06** = Mes-sages	See [15]	M	

Table 7.1: Synchronization Service Record

*. Values that are of the type UUID are defined in the Assigned Numbers specification [15].

7.1.2 Sync Command Service

The Sync Command service is used for initiating the synchronization from the IrMC server device. The following service records must be put into the SDDB by the application which provides this service.

Item	Definition:	Type/ Size:	Value:*	AttrID:	Status:	Default Value:
Service Class ID List				See [15]	M	
Service Class #0		UUID	IrMCSync-Command		M	
Protocol Descriptor list				See [15]	M	
Protocol ID #0		UUID	L2CAP		M	
Protocol ID #1		UUID	RFCOMM	See [15]	M	
Param #0	CHANNEL	Uint8	Varies		M	
Protocol ID #2		UUID	OBEX		M	
Service name	Displayable Text name	String	Varies		O	‘Sync Com-mand Service’
BluetoothProfileDe-scriptorList	Supported profiles and versions			See [15]	O	
Profile #0		UUID	IrMCSync			IrMC-Sync
Version #0		Uint16	Varies			0x0100

Table 7.2: Sync Command Service Record

*. Values that are of the type UUID are defined in the Assigned Numbers specification [15].

7.2 SDP PROTOCOL DATA UNITS

Table 7.3 shows the specified SDP PDUs (Protocol Data Units) which are required in the Synchronization profile.

PDU no.	SDP PDU	Ability to Send		Ability to Retrieve	
		IrMC Client	IrMC Server	IrMC Client	IrMC Server
1	SdpErrorResponse	M*	M	M	O*
2	SdpServiceSearchAttribute-Request	M	O*	M*	M
3	SdpServiceSearchAttribute-Response	M*	M	M	O*

Table 7.3: SDP PDUs

The PDUs marked with '*' refer to the Sync Command feature, of which the support in the IrMC Server is optional.

8 REFERENCES

8.1 NORMATIVE REFERENCES

[1] Bluetooth Special Interest Group, IrDA Interoperability.

[2] Bluetooth Special Interest Group, Generic Object Exchange 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.

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

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

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

[15] Bluetooth Special Interest Group, Bluetooth Assigned Numbers.

[16] Bluetooth Special Interest Group, Bluetooth Generic Access Profile Specification.

Appendix I

REVISION HISTORY

REVISION HISTORY

Part K:1
Generic Access Profile

Rev	Date	Comments
1.0	June 20th 1999	• Released for final review. • Release for sign-off. • Updated based on received comments. • Final updated version. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.

Part K:2
Service Discovery Application Profile

Rev	Date	Comments
1.0	June 20th 1999	• Aligned with GAP whenever necessary. • Emphasized that SDAP can be used as the basis for the service discovery portion of other profiles. • Added section 5.1 with SDP PDU exchange examples. • Emphasized that normal operation requires a LocDev to initiate and terminate L2CAP connections for SDP. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.

Part K:3
Cordless Telephony Profile

Rev	Date	Comments
1.0	June 20th 1999	• Ftf preversion, some editorial updates and minor content changes - number of TLs 8 -> 7, Master-Slave Switch made conditional, restrictions in digits for Called&Calling party IEs, updates to CoD and SDP sections. • Updates after ftf. Added feature "Register recall", removed feature "service call" and redefined "Multi-terminal support" to reflect decisions on WUG status. Added description of Register recall to section 4.3. Removed emergency, service and ad-hoc call classes. Added description of piconet handling to 4.1.2. Updated and reworked SDP record. Additions to contributor list. Figure in section 8.2 removed. "Status" chapter removed. Added remark on security with respect to L2CAP connectionless. Editorial updates to section 4.4. • Updates to incorporate GAP and editorial guidelines for the specification • Errors in tables 3 and 4 and section 4.2. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.

Part K:4
Intercom Profile

Rev	Date	Comments
1.0	June 20th 1999	• Update after F2F, incorporating technical issues only. • Editorial improvements. • Replaced bonding with authentication in Section 2.4. • Corrected references to LMP. • Removed PSM field from service record, and rephrased opening statement of SDP section. • Added chapter on GAP interoperability requirements. • Final GAP alignment. • Mandated call confirmation as SETUP confirmation. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.

Part K:5
Serial Port Profile

Rev	Date	Comments
1.0	June 20th 1999	• Added more details on application layer procedures (chapter 3). First alignment with Generic Access Profile. • Added requirements on SDP procedures. More alignment with GAP. • Corrected some typos. Removed section 5.3.3 (Link Power Mode in L2CAP). Removed “Management entity” throughout document. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.

Part K:6
Headset Profile

Rev	Date	Comments
1.0	June 20th 1999	• Update after F2F, incorporating outstanding issues as discussed (volume control and synchronisation, added AT command +VGS and +VGM, extended audio connection transfer description, authentication/ encryption optional to use, status change of outgoing audio connection, service record updated) and various editorial issues (amongst others update of contributors list). Removed status and history section. • Remote audio volume control: replaced may’s with shall’s to make it more consistent (if Remote audio volume control is supported, the entire procedure shall be supported as specified). • SDP - removed PSM for RFCOMM, added misplaced server channel. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.

Part K:7
Dial-up Networking Profile

Rev	Date	Comments
1.0	June 20th 1999	• Some SDP values filled in, CoD updated after assigned numbers doc. • Updates after Tampere ftf: SDP record updated and reworked. Removed table from chapter 5.1 (now in RFCOMM). Updated contributors list. Figure removed from section 5.5.1. • Added profile structure section. Alignment with GAP (section 6) added. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.

Part K:8
Fax Profile

Rev	Date	Comments
1.0	June 20th 1999	• Replaced bonding with authentication in Section 2.4. • Corrected references to LMP. • removed PSM field from service record, and rephrased opening statement of SDP section. • Added chapter on GAP interoperability requirements. • Updated Figure 1, Service discovery Profile to Service Discovery Application Profile. • Removed “ME” block from both sides of figure 2. • Removed paragraph discussing “ME” in section 2.1. • Renamed Section Heading 4 From Dialling and Control to Dialling and Control Interoperability requirements. • Re-worded section 4.1.2. • Removed the words “the” and “section” from the last sentence in section 5.3, paragraph 2. • Removed section 5.6. • Aligned section 6 with new changes from Dialup networking profile. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.

Part K:9
LAN Access Profile

Rev	Date	Comments
1.0	June 20th 1999	• Updated Service records in line with “best practice”. • Removed Security section. • Editorial changes in Section 4.1. • Editorial changes in Section 5.1. • Editorial changes in Section 11.2. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.

Part K:10
Generic Object Exchange Profile

Rev	Date	Comments
1.0	June 20th 1999	• Updated Chapter 1.2 and added reference to GAP regarding link and channel establishment. • Removed the security statement from the Profile fundamentals chapter, clarified the use of the limited discoverable mode in the Inquiry and Inquiry Scan chapter, and added the GAP requirement chapter. • Changed the 'initialization' wording to 'bonding', added some cross-references, and included the errata for IrOBEX in the reference list. • Management entity removed and the fall back procedure added if the Limited Inquiry procedure is supported. • Clarified that the fall back to the General inquiry is mandatory if Limited Inquiry is used. • Editorial changes and Chapter 7.3.1 (Bonding) updated to describe the result of Bonding. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.

Part K:11
Object Push Profile

Rev	Date	Comments
1.0	June 20th 1999	• Removed PSM from SDP record. Updated text in Profile Structure 1.2. GAP alignment in Profile Fundamentals. • Editorial. • Renamed Initialization to Bonding. • Removed the ME section and references to ME. Stated in profile fundamentals that bonding is mandatory to support and optional to use. Removed “Notation for timers and counters”. Changed wording in application procedure for object push feature. Minor update of SDP record. • Changed recommended inquiry procedure in chapter 3 to reference to the GOEP. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.

Part K:12
File Transfer Profile

Rev	Date	Comments
1.0	June 20th 1999	• SDP table changes, addition of references to doc [16]. • More SDP table changes, alignment with GAP, contributors update, and copyright notice. • Editorial and reference corrections. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view.

Part K:13
Synchronization Profile

Rev	Date	Comments
1.0	June 20th 1999	• Updated service records, IrMC chapter updated. • Chapter 1.2 updated, profile fundamentals clarified, recommended inquiry procedure added into Chapter 3 and service records updated. • Security issues clarified in Profile Fundamentals chapter and some editorial changes. • Change the 'Initialization' wording to 'Bonding', updated cross-references, and editorial changes. • Remove Management entity, removed statement that IrMC client must initiate the link establishment when bonding is not performed, and added a reference to the inquiry procedures of GOEP. • Editorial changes. • 1.0 Draft
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.

Appendix II

CONTRIBUTORS

CONTRIBUTORS

Part K:1
Generic Access Profile

Patric Lind (editor)	Ericsson Mobile Communications AB
Elg, Johannes	Ericsson Mobile Communications AB
Johan Sörensen	Ericsson Mobile Communications AB
Erik Slottboom	Ericsson Mobile Communications AB
Thomas Muller	Nokia Mobile Phones
Stephane Bouet	Nokia Mobile Phones
Martin Roter	Nokia Mobile Phones
Brian Redding	Motorola
Chatschik Bisdikian	IBM
Ken Morley	3COM
Jon Inouye	Intel

Part K:2
Service Discovery Application Profile

Elg, Johannes	Ericsson Mobile Communications AB
Bisdikian, Chatschik (editor)	IBM Corporation
Miller, Brent	IBM Corporation
Muller, Thomas	Nokia
Pascoe, Robert A.	XtraWorX

Part K:3
Cordless Telephony Profile

Olof Dellien (editor)	Ericsson Mobile Communications AB
Erik Slotboom	Ericsson Mobile Communications AB
Gert-jan van Lieshout	Ericsson Mobile Communications AB
Martin Roter	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Ken Morley	3COM
Richard Shaw	3COM
Brian Redding	Motorola
Alex Feinman	Motorola
Sridhar Rajagopal	Intel Corporation
Ramu Ramakesavan	Intel Corporation
Jun'ichi Yoshizawa	Toshiba

Part K:4
Intercom Profile

Erik Slotboom (editor)	Ericsson Mobile Communications AB
Olof Dellien	Ericsson Mobile Communications AB
Gert-jan van Lieshout	Ericsson Mobile Communications AB
Richard Shaw	3COM
Ken Morley	3COM
Shridar Rajagopal	Intel Corporation
Ramu Ramakeshavan	Intel Corporation
Brian Redding	Motorola
Thomas Müller	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Jun'ichi Yoshizawa	Toshiba

Part K:5
Serial Port Profile

Johan Sörensen (editor)	Ericsson Mobile Communications AB
Olof Dellien	Ericsson Mobile Communications AB
Riku Mettälä	Nokia Mobile Phones

Part K:6
Headset Profile

Richard Shaw	3Com
Ken Morley	3Com
Erik Slotboom (editor)	Ericsson Mobile Communications AB
Olof Dellien	Ericsson Mobile Communications AB
Bailey Cross	Intel Corporation
Shridar Rajagopal	Intel Corporation
Brian Redding	Motorola
Alex Feinman	Motorola
Thomas Müller	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Martin Roter	Nokia Mobile Phones
Jun’ichi Yoshizawa	Toshiba

Part K:7
Dial-up Networking Profile

Olof Dellien (editor)	Ericsson Mobile Communications AB
Erik Slotboom	Ericsson Mobile Communications AB
Riku Mettälä	Nokia Mobile Phones
Martin Roter	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Ken Morley	3COM
Richard Shaw	3COM
Brian Redding	Motorola
Alex Feinman	Motorola
Sridhar Rajagopal	Intel Corporation
Jun'ichi Yoshizawa	Toshiba

Part K:8
Fax Profile

Richard Shaw (editor)	3COM
Ken Morley	3COM
Olof Dellien	Ericsson Mobile Communications ABn
Erik Slotboom	Ericsson Mobile Communications AB
Gert-jan van Lieshout	Ericsson Mobile Communications AB
Riku Mettälä	Nokia Mobile Phones
Christian Zechlin	Nokia Mobile Phones
Thomas Müller	Nokia Mobile Phones
Martin Roter	Nokia Mobile Phones
Sridhar Rajagopal	Intel Corporation
Ramu Ramakeshavan	Intel Corporatio
Brian Redding	Motorola
Alex Feinman	Motorola
Jun'ichi Yoshizawa	Toshiba

Part K:9
LAN Access Profile

Jon Burgess	3COM
Paul J Moran (editor)	3COM
Phil Crooks	3COM
Johan Sorensen	Ericsson Mobile Communications AB
Chatschik Bisdikian	IBM Corporation
Marcia Peters	IBM Corporation
Pravin Bhagwat	IBM Corporation
Kris Fleming	Intel Corporation
Michael Camp	Nokia Mobile Phone
Shaun Astarabadi	Toshiba Corporation
Yosuke Tajika	Toshiba Corporation

Part K:10
Generic Object Exchange Profile

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

Part K:11
Object Push Profile

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

Part K:12
File Transfer Profile

Shaun Astarabadi (editor)	Toshiba Corporation
David Suvak	Extended Systems
Patrik Olsson	Ericsson Mobile Communications AB
Apratim Purakayastha	IBM Corporation
Aron Walker	IBM Corporation
Jon Inouye	Intel Corporation
Stephane Bouet	Nokia Mobile Phones
Riku Mettälä	Nokia Mobile Phones
James Scales	Nokia Mobile Phones
Steve Rybicki	Puma Technology
Katsuhiro Kinoshita	Toshiba Corporation

Part K:13
Synchronization Profile

David Kammer	3COM
Patrik Olsson	Ericsson Mobile Communication AB
David Suvak	Extended Systems
Brent Miller	IBM Corporation
Apratim Purakayastha	IBM Corporation
Aron Walker	IBM Corporatio
Jon Inouye	Intel Corporation
Stephane Bouet	Nokia Mobile Phones
Riku Mettälä (editor)	Nokia Mobile Phone
James Scales	Nokia Mobile Phone
Steve Rybicki	Puma Technology
Shaun Astarabadi	Toshiba Corporation
Katsuhiro Kinoshita	Toshiba Corporation

The Bluetooth Specification was compiled and edited by Dan Sönnerstam, Pyramid Communication AB.

Appendix III

ACRONYMS AND ABBREVIATIONS

LIST OF ACRONYMS AND ABBREVIATIONS

Abbreviation or Acronym	Meaning
ACL	Asynchronous Connectionless
AG	Audio Gateway
AP	Access Point
B	
BB	Baseband
BD_ADDR	Bluetooth Device Address
C	
CC	Call Control
CL	Connectionless
CO	Connection-oriented
CoD	Class Of Device
CTP	Cordless Telephony Profile
D	
DAC	Device Access Code
DIAC	Dedicated Inquiry Access Code
DT	Data Terminal
DT	Data Terminal
F	
FHS	Frequency Hopping Synchronization
G	
GAP	Generic Access Profile
GIAC	General Inquiry Access Code
GM	Group Management
GOEP	Generic Object Exchange Profile
GW	Gateway
H	
HCI	Host Controller Interface
HS	Headset

I	
IP	Internet Protocol
IPX	Internet Protocol eXchange
IrDA	Infrared Data Association
IrMC	Ir Mobile Communications
L	
L2CA	Logical Link Control And Adaptation
L2CAP	Logical Link Control And Adaptation Protocol
LAN	Local Area Network
LAP	LAN Access Point
LC	Link Controller
LIAC	Limited Inquiry Access Code
LM	Link Manager
LMP	Link Manager Protocol
LocDev	Local Device
M	
ME	Management Entity
MM	Mobility Management
MSC	Message Sequence Chart
MTU	Maximum Transmission Unit
O	
OBEX	Object Exchange Protocol
P	
PC	Personal Computer
PDA	Personal Digital Assistant
PDU	Protocol Data Unit
PIM	Personal Information Management
PPP	Point-to-Point Protocol
PSTN	Public Switched Telephone Network
Q	
QoS	Quality Of Service

R	
RemDev	Remote Device
RFCOMM	Serial Cable Emulation Protocol
S	
SD	Service Discovery
SDDB	Service Discovery Database
SDP	Service Discovery Protocol
SeP	Serial Port
SIG	Special Interest Group
SrvDscApp	Service Discovery Application
T	
TCP	Transport Control Protocol
TCS	Telephony Control Specification
TL	Terminal
TL_O	Terminal Originating A Call
TL_T	Terminal Terminating A Call
U	
UDP	User Datagram Protocol
UI	User Interface
UIAC	Unlimited Inquiry Access Code
UUID	Universally Unique Identifier
W	
WUG	Wireless User Group

INDEX

Numerics

3-in-1 phone 100

A

adaptation layer 178
Assigned Numbers 383
AT command set 231, 254
Audio connection transfer 204
Audio Gateway 197
Authentication 54
authentication 33, 173, 175
Automatic Synchronization 402

B

Bluetooth Bonding 42
Bluetooth channel establishment 48
Bluetooth connection establishment 50
Bluetooth Device Address 25
Bluetooth Device Class 27
Bluetooth Device Discovery 41
Bluetooth Device Inquiry 37, 39
Bluetooth Device Name 26
Bluetooth Device Name Discovery 40
Bluetooth Device Type 27
Bluetooth link establishment 45
Bluetooth Passkey 26
Bluetooth Service Type 27
bondable mode 32
Bonding 55, 173
bonding 42
browsing 174
Business Card Exchange 346
Business Card Exchange function 340
Business Card Pull 345
Business Card Pull function 340

C

Calendar 344
Calendar application 403
Call information 106, 147
Calling line identification presentation 106
Channel establishment 54
channel establishment 48
Class of Device 185
Client 310
CLIP 106
Connectable device 53
connectable mode 31
Connection establishment 54
connection establishment 50
Connection ID 375, 376
Connection Management 106
Content Formats 344
content formats 344
Cordless Telephony profile 100

D

Data Access Points 246
Data calls 230
data link connection 175
Data Terminal 227, 250, 272
DCE 172
Default Get Object 346
Device discovery 54
Dial-up Networking 223
Direct Cable Connection 294
Discoverable device 53
discoverable mode 29
DT 431
DTE 172
DTMF signalling 106

E

encryption 173, 175, 365

F

Fax profile 246
Fax service 253
File Browsing UUID 375
File Transfer 359
File Transfer mode 367
File Transfer profile 360
flow control 178
flush timeout 180

G

Gateway 104, 227, 250
General Discoverable 367
general discoverable mod 31
general inquir 37
General Inquiry procedure 185
General Sync mode 399
Generic Object Exchange profile 306, 360, 392

GET-operation 322
GOEP 360, 392
GW 104

H
Headset 193, 197
Headset Control 196

I
Incoming audio connection 201
Incoming external call 106
Initialisation 106
Initialization Sync mode 399
inquiry 185
inquiry scan 185
Intercom 143
Intercom call 107, 147
Internet Bridge 223
IP 269
IPX 269
IrMC Client 397
IrMC Server 397
IrMC specification 404
IRMC_SYNC 407

L
LAN Access 269
LAN Access Point 271
latency 180
legacy applications 171, 175
Limited Discoverable 367
Limited discoverable mode 185
limited discoverable mode 30
limited inquiry 38
Link 52
Link establishment 54
link establishment 45
link level authentication 365
low power mode 175

M
ME 432
Messaging 344
Messaging application 403
Modem Status Command 178
MTU sizes 180
Multi-terminal support 107

N
Name discovery 54
name discovery 40
non-bondable mode 32
non-connectable mode 31
non-discoverable mode 29
non-pairable mode 32
Notes 345
Notes application 403

O
OBEX 305
OBEX authentication 315, 365, 375, 407
OBEX Connect 376
OBEX connection 375
OBEX Headers 348
OBEX headers 348, 349
OBEX operation 314
OBEX Operations 348
OBEX operations 348
Object Exchange mode 340
Object Push 334, 344
Object Push function 340
On hook 107, 147
Outgoing audio connection 202
Outgoing external call 107
owner's business card 346

P
paging 185
pairable mode 32
Phone Book 344
Phone Book application 403
PIM 392
Post-dialling 107
PPP 269, 281
Push Client 339
Push Server 339
PUT-operation 321

Q
Quality of Service 180

R
Register recall 107
Remote audio volume control 205
Remote Line Status indication command 178
Remote Port Negotiation Command 178
RFCOMM Server channel 174
RFCOMM session 174
RS232 control signalling 172
RS232 control signals 178

S

SDP database 175
security features 173
security mode 34
serial port 171
Server 310
Service Class ID 174
service record 181
Sync Command 402
Synchronization 391
Synchronization profile 392

T

Target header 375
Terminal 104
TL 104

U

UUID 383

V

virtual serial cable 175
virtual serial port 172

W

walkie-talkie 143
Wide Area Networks 246