

# RW BLE Alert Notification Profile Interface Specification

---

Interface Specification

RW-BLE-ANP-IS

Version 8.0

2015-07-29

---

## Revision History

Version	Date	Revision Description	Author
0.1	April 16 <sup>th</sup> 2013	Initial draft	LT
1.0	June 24 <sup>th</sup> 2013	Initial release	LT
1.1	July 17, 2014	Few updates, supply missing tables Some corrections, clarifications, formats	KY
7.0	November 28 <sup>th</sup> , 2014	Updated for BLE 4.1	CM
8.0	July 29 <sup>th</sup> , 2015	Updated for BLE 4.2	CM



## Table of Contents

1	Overview .....	4
1.1	Document Overview .....	4
1.2	BLE Alert Notification Profile Overview .....	4
2	ANP Server Role API .....	7
2.1	Environment .....	7
2.2	API Messages .....	7
2.2.1	Initialization/Database creation .....	7
2.2.2	ANPS_ENABLE_REQ .....	8
2.2.3	ANPS_ENABLE_RSP .....	8
2.2.4	ANPS_NTF_ALERT_CMD .....	9
2.2.5	ANPS_NTF_IMMEDIATE_REQ_IND .....	9
2.2.6	ANPS_NTF_STATUS_UPDATE_IND .....	10
2.2.7	ANPS_CMP_EVT .....	10
3	ANP Client Role API .....	11
3.1	Environment .....	11
3.2	API Messages .....	11
3.2.1	Initialization .....	11
3.2.2	ANPC_ENABLE_REQ .....	11
3.2.3	ANPC_ENABLE_RSP .....	12
3.2.4	ANPC_READ_CMD .....	13
3.2.5	ANPC_WRITE_CMD .....	14
3.2.6	ANPC_VALUE_IND .....	14
3.2.7	ANPC_CMP_EVT .....	15
4	Miscellaneous .....	16
5	Abbreviations .....	18
6	References .....	19



## 1 Overview

### 1.1 Document Overview

This document describes the non-standard interface of the RivieraWaves (RW) Bluetooth Low Energy (BLE) Alert Notification Profile (ANP) implementation. Along this document, the interface messages will be referred to as API messages for the profile block(s).

Their description will include their utility and reason for implementation for a better understanding of the user and the developer that may one day need to interface them from a higher application.

### 1.2 BLE Alert Notification Profile Overview

The ANP allows a client device (a watch for instance) to be alerted when another device (basically a smartphone) receives a SMS, a MMS, an email ... It provides information about the number of new messages for a given category and also the number of unread messages for these categories.

This service has been implemented as a profile. Within this profile, two roles can be supported: Server role (ANPS) and Client role (ANPC). The Client role must support the GAP Peripheral Role and the Server role, the GAP Central role. The profile requires a connection to be established between the two devices for its functionality.

The various documents edited by the Bluetooth SIG present different use cases for this profile, their GATT, GAP and security, mandatory and optional requirements. The Alert Notification Profile specifications have been adopted by the Bluetooth SIG on September 15th 2011 ([1] and [3]). For a better understanding, we highly recommend the user to read these specifications before reading this IS. Their related Test Specifications have been released at the same time and are referenced in [2] and [4].

The profile is implemented in the RW-BLE software stack as two tasks, one for each role. Each task has an API decided after the study of the profile specifications and test specifications, and it is considered to be minimalistic and designed for a future application which would combine the profile functionality with the device connectivity and security procedures.



The structure of the Alert Notification service is defined in the table below:

Characteristic Name	Requirements	Properties	Security	Descriptors
<b>Supported New Alert Category</b>	Mandatory	Read	None	None
<b>New Alert</b>	Mandatory	Notify	None	Client Characteristic Configuration
<b>Supported Unread Alert Category</b>	Mandatory	Read	None	None
<b>Unread Alert Status</b>	Mandatory	Notify	None	Client Characteristic Configuration
<b>Alert Notification Control Point</b>	Mandatory	Write	None	None

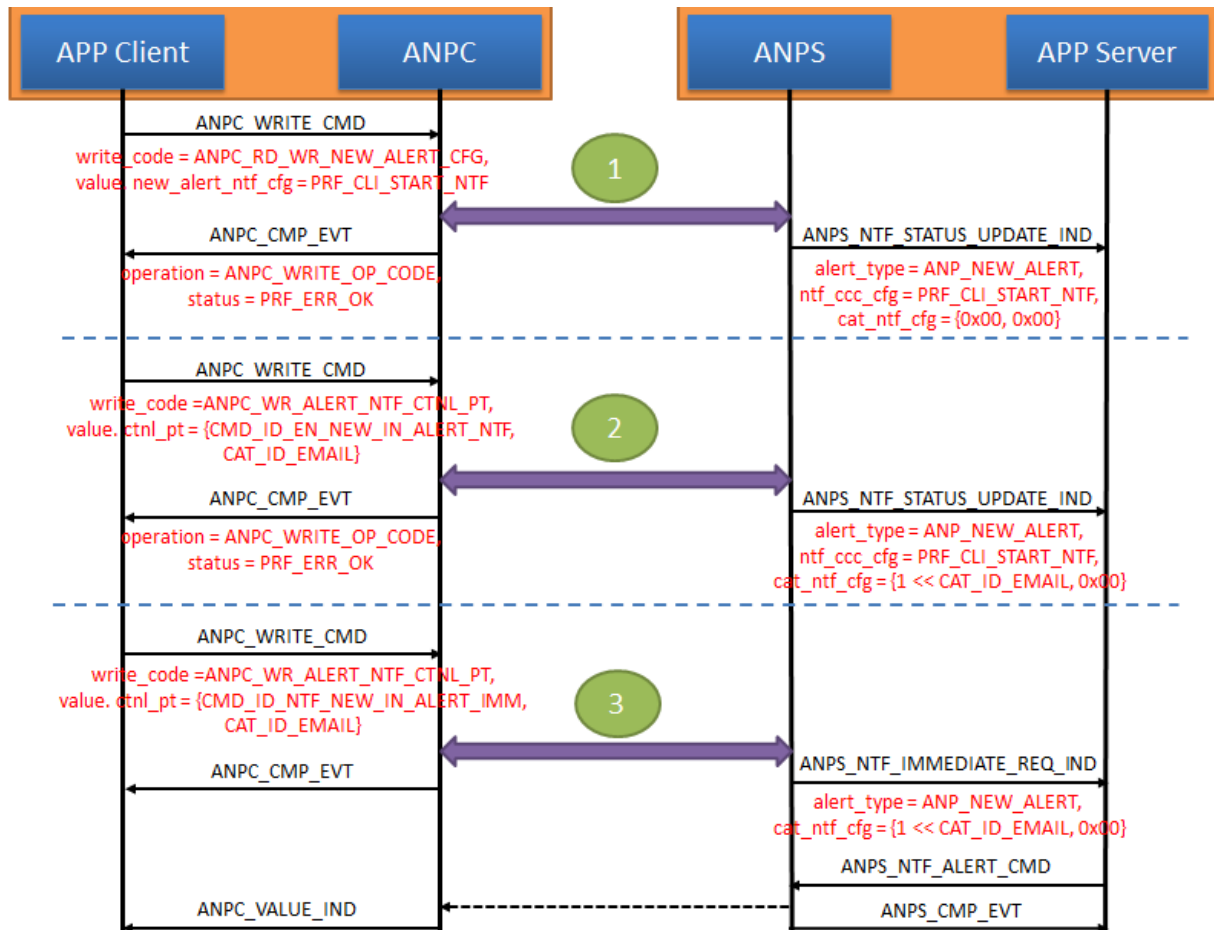
The Supported New Alert Category and the Supported Unread Alert Category characteristics expose the categories that are supported by the server device.

The concept of category is used to identify the type of alert information:

- Simple Alert
- Email
- News
- Call
- Missed Call
- SMS/MMS
- Voice Mail
- Schedule
- High Prioritized Alert
- Instant Message

The Client Characteristic Configuration descriptor shall be written with the value 0x0001 (PRF\_CLI\_START\_NTF) to enable or with the value 0x0000 (PRF\_CLI\_STOP\_NTFIND) to disable sending of notifications.

The Alert Notification Control Point characteristic shall be used to enable or disable sending of notifications for one of the supported categories. It also allows requesting the server device to notify immediately the current number of new alerts or unread alerts for a specified category. The following MSC shows which message are received at an application level when a central device tries to enable sending of notification for a category:



Phase 1 allows the client to configure the Client Characteristic Configuration descriptor of the New Alert Characteristic so that the server can send notifications for this characteristic. At this point the server is still not able to send a notification because the Alert Notification profile requires that the client also enable sending of notification for each category.

On Phase 2, the client writes the Control Point characteristic to enable sending of notification. In this example, the E-Mail category is enabled.

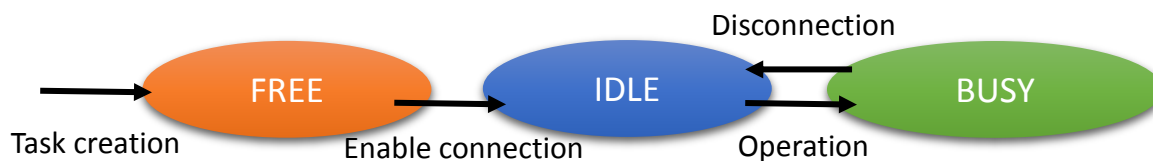
On Phase 3, the client writes to the Control Point characteristic the type of alert notification for the category. In this example, the value set is immediate alert.

## 2 ANP Server Role API

### 2.1 Environment

Within the ANPS task, four states are defined: DISABLED, IDLE, CONNECTED and BUSY.

The busy state is used when a procedure is currently being processed by a connected device (read, write, ...). When the state is busy, no command message sent by a higher layer can be handled, this message will be stored until the end of the procedure and handled once the procedure is over. Thus it can be considered as a connected state from an application point of view.



**Important Note:** The TASK\_ANPS task is multi-instantiated, one instance is created for each connection for which the profile will be enabled and each of these instances will have a different task ID. Thus, it is very important for the application to keep the source task ID of the first received ANPS\_CMP\_EVT message to be able to communicate with the peer device linked to this task ID once it has been enabled.

The term TASK\_ANPS\_IDX will be used in the rest of the document to refer to any instance of the Alert Notification profile Server Role Task. The term TASK\_ANPS will refer to the first instance of this task.

### 2.2 API Messages

#### 2.2.1 Initialization/Database creation

During the initialization phase of the Alert Notification Sensor, the memory for this task must be allocated using the message GAPM\_PROFILE\_TASK\_ADD\_CMD provided by the GAPM interface. Apart from the security level, the following parameters should be filled:

Type	Parameters	Description
struct anp_cat_id_bit_mask	supp_new_alert_cat	Supported New Alert Category Characteristic Value - Shall not be 0 (see cat_id_mask_0 shall be used for the categories: Simple Alert (0) to Schedule (7). cat_id_mask_1 shall be used for the High Priority Alert and Instant Message categories. Mask provided in Table 1 and Table 2 shall be used to set the supported values.  Table 5
struct anp_cat_id_bit_mask	supp_unread_alert_cat	Supported Unread Alert Category Characteristic Value - Can be 0 (see cat_id_mask_0 shall be used for the categories: Simple Alert (0) to Schedule (7). cat_id_mask_1 shall be used for the High Priority Alert and Instant Message categories. Mask provided in Table 1 and Table 2 shall be used to set the supported values.



		Table 5
--	--	---------

Response: GAPM\_PROFILE\_TASK\_ADDED\_IND

Description: This API message shall be used to add one instance of the Alert Notification Service.

The security level is very important because it allows the application to modulate the protection of the attributes related to the profile. The implementation only allows modulation of the read/write/notify permissions (not properties) of the characteristic values. (if unauthenticated is requested, then a Read/Write to a characteristic won't be allowed if the link between the devices is not unauthenticated level of security.

The PERM(access, right) macro that can be found in the attm.h file shall be used to set the security level. The access value shall be set to PERM\_ACCESS\_SVC (0). The right value will set the minimum required security level:

```
/// Enable access
PERM_RIGHT_ENABLE = 1,
/// Access Requires Unauthenticated link
PERM_RIGHT_UNAUTH = 2,
/// Access Requires Authenticated link
PERM_RIGHT_AUTH = 3
```

### 2.2.2 ANPS\_ENABLE\_REQ

Source: TASK\_APP

Destination: TASK\_ANPS

Required State: IDLE

Parameters:

Type	Parameters	Description
uint16_t	new_alert_ntf_cfg	New Alert Characteristic - Saved Client Characteristic Configuration Descriptor Value for a bonded device.
uint16_t	unread_alert_status_ntf_cfg	Unread Alert Status Characteristic - Saved Client Characteristic Configuration Descriptor Value for a bonded device.

Response: ANPS\_ENABLE\_RSP

Description: This API message shall be used after the connection with a peer device has been established in order to restore the bond data used for that connection.

### 2.2.3 ANPS\_ENABLE\_RSP

Source: TASK\_ANPS

Destination: TASK\_APP

Parameters:

Type	Parameters	Description
uint8_t	status	Status of the operation

Description: This API message informs the application about the result of the enabling operation.



#### 2.2.4 ANPS\_NTF\_ALERT\_CMD

Source: TASK\_APP

Destination: TASK\_ANPS\_IDX

Required State: IDLE

Parameters:

Type	Parameters	Description
uint8_t	operation	Operation code, indicate which characteristic value need to be updated: <ul style="list-style-type: none"><li>ANPS_UPD_NEW_ALERT_OP_CODE</li><li>ANPS_UPD_UNREAD_ALERT_STATUS_OP_CODE</li></ul>
union	value	
struct anp_new_alert	new_alert	New Alert
struct anp_unread_alert	unread_alert_status	Unread Alert Status

Response: ANPS\_CMP\_EVT

Description: This API message shall be used by the application to send a New Alert or an Unread Alert Status notification to a specific device.

The implementation of the profile takes care of the provided value (category id, ...), if one of these is not within the ranges defined by the profile specification, an ANPS\_CMP\_EVT message with an PRF\_ERR\_INVALID\_PARAM status is sent to the application.

It also checks whether the provided category has been set as supported and if the peer device has enable sending of notifications for this category and for the characteristic to update. If no notification can be sent, the received status will be PRF\_ERR\_NTF\_DISABLED.

In the case where a New Alert value is sent, this message shall be allocated dynamically using the KE\_ALLOC\_MSG\_DYN macro because the length of the information string can be different between messages.

#### 2.2.5 ANPS\_NTF\_IMMEDIATE\_REQ\_IND

Source: TASK\_ANPS\_IDX

Destination: TASK\_APP

Parameters:

Type	Parameters	Description
uint8_t	alert_type	Alert Type: <ul style="list-style-type: none"><li>ANP_NEW_ALERT</li><li>ANP_UNREAD_ALERT</li></ul>
struct anp_cat_id_bit_mask	cat_ntf_cfg	Categories to notify

Description: This message is sent to the application when the Alert Notification Control Point Characteristic value is written by the peer device with a 'Notify New Incoming Alert Immediately' or 'Notify Unread Category Status Immediately' command id.

The cat\_ntf\_cfg parameter provided information about the categories that has been required by the peer device and which can be notified (supported + enabled).

If there are no new alerts or no unread alerts for the specified category, the provided number of alert shall be set to 0.

**Note:** As the ANP Server role has been design to be used by devices such as smartphone, tablets ... which have a higher amount of memory, it is application implementer responsibility to take care about the management of the number of alerts. The profile doesn't store the previously sent values and thus will be sent every data received from an application.

## 2.2.6 ANPS\_NTF\_STATUS\_UPDATE\_IND

Source: TASK\_ANPS\_IDX

Destination: TASK\_APP

Parameters:

Type	Parameters	Description
uint8_t	alert_type	Alert Type: <ul style="list-style-type: none"><li>ANP_NEW_ALERT</li><li>ANP_UNREAD_ALERT</li></ul>
uint16_t	ntf_ccc_cfg	Client Characteristic Configuration State
struct anp_cat_id_bit_mask	cat_ntf_cfg	Status for each category

Description: This message is sent to the application when the value of one of the two Client Characteristic Configuration descriptors has been written by the peer device to enable or disable sending of notifications. It is also sent upon reception of a write request for the Alert Notification Control Point characteristic with the following command ids:

- Enable New Incoming Alert Notifications
- Enable Unread Category Status Notifications
- Disable New Incoming Alert Notifications
- Disable Unread Category Status Notifications

## 2.2.7 ANPS\_CMP\_EVT

Source: TASK\_ANPS\_IDX

Destination: TASK\_APP

Parameters:

Type	Parameters	Description
uint8_t	operation	Operation Code: <ul style="list-style-type: none"><li>ANPS_UPD_NEW_ALERT_OP_CODE</li><li>ANPS_UPD_UNREAD_ALERT_STATUS_OP_CODE</li></ul>
uint8_t	status	Status

Description: The API message is used by the ANPS task to inform the sender of a command that the procedure is over and contains the status of the procedure.



## 3 ANP Client Role API

### 3.1 Environment

Within the ANPC task, four states are defined: FREE, IDLE, DISCOVERING and BUSY.

**Important Note:** The TASK\_ANPC task is multi-instantiated, one instance is created for each connection for which the profile will be enabled and each of these instances will have a different task ID. Thus, it is very important for the application to keep the source task ID of the first received ANPC\_CMP\_EVT message to be able to communicate with the peer device linked to this task ID once it has been enabled.

The term TASK\_ANPC\_IDX will be used in the rest of the document to refer to any instance of the Alert Notification profile Client Role Task. The term TASK\_ANPC will refer to the first instance of this task.

### 3.2 API Messages

#### 3.2.1 Initialization

During the initialization phase of the Alert Notification Collector, the memory for this task must be allocated using the message GAPM\_PROFILE\_TASK\_ADD\_CMD provided by the GAPM interface.

#### 3.2.2 ANPC\_ENABLE\_REQ

Source: TASK\_APP

Destination: TASK\_ANPC\_IDX

Required State: IDLE

Parameters:

Type	Parameters	Description
uint8_t	con_type	Connection Type
struct anp_cat_id_bit_mask	new_alert_enable	New Alert Category to Enable for Notifications
struct anp_cat_id_bit_mask	unread_alert_enable	Unread Alert Category to Enable for Notifications
struct anpc_ans_content	ans	Service structure previously discovered in the database of the peer device.

Response: ANPC\_ENABLE\_RSP

Description: This API message is used for enabling the Client role of the ANP. This Application message contains the connection type and the previously saved discovered ANS details on peer.

The connection type may be PRF\_CON\_DISCOVERY (0x00) for discovery/initial configuration or PRF\_CON\_NORMAL (0x01) for a normal connection with a bonded device. Application shall save the information to reuse them for other connections. During normal connection, previously discovered device information can be reused.

For a normal connection, the response to this request is sent right away after saving the ANS content in the environment and registering ANPC in GATT to receive the notifications for the known attribute handles in ANS that would be notified.

For a discovery connection, discovery of the peer ANS is started and the response will be sent at the end of the discovery with the discovered attribute details.

Once the content of the peer ANS has been stored, the ANPC will automatically read the value of the Supported New Alert Category and the Supported Unread Status Category Characteristics as required by the Alert Notification Profile specification.

The figure below exposes the different messages exchanged between application layer and the ANPC task after a connection has been established with a peer device.

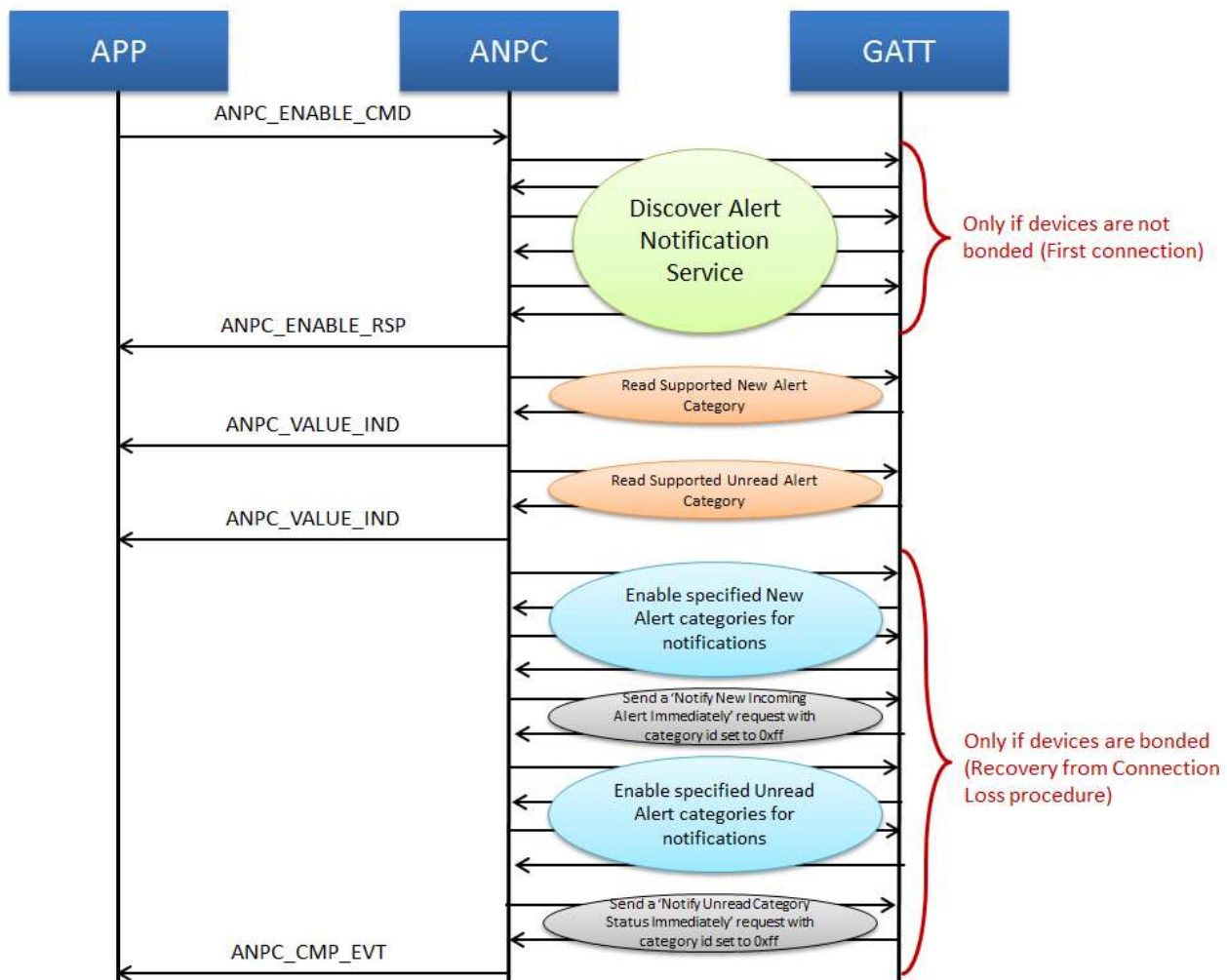


Figure 1 – Connection Establishment procedure

In the case where the profile is enabled for a bonded device, the Recovery from Connection Loss Procedure shall occur. In this procedure the client device shall enable notifications for every desired category and then send a request so that it can be notified about all alert status.

### 3.2.3 ANPC\_ENABLE\_RSP

Source: TASK\_ANPC\_IDX

Destination: TASK\_APP

Parameters:

Type	Parameters	Description
uint8_t	status	Status of the operation.
struct anpc_ans_content	ans	Service structure previously discovered in the database of the peer device

Description: The API message informs the Application about the result of the enabling operation.



### 3.2.4 ANPC\_READ\_CMD

Source: TASK\_APP

Destination: TASK\_ANPC\_IDX

Parameters:

Type	Parameters	Description
uint8_t	operation	Operation Code will be set by the profile task.
uint8_t	read_code	Read Code: <ul style="list-style-type: none"> <li>ANPC_RD_WR_NEW_ALERT_CFG</li> <li>ANPC_RD_WR_UNREAD_ALERT_STATUS_CFG</li> </ul>

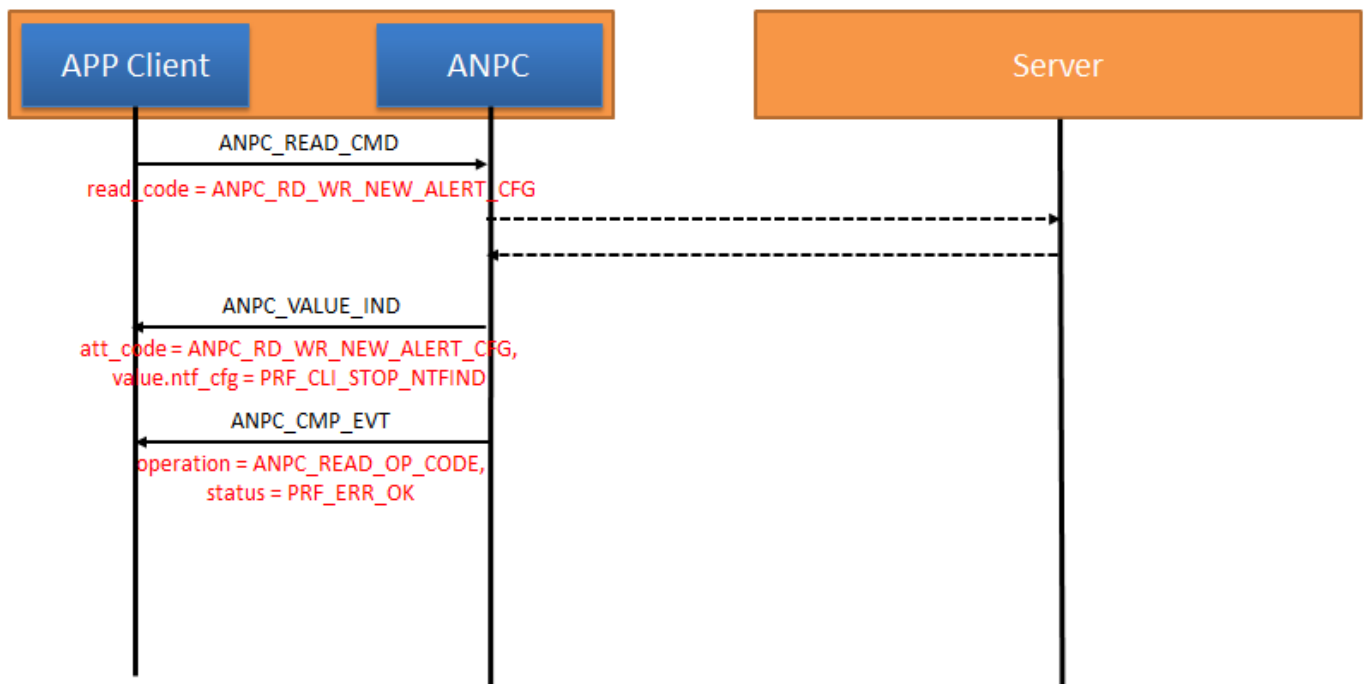
Response: ANPC\_VALUE\_IND and ANPC\_CMP\_EVT

Description: The API message shall be used to read the value of an attribute in the peer device database.

Please note that this command only allows reading the value of the Client Characteristic Configuration descriptors (New Alert characteristic or Unread Alert Status characteristic).

The value of the Supported New Alert Category characteristic and the Supported Unread Alert Category characteristics are not supposed to change during a connection and then are read once during the discovery procedure (see 1.2).

The scheme below gives an example of the message exchanged upon a read procedure:





### 3.2.5 ANPC\_WRITE\_CMD

Source: TASK\_APP

Destination: TASK\_ANPC\_IDX

Required State: IDLE

Parameters:

Type	Parameters	Description
uint8_t	operation	Operation Code will be set by the profile task.
uint8_t	write_code	Write Code: <ul style="list-style-type: none"> <li>ANPC_WR_ALERT_NTF_CTLN_PT</li> <li>ANPC_RD_WR_NEW_ALERT_CFG</li> <li>ANPC_RD_WR_UNREAD_ALERT_STATUS_CFG</li> </ul>
union	value	
struct anp_ctln_pt	ctl_n_pt	Alert Notification Control Point characteristic value
uint16_t	new_alert_ntf_cfg	New Alert characteristic client characteristic configuration value
uint16_t	unread_alert_status_ntf_cfg	Unread Alert Status characteristic client characteristic configuration value

Response: ANPC\_CMP\_EVT

Description: This API message shall be used by the application to write the value of one of the writable attribute in the peer device database (see 1.2).

### 3.2.6 ANPC\_VALUE\_IND

Source: TASK\_ANPC\_IDX

Destination: TASK\_APP

Parameters:

Type	Parameters	Description
uint8_t	att_code	Attribute Code: <ul style="list-style-type: none"> <li>ANPC_RD_SUP_NEW_ALERT_CAT</li> <li>ANPC_RD_SUP_UNREAD_ALERT_CAT</li> <li>ANPC_RD_WR_NEW_ALERT_CFG</li> <li>ANPC_RD_WR_UNREAD_ALERT_STATUS_CFG</li> </ul>
union	value	
struct anp_cat_id_bit_mask	supp_cat	List of supported categories
struct anp_new_alert	new_alert	New Alert
struct anp_unread_alert	unread_alert	Unread Alert
uint16_t	ntf_cfg	Client Characteristic Configuration Descriptor Value

Description: This API is sent to the application once an attribute value has been received from the peer device upon a notification or a read response message.



### 3.2.7 ANPC\_CMP\_EVT

Source: TASK\_ANPC\_IDX

Destination: TASK\_APP

Parameters:

Type	Parameters	Description
uint8_t	operation	Operation Code: <ul style="list-style-type: none"><li>ANPC_ENABLE_OP_CODE</li><li>ANPC_READ_OP_CODE</li><li>ANPC_WRITE_OP_CODE</li></ul>
uint8_t	status	Status

Description: The API message is used by the ANPC task to inform the sender of a command that the procedure is over and contains the status of the procedure.

## 4 Miscellaneous

Name	Value	Description
ANP_CAT_ID_SPL_ALERT_SUP	0x01	Simple Alert
ANP_CAT_ID_EMAIL_SUP	0x02	Email
ANP_CAT_ID_NEWS_SUP	0x04	News
ANP_CAT_ID_CALL_SUP	0x08	Call
ANP_CAT_ID_MISSED_CALL_SUP	0x10	Missed Call
ANP_CAT_ID_SMS_MMS_SUP	0x20	SMS/MMS
ANP_CAT_ID_VOICE_MAIL_SUP	0x40	Voice Mail
ANP_CAT_ID_SCHEDULE_SUP	0x80	Schedule

Table 1 – Supported Categories Bit Mask 0 Flags

Name	Value	Description
ANP_CAT_ID_HIGH_PRTY_ALERT	0x01	High Priority Alert
ANP_CAT_ID_INST_MSG	0x02	Instant Message

Table 2 – Supported Categories Bit Mask 1 Flags

Name	Value	Description
CAT_ID_SPL_ALERT	0	Simple Alert
CAT_ID_EMAIL	1	Email
CAT_ID_NEWS	2	News
CAT_ID_CALL	3	Call
CAT_ID_MISSED_CALL	4	Missed Call
CAT_ID_SMS_MMS	5	SMS/MMS
CAT_ID_VOICE_MAIL	6	Voice Mail
CAT_ID_SCHEDULE	7	Schedule
CAT_ID_HIGH_PRTY_ALERT	8	High Priority Alert
CAT_ID_INST_MSG	9	Instant Message
CAT_ID_ALL_SUPPORTED_CAT	255	All supported categories

Table 3 – Category IDs Keys

Name	Value	Description
CMD_ID_EN_NEW_IN_ALERT_NTF	0	Enable New Incoming Alert Notification
CMD_ID_EN_UNREAD_CAT_STATUS_NTF	1	Enable Unread Category Status Notification
CMD_ID_DIS_NEW_IN_ALERT_NTF	2	Disable New Incoming Alert Notification
CMD_ID_DIS_UNREAD_CAT_STATUS_NTF	3	Disable Unread Category Status Notification
CMD_ID_NTF_NEW_IN_ALERT_IMM	4	Notify New Incoming Alert immediately
CMD_ID_NTF_UNREAD_CAT_STATUS_IMM	5	Notify Unread Category Status immediately

Table 4 - Command IDs Keys

Type	Parameters	Description
uint8_t	cat_id_mask_0	Category ID Bit Mask 0
uint8_t	cat_id_mask_1	Category ID Bit Mask 1

cat\_id\_mask\_0 shall be used for the categories: Simple Alert (0) to Schedule (7). cat\_id\_mask\_1 shall be used for the High Priority Alert and Instant Message categories. Mask provided in Table 1 and Table 2 shall be used to set the supported values.

Table 5 - Alert Category ID Bit Mask Structure (struct anp\_cat\_id\_bit\_mask)



Type	Parameters	Description
uint8_t	info_str_len	Information String Length
uint8_t	cat_id	Category ID
uint8_t	nb_new_alert	Number of alerts
uint8_t	str_info[1]	Text String Information

Table 6 - New Alert Characteristic Value Structure (struct anp\_new\_alert)

Type	Parameters	Description
uint8_t	cat_id	Category ID
uint8_t	nb_unread_alert	Number of alerts

Table 7 - Unread Alert Characteristic Value Structure (struct anp\_unread\_alert)

Type	Parameters	Description
uint8_t	cmd_id	Command ID
uint8_t	cat_id	Category ID

Table 8 - Alert Notification Control Point Characteristic Value Structure (struct anp\_ctl\_pt)

Type	Parameters	Description
struct prf_svc	svc	Service information
struct prf_char_inf	chars[ANPC_CHAR_MAX]	Profile characteristic information
struct prf_char_desc_inf	desc[ANPC_DESC_MAX]	Profile descriptor information

Table 9 - Alert Notification Content Structure (struct anpc\_ans\_content)

Type	Parameters	Description
U16	shdl	Start handle of the HID Service.
U16	ehdl	End handle of the HID Service.

Table 10 - Service description structure (struct prf\_svc)

Type	Parameters	Description
U16	char_hdl	Characteristic declaration attribute handle.
U16	val_hdl	Characteristic value attribute handle.
U8	prop	Properties
U8	char_ehdl_off	Number of attribute within the Characteristic.

Table 11 - Characteristic description structure (struct prf\_char\_inf)

Type	Parameters	Description
U16	desc_hdl	Descriptor attribute handle

Table 12 - Descriptor description structure (struct prf\_char\_desc\_inf)



## 5 Abbreviations

Abbreviation	Original Terminology
API	Application Programming Interface
BLE	Bluetooth Low Energy
GAP	Generic Access Profile
GATT	Generic Attribute Profile
ANP	Alert Notification Profile
ANPS	Alert Notification Server Role
ANPC	Alert Notification Client Role
ANS	Alert Notification Service
RW	RivieraWaves SAS

## 6 References

<b>[1]</b>	<b>Title</b>	ALERT NOTIFICATION PROFILE SPECIFICATION		
	<b>Reference</b>	ANP_SPEC_V10		
	<b>Version</b>	V10r00	<b>Date</b>	2011-09-15
	<b>Source</b>	Bluetooth SIG		

<b>[2]</b>	<b>Title</b>	ALERT NOTIFICATION PROFILE TEST SPECIFICATION		
	<b>Reference</b>	ANPTS.1.0.0		
	<b>Version</b>	1.0.0	<b>Date</b>	2011-09-15
	<b>Source</b>	Bluetooth SIG		

<b>[3]</b>	<b>Title</b>	ALERT NOTIFICATION SERVICE SPECIFICATION		
	<b>Reference</b>	ANPS_SPEC_V10		
	<b>Version</b>	V10r00	<b>Date</b>	2011-09-15
	<b>Source</b>	Bluetooth SIG		

<b>[4]</b>	<b>Title</b>	ALERT NOTIFICATION SERVICE TEST SPECIFICATION		
	<b>Reference</b>	ANPS.TS.1.0.0		
	<b>Version</b>	1.0.0	<b>Date</b>	2011-09-15
	<b>Source</b>	Bluetooth SIG		