BLUETOOTH DOC	Date / Year-Month-Day	Approved	Revision	Document No	
	2012-07-24	Draft	V12	SPP_SPEC	
Prepared By	E-mail Address			N.B.	
BARB	barb-main@bluetooth.org				

SERIAL PORT PROFILE

Abstract:

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.

BLUETOOTH DOC	Date / Year-Month-Day	Approved	Revision	Document No	
	2012-07-24	Draft	V12	SPP_SPEC	
Prepared By	E-mail Address			N.B.	
BARB	barb-main@bluetooth.org				

Revision History

Revision	Date	Comments
D12r00	15 August 2005	Updated for v1.2 or Later edits
D12r01	14 September 2005	Editorially updated
D12r02	31 October 2005	Formatting changes
D12r03	30 November 2005	Editorial updates
D12r04	15 December 2005	Further updates for v1.2
D12r05	08 February 2006	Editorial updates
D12r06	20 March 2006	Input Reviewer's comments
D12r07	01 August 2007	Updates for 2.1 + EDR, updates to References.
D12r08	30-January 2012	ESR05: E386, Section 2.1
D12r09	30 April 2012	Addressed review comments; spell checked; added link for Bluetooth Assigned Numbers; added links to references.
D12r10	29 May 2012	Add outline to Figure 1.1 Change equally to Equally, in [386] Change "the" to uppercase in Informative note. Added space after cross reference.
V12	24 July 2012	Adopted by the Bluetooth SIG Board of Directors

Contributors

Name	Company
Riku MettŠlä	Nokia Mobile Phones
Olof Dellien	Telefonaktiebolaget LM Ericsson
Johan Sšrensen	Telefonaktiebolaget LM Ericsson

Disclaimer and Copyright Notice

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

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

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

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

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

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

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

Copyright © 2012. Bluetooth® SIG, Inc. All copyrights in the Bluetooth Specifications themselves are owned by Ericsson AB, Lenovo (Singapore) Pte. Ltd., Intel Corporation, Microsoft Corporation, Motorola Mobility, Inc., Nokia Corporation, and Toshiba Corporation.

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

Contents

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.
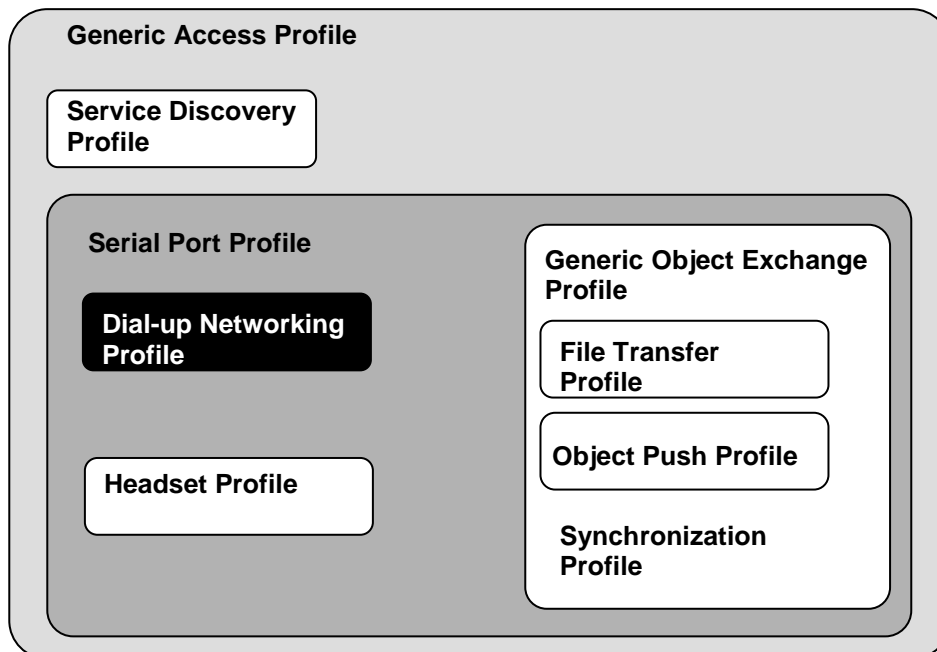

Figure 1.1: Bluetooth Profiles

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

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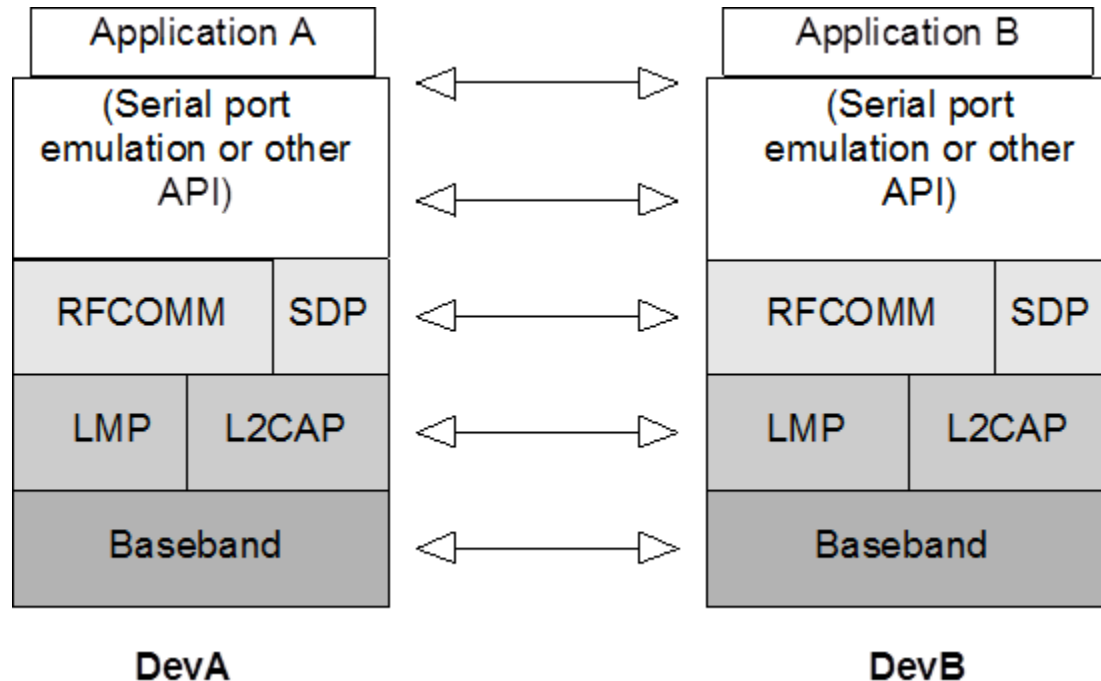

Figure 2.1: Protocol model

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 2.2 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). Nevertheless, 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.) Equally, though, non-legacy applications wishing to perform serial communications over Bluetooth must also adhere to the behavior specified in this profile. This is the case regardless of whether they make use of a Bluetooth-aware helper as described above, or by use of some other interface to the Bluetooth stack. This ensures that all combinations of legacy and non-legacy applications remain interoperable at the Bluetooth level.

2.2 Configurations and roles

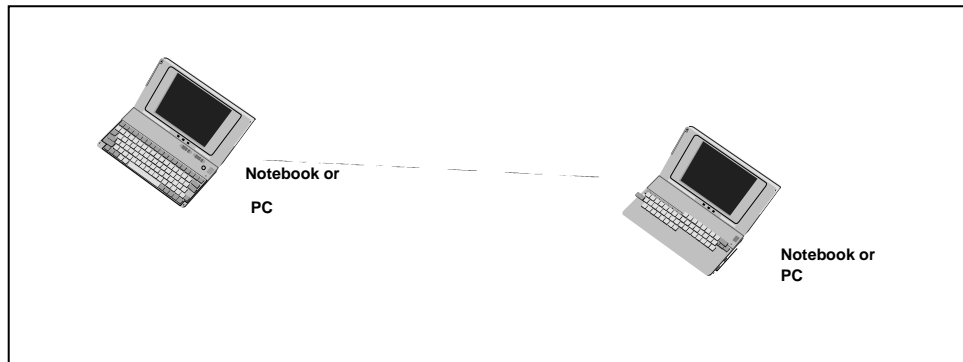Figure 2.2shows one possible configuration of devices for this profile:

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 signaling).

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.

- 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. Alternatively, 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 is 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. However, 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, is 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.	Signaling	
	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
	Retransmission and Flow Control	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 broadcast data and unicast connectionless data 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 Signaling

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 signaling.

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 unless L2CAP Enhanced Retransmission mode is negotiated for the L2CAP channel bearing RFCOMM (see section 5.3.4). If L2CAP Enhanced Retransmission mode is negotiated, the flush timeout value may be set according to other profiles running over the same ACL logical transport.

5.3.3 Quality of Service

Negotiation of Quality of Service is optional in this profile.

5.3.4 Flow and Error Control

For any products that will be transferring large data files and where the receiving device will be subject to radio interference causing packet losses, it is recommended that the Error Control feature in L2CAP (Core Specification V3.0 and later) be utilized by configuring the channel to use Enhanced Retransmission mode.

Serial Port data is carried over a reliable L2CAP channel. The reliability of this channel may be realized using an infinite flush timeout on the ACL logical transport. However, if the Serial Port profile coexists with other profiles on the same physical link then it may not be acceptable to configure the ACL logical transport in this way. Therefore, for devices that expect the Serial Port profile to coexist with other profiles on a physical link, L2CAP Enhanced Retransmission mode (introduced in Core Specification Version 3.0) should be negotiated for the channel bearing RFCOMM. Realizing the reliable L2CAP channel in this way allows the ACL logical transport to have a finite flush timeout suitable for other profiles.

L2CAP Enhanced Retransmission mode may also be negotiated for applications using the Serial Port profile that require greater reliability than that provided by a non-flushing ACL logical transport.

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.

Serial Port Profile versions earlier than v1.2 did not mandate the BluetoothProfileDescriptorList attribute. New DevA devices shall assume that a remote DevB without this attribute does not support Serial Port Profile version 1.2 or greater.

Item			Definition	Type/Size	Value	Status	Default Value
ServiceClassIDList					Note1	M	
	ServiceClass #0			UUID	SerialPort Notes 1 & 3	M	
ProtocolDescriptorList						M	
	Protocol ID #0			UUID	L2CAP Note1	M	
	Protocol ID #1			UUID	RFCOMM Note1	M	
		ProtocolSpecificParameter0	Server Channel	Uint8	N = server channel #	M	
BluetoothProfileDescriptorList						M	
	Profile ID #0		Supported Profiles	UUID	SerialPortProfile	M	
		Parameter #0	Profile Version	Uint16	0x0102	M	
ServiceName			Displayable text name	String	Varies	O	"COM5" Notes 2 & 4

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 [2]), 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 if a low-power mode is used, or when it is used. 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 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.

8.2 Inquiry Scan

For inquiry scan, (at least) the GIAC shall be used, according to one of the discoverable modes defines in GAP [8] Section 4.1.2 and Section 4.1.3. That is, it is allowed to use only 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 whether DevB is engaged in the execution of the Serial Port Profile or not (due to 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.)

8.3 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.)

8.4 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] Baseband specification v1.2 or later

[2] Link Manager Protocol v1.2 or later

[3] L2CAP Specification v1.2 or later

[4] RFCOMM with TS 07.10

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

[6] Service Discovery Protocol (SDP) v1.2 or later

[7] Service Discovery Application Profile v1.2 or later

[8] Bluetooth Assigned Numbers

[9] Generic Access Profile v1.2 or later