BLUETOOTH® DOC	Date / Year-Month-Day 2011-09-15	Approved	Revision V10r00	Document No ANP_SPEC
Prepared By PUID WG	E-mail Address rd-main@bluetooth.org			N.B.

ALERT NOTIFICATION PROFILE

Abstract

This profile enables a client device to receive different types of alerts and event information, as well as information on the count of new alerts and unread items, which exist in the server device.

Revision History

Revision	Date	Comments
D09r01	2011-04-05	First Draft
D09r02	2011-05-02	Update based on the latest template
D09r03	2011-05-26	Added connection establishment and update with ICSD09r02
D09r03	2011-06-02	Added Characteristic Discovery by UUID and others
D09r04	2011-06-24	Deleted BR/EDR, added security description and new characteristic related issue
D09r05	2011-07-05	Updated Section5 with FMP
D09r06	2011-07-21	Updated section 2.3,2.4, 5, 6 based on PAS profile and others
D09r07	2011-07-27	Clean version for final WG review
D09r08	2011-08-01	Added futures after reconnection
D09r09	2011-08-10	Cleaned up word (reference issues before Barb submission)
D09r10	2011-08-18	Responded to Barb and GPA reviewers
D09r11	2011-08-20	Responded to some additional comments from Terry
D09r12	2011-08-24	Added requirement to read “Supported...” after connection setup
V09r00	2011-08-26	Adopted prototype specification
D10r01	2011-09-02	First draft D10
V10r00	2011-09-15	Adopted by the Bluetooth SIG Board of Directors

Contributors

Name	Company
Koyama Shunsuke	Seiko Epson
Satoshi Oshiyama	Seiko Epson
Sadao Nagashima	Casio
Daisuke Matsuoh	Citizen
Steve Davies	Nokia
Frank Bentsen	Nordic

Disclaimer and Copyright Notice

Document Terminology

The Bluetooth SIG has adopted Section 13.1 of the IEEE Standards Style Manual, which dictates use of the words ``shall'', ``should'', ``may'', and ``can'' in the development of documentation, as follows:

The word *shall* is used to indicate mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).

The use of the word *must* is deprecated and shall not be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

The use of the word *will* is deprecated and shall not be used when stating mandatory requirements; *will* is only used in statements of fact.

The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

Table of Contents

1 Introduction

The Alert Notification profile allows a device like a watch to obtain information from a cellphone about incoming calls, missed calls and SMS/MMS messages. The information may include the caller ID for an incoming call or the sender's ID for email/SMS/MMS but not the message. This profile also enables the client device to get information about the number of unread messages on the server device.

1.1 Profile Dependencies

This profile is compatible with any *Bluetooth* core specification host that includes the Generic Attribute Profile (GATT).

1.2 Conformance

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

2 Configuration

2.1 Roles

The profile defines two roles:

- Alert Notification Server
- Alert Notification Client

2.2 Roles/Service Relationship

The diagram below shows the relationships between service and the two profile roles.

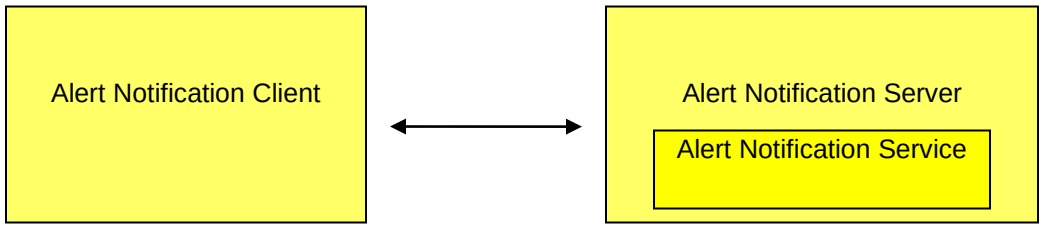

Figure 2.1: Role and Service Relationships

2.3 Concurrency

The Alert Notification profile may run concurrently with other profiles.

In multi-profile scenarios (for example, with MAP), the Alert Notification Client should act as an extension of the phone's UI and thus the alerts of the Alert Notification device should present only alert information that can be available via the phone UI in any given situation.

2.4 Topology

The Alert Notification Server shall implement the GAP central role and may implement the GAP Peripheral role. The Alert Notification Client shall implement the GAP Peripheral role and may implement the GAP Central role.

2.5 Transport Dependencies

This profile is specified for operation over the LE transport.

3 Alert Notification Server Requirements

This profile does not impose any additional requirements on the Alert Notification service beyond those defined by the Alert Notification Service specification.

3.1 Alert Notification Service

The Alert Notification Server Profile role shall have one instance of the Alert Notification service.

4 Alert Notification Client Requirements

This section describes the procedure requirements for an Alert Notification Client.

	Procedure	Ref.	Support in Alert Notification Client
1.	Service Discovery	4.1	M
2.	Characteristic Discovery	4.2	M
3.	Read the Value of Supported New Alert Category	4.3	M
4.	Read the Value of Supported Unread Alert Category	4.4	O
5.	Receive Notification of New Alert	4.5	M
6	Request to Notify when New Alert Count Change	4.6	M
7	Receive Notification of Unread Alert Status	4.7	O
8	Request to Notify when Unread Alert Status Change	4.8	C.1
9	Configure Alert Notification Control Point	4.9	M
10	Recovery from Connection Loss for New Alerts	4.10	M
11	Recovery from Connection Loss for Unread Alerts	4.11	C.1
12	Check Supported New Alert Category after Connection Setup	4.12	M
13	Check Supported Unread Alert Status Category after Connection Setup	4.13	C.2
C.1: Mandatory if procedure 7 is supported, otherwise excluded C.2: Mandatory if procedure 4 is supported, otherwise excluded			

4.1 Service Discovery

The Alert Notification Client shall perform service discovery using the GATT *Discover All Primary Services* sub-procedure or the GATT *Discover Primary Services by Service UUID* sub-procedure using «Alert Notification Service» for the service UUID.

4.2 Characteristic Discovery

The GATT sub-procedure *Discover All Characteristics of a Service* or the GATT sub-procedure *Discover Characteristics by UUID* shall be used to discover the characteristics of the Alert Notification service.

The GATT *Discover All Characteristic Descriptors* sub-procedure shall be used to discover the Client Characteristic Configuration descriptor.

4.3 Read the Value of Supported New Alert Category

The Alert Notification Client shall read the value of the Supported New Alert Category in the Alert Notification Server.

4.4 Read the Value of Supported Unread Alert Category

The Alert Notification Client shall read the value of the Supported Unread Alert Category in the Alert Notification Server.

4.5 Receive Notification of New Alert

The Alert Notification Client shall receive notifications of the New Alert characteristic from the Alert Notification Server.

4.6 Request to Notify when New Alert Count Changes

The Alert Notification Client can receive notifications from the Alert Notification Server by configuring the *Client Characteristic Configuration* descriptor for the New Alert characteristic in the Alert Notification Server. To configure the notification of New Alert, the handle of the New Alert characteristic and its *Client Characteristic Configuration* descriptor shall have been discovered and stored using the Characteristic Discovery procedure.

To enable the notification of the New Alert characteristic, the GATT *Write Characteristic Descriptors* sub-procedure shall be used to set the Notification bit in the *Client Characteristic Configuration* descriptor.

4.7 Receive Notification of Unread Alert Status

The Alert Notification Client may receive notifications of the Unread Alert Status characteristic from the Alert Notification Server.

4.8 Request to Notify when Unread Alert Status Changes

The Alert Notification Client can receive notification from the Alert Notification Server by configuring the *Client Characteristic Configuration* descriptor for the Unread Alert Status characteristic in the Alert Notification Server. To configure the notification of Unread Alert Status, the handle of the Unread Alert Status characteristic and its *Client Characteristic Configuration* descriptor shall have been discovered and stored using the Characteristic Discovery procedure.

To enable the notification of the Unread Alert Status characteristic, the GATT *Write Characteristic Descriptors* sub-procedure shall be used to set the Notification bit in the *Client Characteristic Configuration* descriptor.

4.9 Configure Alert Notification Control Point

The Alert Notification Client may write commands including Category ID for a subset of the supported categories to the Alert Notification Control Point to restrict the delivery of notifications to those categories only. The commands will apply to new alerts or unread alerts, or both.

To request the Alert Notification Server to configure the New Alert and Unread Alert Status notifications, the Alert Notification Client can write commands to the Alert Notification Control Point.

- Set the server to enable for specific category.
- Set the server to disable for specific category.
- Request the server to notify immediately.

4.10 Recovery from Connection Loss for New Alerts

When recovering from a connection loss, the Alert Notification Client shall write to the Alert Notification Control Point:

a) an “Enable New Alert Notification” command to enable ongoing notification of new alerts for each of the desired categories

   and

b) a “Notify New Alert Immediately” command with the Category ID field set to ‘0xff’ to get the current message counts (that may have been updated while the link was dropped).

The client shall not alert the user if the count for each of categories configured for reporting matches the count prior to connection loss.

4.11 Recovery from Connection Loss for Unread Alerts

When recovering from a connection loss, the Alert Notification Client shall write to the Alert Notification Control Point:

a) an “Enable Unread Alert Status Notification” command to enable ongoing notification of new alerts for each of the desired categories

   and

b) a “Notify Unread Alert Status Immediately” command with the Category ID field set to ‘0xff’ to get the current unread message counts (that may have been updated while the link was dropped).

The client shall not alert the user if the count for each of categories configured for reporting matches the count prior to connection loss.

4.12 Check Supported New Alert Category after Connection Setup

The Alert Notification Client shall execute procedure 4.3 to read the Supported New Alert Category characteristic after completing connection setup to discover any changes in the supported categories on the server side. The information read can be used to enable new functionality in the Alert Notification Client.

4.13 Check Supported Unread Alert Status Category after Connection Setup

The Alert Notification Client may execute procedure 4.4 to read the Supported Unread Alert Status Category characteristic after completing connection setup to discover any changes in the supported categories on the server side. The information read can be used to enable new functionality in the Alert Notification Client.

5 Connection Establishment

This section describes the connection establishment procedures used by an Alert Notification Server and Alert Notification Client. Since there are no topology restrictions imposed by this profile, the procedures are described in terms of GAP Peripheral Role (referred to as the Peripheral) and GAP Central Role (referred to as the Central).

5.1 GAP Peripheral Role Connection Establishment

5.1.1 Device Discovery

The Peripheral shall enter a GAP *Limited Discoverable Mode* when establishing an initial connection. The T_{GAP} (lim_adv_timeout) used during GAP *Limited Discoverable Mode* may be larger than the value specified in Section 16, Appendix A in the GAP specification [1], but the value shall be less than or equal to 180 seconds.

5.1.2 Connection Procedure for Unbonded Devices

This procedure is used for device discovery and connection establishment when the Peripheral connects to a Central to which it is not bonded. This procedure is initiated by user interaction (like activating the device by battery insertion).

It is recommended that the Peripheral advertises using the parameters in Table 5.1. The interval values in the first row are designed to attempt fast connection during the first 30 seconds; however, if a connection is not established within that time, the interval values in the second row are designed to reduce power consumption for devices that continue to advertise.

Advertising Duration	Parameter	Value
First 30 seconds (fast connection)	Advertising Interval	20 ms to 30 ms
After 30 seconds (reduced power)	Advertising Interval	1 s to 2.5 s

Table 5.1: Recommended Advertising Interval Values

The advertising interval and time to perform advertising should be configured with consideration for user expectations of connection establishment time.

The Peripheral shall accept any valid values for connection interval and connection latency set by the Central until service discovery, bonding and encryption setup is complete. Only after that should the Peripheral change to the preferred connection parameters that best suits the use case.

If a connection is not established within a time limit defined by the Peripheral, the Peripheral may exit the GAP connectable mode.

After bonding the Peripheral should write the *Bluetooth* address of the Central in the Peripheral controller's white list and set the Peripheral controller's advertising filter policy to 'process scan and connection requests only from devices in the White List'.

5.1.3 Connection Procedure for Bonded Devices

This procedure is used after the Peripheral has bonded with the Central device using the connection procedure in Section 5.1.2 when the user initiates a connection.

A Peripheral shall enter the GAP *Undirected Connectable Mode* when commanded by the user to initiate a connection to a Central device.

The Peripheral should use the advertising filter policy configured when bonded using the connection procedure in Section 5.1.2.

The Peripheral should use the recommended advertising interval values shown in Table 5.1.

The advertising interval and time to perform advertising should be configured with consideration for user expectations of connection establishment time.

The Peripheral shall accept any valid values for connection interval and connection latency set by the Central until service discovery and encryption setup is complete. Only after that should the Peripheral change to the preferred connection parameters that best suits its use case.

If a connection is not established within a time limit defined by the Peripheral, the Peripheral may exit the GAP connectable mode.

5.1.4 Link Loss Reconnection

When a connection is terminated due to link loss a Peripheral should attempt to reconnect to the Central by using the procedures described in sections 5.1.2 or.5.1.3.

5.2 GAP Central Role Connection Establishment

5.2.1 Device Discovery

The Central should use the GAP Limited Discovery Procedure to discover a Peripheral.

5.2.2 Connection Procedure for Unbonded Devices

This procedure is used for connection establishment when the Central connects to a Peripheral to which it is not bonded. This procedure is normally initiated by user interaction.

A Central may use one of the following GAP connection establishment procedures based on its connectivity requirements:

- *General Connection Establishment Procedure*. The Central may use this procedure when it requires connection to one or more Peripheral devices. This procedure allows a Central to connect to a Peripheral discovered during a scan without using the white list.
- *Direct Connection Establishment Procedure*. The Central may use this procedure when it requires connection to a single Peripheral.
- *Auto Connection Establishment Procedure*. The Central may use this procedure when it requires connection to one or more Peripheral devices. This procedure will automatically connect to a Peripheral in the white list.
- *Selective Connection Establishment Procedure*. The Central may use this procedure when it requires connection to one or more Peripheral devices. This procedure

allows a Central to connect to a Peripheral discovered during a scan while using the white list.

A Central should use the recommended scan interval and scan window values shown in Table 5.2. For the first 30 seconds (or optionally continuously for wall powered devices), the Central should use the first scan window / scan interval pair to attempt fast connection. However, if a connection is not established within that time, the Central should switch to one of the other scan window / scan interval options as defined below to reduce power consumption.

Scan Duration	Parameter	Value
First 30 seconds (fast connection)	Scan Interval	30 ms to 60 ms*
	Scan Window	30 ms
After 30 seconds (reduced power) - Option 1	Scan Interval	1.28 s
	Scan Window	11.25 ms
After 30 seconds (reduced power) - Option 2	Scan Interval	2.56 s
	Scan Window	11.25 ms

Table 5.2: Recommended Scan Interval and Scan Window Values

* A scan interval of 60ms is recommended when the Central is supporting other operations to provide a 50% scan duty cycle versus 100% scan duty cycle.

Option 1 in the table above uses the same background-scanning interval used in BR/EDR so the power consumption for LE will be similar to the power consumption used for background scanning on BR/EDR. Option 2 uses a larger background-scanning interval (e.g. twice as long) than used in BR/EDR so the power consumption for LE will be less than the power consumption used for background scanning on BR/EDR. Connection times during background scanning will be longer with Option 2.

After bonding, the Central should write the *Bluetooth* address of the Peripheral in the Central controller's white list and set the Central controller's initiator filter policy to 'process connectable advertisement packets'.

5.2.3 Connection Procedure for Bonded Devices

This procedure is used after the Central has bonded with the Peripheral using the connection procedure in Section 5.2.2 and the user initiates a connection.

A Central may use one of the following GAP connection establishment procedures based on its connectivity requirements:

- *General Connection Establishment Procedure*. The Central may use this procedure when it requires connection to one or more Peripheral devices. This procedure allows a Central to connect to a Peripheral discovered during a scan without using the white list.
- *Direct Connection Establishment Procedure*. The Central may use this procedure when it requires connection to a single Peripheral.
- *Auto Connection Establishment Procedure*. The Central may use this procedure when it requires connection to one or more Peripheral devices. This procedure will automatically connect to a Peripheral in the white list.
- *Selective Connection Establishment Procedure*. The Central may use this procedure when it requires connection to one or more Peripheral devices. This procedure

allows a Central to connect to a Peripheral discovered during a scan while using the White List.

The Central should use the recommended scan interval and scan window values shown in Table 5.2. For the first 30 seconds (or optionally continuously for wall powered devices), the Central should use the first scan window / scan interval pair to attempt fast connection. However, if a connection is not established within that time, the Central should switch to one of the other scan window / scan interval options as defined below to reduce power consumption.

The Central should use a scan window and scan interval suitable to its power and connection time requirements. Increasing the scan window increases the power consumption, but decreases the connection time.

The scan interval and scan window should be configured with consideration for user expectations of connection establishment time.

The Central shall start encryption after each connection creation to verify the status of the bond. If encryption fails upon connection establishment (i.e., the bond no longer exists), the Central must, after user interaction, re-bond, perform service discovery (unless the Central had previously determined that the Peripheral did not have the «Service Changed» characteristic) and reconfigure the Peripheral before using any of the services referenced by this profile in case the configuration was altered or lost.

5.2.4 Link Loss Reconnection

When a connection is terminated due to link loss a Central should attempt to reconnect to the Peripheral using any of the GAP connection procedures and using procedures described in sections 5.2.2 or 5.2.3.

5.2.5 Fast Connection Interval

To avoid very long service discovery and encryption setup times, the Central should use the connection intervals defined in Table 5.3 in the connection request.

Parameter	Value
Minimum Connection Interval	50 ms
Maximum Connection Interval	70 ms

Table 5.3: Recommended connection interval values

At any time a key refresh or encryption setup is required, for example to perform key refresh, this should be preceded with a connection parameter update to the minimum and maximum connection interval values in Table 5.3 and a latency of zero. This fast connection interval should be maintained as long as low latency is required. After that, it should switch to the preferred connection parameters as decided by the Peripheral using the GAP *Connection Parameter Update* procedure.

6 Security Considerations

This section describes the security requirements for an Alert Notification Client and Alert Notification Server. Since there are no topology restrictions imposed by this profile, the requirements are described in terms of GAP Peripheral Role (referred to as the Peripheral) and GAP Central Role (referred to as the Central).

The Peripheral shall support LE Security Mode 1 and Security Levels 2 or 3. The Peripheral should use the SM Slave Security Request procedure only when bonded with the Central to inform the Central of its security requirements.

The Central shall support LE Security Mode 1 and Security Levels 2 and 3. The Central should accept the LE Security Mode and Security Level combination requested by the Peripheral.

7 GATT Interoperability Requirements

The following GATT sub-procedures are required to be implemented by the Alert Notification Client.

GATT Sub-Procedure	Alert Notification Client
Discover All Primary Services	C.1
Discover Primary Services by Service UUID	C.1
Discover All Characteristics of a Service	C.2
Discover Characteristics by UUID	C.2
Discover All Characteristics Descriptors	M
Read Characteristic Value	M
Write Characteristic value	M
Write Characteristic Descriptors	M
Read Characteristic Descriptors	O
Notification	M
C.1: The Alert Notification Client shall support either the Discover All Primary Services sub-procedure or the Discover Primary Services by Service UUID sub-procedure.	
C.2: The Alert Notification Client shall support either the Discover All Characteristics of a Service sub-procedure or the Discover Characteristics by UUID sub-procedure.	

8 Acronyms and Abbreviations

Acronyms and Abbreviations	Meaning
BR/EDR	Basic Rate / Enhanced Data Rate
GAP	Generic Access Profile
GATT	Generic Attribute Profile
LE	Low Energy
SM	Security Manager
UI	User Interface
UUID	Universally Unique Identifier

9 References

[1] *Bluetooth* Core Specification v4.0