

Android 5.0 Bluetooth HCI Requirements

Contents

- [1. Introduction](#)
- [2. General Design Overview](#)
 - [2.1 Chip Capabilities and Configuration](#)
- [3. Vendor-Specific Capabilities](#)
- [4. Multi-advertiser Support](#)
 - [4.1 LE_Multi_Advt_Command](#)
 - [4.1.1 LE_Multi_Advt_Command: Set_Advt_Param_Multi_Sub_Cmd](#)
 - [4.1.2 LE_Multi_Advt_Command: Set_Advt_Data_Multi_Sub_Cmd](#)
 - [4.1.3 LE_Multi_Advt_Command: Set_Scan_Resp_Data_Multi_Sub_Cmd](#)
 - [4.1.4 LE_Multi_Advt_Command: Set_Random_Addr_Multi_Sub_Cmd](#)
 - [4.1.5 LE_Multi_Advt_Command: Set_Advt_Enable_Multi_Sub_Cmd](#)
- [5. Offloaded Resolution of Private Address](#)
 - [5.1 LE_RPA_offload_Command](#)
 - [5.1.1 LE_RPA_offload: Enable_cust_specific_sub_Command](#)
 - [5.1.2 LE_RPA_offload: Add_IRK_to_list_sub_Command](#)
 - [5.1.3 LE_RPA_offload: Remove_IRK_to_list_sub_Command](#)
 - [5.1.4 LE_RPA_offload: Clear_IRK_list_sub_Command](#)
 - [5.1.5 LE_RPA_offload: Read_IRK_list_sub_Command](#)
- [6. Batching of Scan Results](#)
 - [6.1 LE_Batch_Scan_Command](#)
 - [6.1.1 LE_Batch_Scan_Command: Enable Customer Specific feature](#)
 - [6.1.2 LE_Batch_Scan_Command: Set Batch Scan Storage Param sub-command](#)
 - [6.1.3 LE_Batch_Scan_Command: Set Batch Scan Param sub-command](#)
 - [6.1.4 LE_Batch_Scan_Command: Read Batch Scan Results sub-command](#)
- [7. Advertisement Packet Content filter](#)
 - [7.1 LE_APCF_Command](#)
 - [7.1.1 LE_APCF_Command: Enable_sub_cmd](#)
 - [7.1.2 LE_APCF_Command: set_filtering_parameters_sub_cmd](#)
 - [7.1.3 LE_APCF_Command: broadcast_address_sub_cmd](#)
 - [7.1.4 LE_APCF_Command: service_uuid_sub_cmd](#)
 - [7.1.5 LE_APCF_Command: solicitation_uuid_sub_cmd](#)
 - [7.1.6 LE_APCF_Command: local_name_sub_cmd](#)
 - [7.1.7 LE_APCF_Command: manf_data_sub_cmd](#)
 - [7.1.8 LE_APCF_Command: service_data_sub_cmd](#)
- [8. Controller Activity and Energy Information Command](#)
- [9. HCI Event \[Vendor-specific\]](#)

[9.1 Storage Threshold Breach Sub-event](#)

[9.2 LE Multi-Advertising State Change Sub-event](#)

© 2015 Google, Inc. All Rights Reserved. No express or implied warranties are provided for herein. All specifications are subject to change and any expected future products, features or functionality will be provided on an if and when available basis.

1. Introduction

The Host Controller Interface (HCI) is used for interacting with a Bluetooth controller.

This document provides a list of Bluetooth (BT) and Bluetooth Low Energy (BLE) requirements intended for the Android 5.0 platform. The aim of this document is for Host BT stack vendors and BT controller vendors to conform to these requirements for the platform, in order to use the feature set described below.

The Bluetooth Core 4.1 Specification, referred to in this document as the "BT 4.1 core specification," is available on the [Bluetooth SIG website](#) along with other adopted documents.

2. General Design Overview

2.1 Chip Capabilities and Configuration

Android, as an open platform, has a matrix of software releases, OEMs, vendors, and platform and chip capabilities.

To manage the varying landscape and to manage migrations, a design philosophy of allowing BT controllers to expose their capabilities (beyond the standard BT 4.1 core specification) is described in this document. The host BT stack can then use these capabilities to determine which features to enable.

2.2 Supporting Open Standards

One goal of Android Lollipop is support for open standards after ratification in a Bluetooth specification. If a feature described below becomes available in standard HCI methods in a future Bluetooth specification, we will lean towards making that approach the default.

3. Vendor-Specific Capabilities

Vendor specific command: `LE_Get_Vendor_Capabilities_Command`

OCF (OpCode Command Field): 0x153

Command Parameter	Size	Purpose
	NA	Empty command parameter list

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status

<code>max_advt_instances</code>	1 octet	Number of advertisement instances supported
<code>offloaded_resolution_of_private-address</code>	1 octet	BT chip capability of RPA; if supported by a chip, it needs enablement by the host. 0 = Not capable 1 = Capable
<code>total_scan_results_storage</code>	2 octets	Storage for scan results in bytes
<code>max_irk_list_sz</code>	1 octet	Number of IRK entries supported in the firmware
<code>filtering_support</code>	1 octet	Support for filtering in the controller. 0 = Not supported 1 = Supported
<code>max_filter</code>	1 octet	Number of filters supported
<code>activity_energy_info_support</code>	1 octet	Supports reporting of activity and energy information. 0 = Not capable 1 = Capable

The `max_advt_instances` parameter represents the total advertisement instances in the controller. The range of `advt_instance` IDs will be 0 to the following:
`max_advt_instances-1`

An advertisement instance with an ID equal to 0 will map to an existing (default/standard) HCI instance. When operating on a default/standard HCI interface, the standard HCI command set should be used.

4. Multi-advertiser Support

The objectives of multi-advertiser support are the following:

- Ability to support multiple advertisements (`max_advt_instances`)
- Different transmit powers to allow for a varying range
- Different advertising content
- An individualised response for each advertiser
- Privacy (non-trackable) for each advertiser
- Connectable

To keep this specification close to existing standards, the following vendor-specific commands are provided, and are derived from the Bluetooth Core 4.1 Specification.

4.1 LE_Multi_Advt_Command

OCF: 0x154

Command Parameter	Size	Purpose
Multi_advt_opcode	1 octet	0x01 - Set_Advt_Param_Multi_Sub_Cmd 0x02 - Set_Advt_Data_Multi_Sub_Cmd 0x03 - Set_Scan_Resp_Data_Multi_Sub_Cmd 0x04 - Set_Random_Addr_Multi_Sub_Cmd 0x05 - Set_Advt_Enable_Multi_Sub_Cmd

A command complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Multi_advt_opcode	1 octet	0x01 - Set_Advt_Param_Multi_Command 0x02 - Set_Advt_Data_Multi_Command 0x03 - Set_Scan_Resp_Data_Multi_Command 0x04 - Set_Random_Addr_Multi_Command 0x05 - Set_Advt_Enable_Multi_Command

4.1.1 LE_Multi_Advt_Command: Set_Advt_Param_Multi_Sub_Cmd

Sub OCF: 0x01

Base reference (referred to below as "spec"): The BT 4.1 core specification, page 964 (LE Set Advertising Parameter Command)

Sub-command Parameter	Size	Purpose
Advertising_Interval_Min	Per spec	Per spec
Advertising_Interval_Max	Per spec	Per spec
Advertising_Type	Per spec	Per spec
Own_Address_Type	Per spec	Per spec
Own_Address	Per spec	Per spec
Direct_Address_Type	Per spec	Per spec
Direct_Address	Per spec	Per spec
Advertising_Channel_Map	Per spec	Per spec
Adverstising_Filter_Policy	Per spec	Per spec

Advertising_Instance	1 octet	Specifies the applicability of the above parameters to an instance
Tx_power	1 octet	Transmit_Power Unit - in dBm (signed integer) Range (-70 to +20)

The `Own_Address` parameter could be a host-configured address at the time of setting up of this multi-advertisement instance. This provides the ability to have a resolvable private address at the time of the transmit of the first beacon. Advertisement on an instance will continue irrespective of the connection. The host BT stack could issue a command to start advertisement on an instance, post connection.

A Command Complete event will be generated for this command as specified in the Bluetooth Core 4.1 Specification, per the above command. Additionally, the controller shall respond with a non-success (invalid parameter) code if the advertising instance or `Tx_Power` parameters are invalid.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Multi_advt_opcode	1 octet	0x01 [Set_Advt_Param_Multi_Sub_Cmd]

4.1.2 LE_Multi_Advt_Command: Set_Advt_Data_Multi_Sub_Cmd

Sub OCF: 0x02

Base reference: The BT 4.1 core specification, page 969 (LE Set Advertising Data Command)

Sub-command Parameter	Size	Purpose
Advertising_Data_Length	Per spec	Per spec
Advertising_Data	Per spec	Per spec
Advertising_Instance	1 octet	Specifies the applicability of the above parameters to an instance

A Command Complete event will be generated for this command as specified in the Bluetooth Core 4.1 Specification, per the above command. Additionally, the controller shall respond with a non-success code if the the advertising instance or `Tx_Power` parameters are invalid.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Multi_advt_opcode	1 octet	0x02 [Set_Advt_Data_Multi_Sub_Cmd]

4.1.3 LE_Multi_Advt_Command: Set_Scan_Resp_Data_Multi_Sub_Cmd

Base reference: The BT 4.1 core spec, page 970 (LE Set Scan Response Data Command)

Sub OCF: 0x03

Sub-command Parameter	Size	Purpose
Scan_Response_Data_Length	Per spec	Per spec
Scan_Response_Data	Per spec	Per spec
Advertising_Instance	1 octet	Specifies the applicability of the above parameters to an instance

A Command Complete event will be generated for this command as specified in the Bluetooth Core 4.1 Specification, per the above command. Additionally, the controller shall respond with a non-success code (invalid parameter) if the advertising instance or Tx_Power parameters are invalid.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Multi_advt_opcode	1 octet	0x03 [Set_Scan_Resp_Data_Multi_Sub_Cmd]

4.1.4 LE_Multi_Advt_Command: Set_Random_Addr_Multi_Sub_Cmd

Base reference: The BT 4.1 core spec, page 963 (LE Set Random Address Command)

Sub OCF: 0x04

Sub-command Parameter	Size	Purpose
Random Address	Per spec	Per spec
Advertising_Instance	1 octet	Specifies the applicability of the above parameters to an instance

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Multi_advt_opcode	1 octet	0x04 [Set_Random_Addr_Multi_Sub_Cmd]

4.1.5 LE_Multi_Advt_Command: Set_Advt_Enable_Multi_Sub_Cmd

Base reference: The BT 4.1 core spec, page 971 (LE Set Advertise Enable Command in that core spec)

OCF: 0x05

Sub-command Parameter	Size	Purpose
Advertising_Enable	1 octet	A value of 1 means to enable. A different value means to disable.
Advertising_Instance	1 octet	Specifies the applicability of the above parameters to an instance. Instance 0 means a standard HCI instance.

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Multi_advt_opcode	1 octet	0x05 [Set_Advt_Enable_Multi_Sub_Cmd]

5. Offloaded Resolution of Private Address

The design intent of this feature is to allow the resolution of a private address in the controller firmware or hardware, which provides the following benefits:

- Latency involved with the host in resolving a private address
- Saving power by refraining from waking up the host

5.1 LE_RPA_offload_Command

OCF: 0x155

Command Parameter	Size	Purpose
RPA_offload_opcode	1 octet	0x1 - Enable customer specific feature 0x2 - Add IRK to the list 0x3 - Remove IRK from the list

		0x4 - Clear IRK list 0x5 - Read IRK list entry
--	--	---------------------------------------------------

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Event_RPA_offload_opcode	1 octet	0x1 - Enable customer specific feature 0x2 - Add IRK to the list 0x3 - Remove IRK from the list 0x4 - Clear IRK list 0x5 - Read IRK list entry

5.1.1 LE_RPA_offload: Enable_cust_specific_sub_Command

Sub OCF: 0x01

Sub-command Parameter	Size	Purpose
enable_customer_specific_feature_set	1 octet	0x01 - Enable offloaded RPA feature 0x00 - Disable offloaded RPA feature

RPA offload is required to be enabled by the host, based on the chip capability. Refer to the LE_Get_Vendor_Capabilities_Command. Each chip can have a varying max_irk_list_sz in the firmware.

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Event_cust_specific_feature_opcode	1 octet	0x01 [Enable customer-specific feature]

5.1.2 LE_RPA_offload: Add_IRK_to_list_sub_Command

Sub OCF: 0x02

Sub-command Parameter	Size	Purpose
LE_IRK	16 octets	LE IRK (1st byte LSB)
Address_Type	1 octet	0: Public address 1: Random address

LE_Device_Address	6 octets	Public or random address associated to IRK (1st byte LSB)
-------------------	----------	-----------------------------------------------------------

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Event_cust_specific_feature_opcode	1 octet	0x02 [Add IRK to the list]
LE_IrkList_AvailableSpaces	1 octet	Available IRL list entries after current operation

5.1.3 LE_RPA_offload: Remove_IRK_to_list_sub_Command

Sub OCF: 0x03

Sub-command Parameter	Size	Purpose
Address_Type	1 octet	0: Public address 1: Random address
LE_Device_Address	6 octets	Public or random address that associated to the IRK

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Event_cust_specific_feature_opcode	1 octet	0x03 [Remove IRK from the list]
LE_IrkList_AvailableSpaces	1 octet	Available IRL list entries after current operation

5.1.4 LE_RPA_offload: Clear_IRK_list_sub_Command

Sub OCF: 0x04

Sub-command Parameter	Size	Purpose
None		

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Event_cust_specific_feature_opcode	1 octet	0x04 [Clear IRK List]
LE_IrkList_AvailableSpaces	1 octet	Available IRL list entries after current operation [max_irk_list_sz]

5.1.5 LE_RPA_offload: Read_IRK_list_sub_Command

Sub OCF: 0x05

Sub-command Parameter	Size	Purpose
LE_read_IRK_list_entry-index	1 octet	Index of the IRK list [0, max_irk_list_sz-1]

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Event_cust_specific_feature_opcode	1 octet	0x05 [Read IRK List Entry]
LE_Read_IRK_List_entry	1 octet	Index of the IRK that the host wants to read back (maximum IRK list size is 32)
LE_IRK	16 octets	IRK value
Address_Type	1 octet	0: Public address 1: Random address
LE_Device_Address	6 octets	Public or random address associated to the IRK
LE_Resolved_Private_Address	6 octets	Current resolved resolvable private address of this IRK

6. Batching of Scan Results

A design goal is to enhance how the Bluetooth LE Scan Response event notifications are delivered to the host, in order to save power in the host.

Reduced power consumption in the host is achieved by enabling the host application processor to stay in idle/sleep longer, by reducing how often it is notified by the controller to scan result events. The chip capability for storage of scan results is indicated via the following return parameter of `LE_Get_Vendor_Capabilities_Command`:

`total_scan_results_storage`

This feature focuses on the management and configuration of the LE Scan Results storage facility in the Bluetooth controller. The storage is used to temporarily batch advertisement data, scan data and metadata that are received by the controller for later delivery to the host.

Firmware shall support two types of batching, which can be engaged simultaneously:

- Truncated. Contains the following information elements: {MAC, TX Power, RSSI, Timestamp}
- Full. Contains the following information elements: {MAC, TX Power, RSSI, Timestamp, Adv Data, Scan Response}

6.1 LE_Batch_Scan_Command

OCF: 0x156

Command Parameter	Size	Purpose
<code>Batch_Scan_opcode</code>	1 octet	0x1 - Enable customer-specific feature 0x2 - Set Batch Scan Storage parameters 0x3 - Set Batch Scan parameters 0x4 - Read Batch Scan Result parameters

A Command Complete event will be generated for this command. Enabling the customer-specific feature doesn't start the scan.

Return Parameter	Size	Purpose
<code>Status</code>	1 octet	Command Complete status
<code>Batch_Scan_opcode</code>	1 octet	0x1 - Enable customer-specific feature 0x2 - Set Batch Scan Storage parameters 0x3 - Set Batch Scan parameters 0x4 - Read Batch Scan Result parameters

6.1.1 LE_Batch_Scan_Command: Enable Customer Specific feature

Sub OCF: 0x01

Sub-command Parameter	Size	Purpose
enable_customer_specific_feature_set	1 octet	0x01 - Enable Batch Scan feature 0x00 - Disable Batch Scan feature

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Batch_Scan_opcode	1 octet	0x1 - Enable customer-specific feature 0x2 - Set Batch Scan Storage parameters 0x3 - Set Batch Scan parameters 0x4 - Read Batch Scan Result parameters

6.1.2 LE_Batch_Scan_Command: Set Batch Scan Storage Param sub-command

Sub OCF: 0x02

Sub-command Parameter	Size	Purpose
Batch_Scan_Full_Max	1 octet	Max storage space (in %) allocated to full style [Range: 0-100]
Batch_Scan_Truncated_Max	1 octet	Max storage space (in %) allocated to truncated style [Range: 0-100]
Batch_Scan_Notify_Threshold	1 octet	Setup notification level (in %) for individual storage pool [Range: 0-100]. Setting to 0 will disable notification. Vendor specific HCI event is generated (Storage threshold breach sub-event)

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Batch_scan_opcode	1 octet	0x02 [Set Batch Scan parameters]

6.1.3 LE_Batch_Scan_Command: Set Batch Scan Param sub-command

Sub OCF: 0x03

Sub-command Parameter	Size	Purpose
Batch_Scan_Mode	1 octet	0x00 – Batch scan is disabled 0x01 – Truncated mode is enabled 0x02 – Full mode is enabled 0x03 – Truncated and Full mode are enabled.
Duty_cycle_scan_window	4 octets	Batch Scan scan time (# of slot)
Duty_cyle_scan_interval	4 octets	Batch Scan interval period (# of slot)
own_address_type	1 octet	0x00 - Public device address 0x01 - Random device address
Batch_scan_Discard_Rule	1 octet	0 - Discard oldest advertisement 1 - Discard advertisement with weakest RSSI

This subcommand will start batch scanning, if enabled. In Truncated scanning, results are stored in truncated form, where the unique key for Truncated style = {BD_ADDR, scan_interval}. This means only one BD_ADDR will be recorded for each scan interval. The record to keep for Truncated mode is the following: {BD_ADDR, Tx Power, RSSI, Timestamp}.

When Full mode is enabled, active scanning will be used and Scan Responses will be recorded. The Full style unique key = {MAC, Ad packet}, irrespective of scan interval. The record to keep for Full mode is {BD_ADDR, Tx Power, RSSI, Timestamp, Ad packet, Scan Response}. In Full style, the same AD packet, when seen multiple times across different scan intervals, is recorded only once. However, in Truncated mode, it is the visibility of BA_ADDR across different scan intervals that is of interest (once per scan interval). The RSSI is the averaged value for of all duplicates of a unique advertisement within a scan interval. A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Batch_scan_opcode	1 octet	0x03 [Set Batch Scan Parameters]

6.1.4 LE_Batch_Scan_Command: Read Batch Scan Results sub-command

Sub OCF: 0x04

Sub-command Parameter	Size	Purpose
Batch_Scan_Data_read	1 octet	0x01 – Truncated mode data 0x02 – Full mode data

A Command Complete event will be generated for this command. When the host issues this command, all the results in the controller may not fit in one Command Complete event. The host will iterate issuing this command until the corresponding results in the Command Complete event indicate 0 in the number of records, which indicates the controller has no more records to communicate to the host. Each Command Complete event could contain multiple records of only one type of data (Full or Truncated).

Controller and host time references are not synchronised. Thus the timestamp needs special explanation. The unit of the timestamp is 50ms. The value of the timestamp is based off when the `Read_Batch_Scan_Results_Sub_cmd` is given by the host. Let's assume that the command arrival time is T_c , in the firmware. That actual time the timestamp was taken in the firmware is T_{fw} . The reporting time shall be: $(T_c - T_{fw}) \cdot T_c$ and T_{fw} are in the firmware time domain. This will let the host compute how long ago the event happened.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
Batch_scan_opcode	1 octet	0x03 [Set Batch Scan parameters]
Batch_Scan_data_read	1 octet	Identifies the format (Truncated or Full)
num_of_records	1 octet	Number of records of Batch_Scan_data_read
format_of_data	Variable	<u>Truncated Mode:</u> Address[0]: 6 octets Address_Type[0]: 1 octet Tx_Pwr[0]: 1 octet RSSI[0] : 1 octet Timestamp[0]: 2 octets [multiple records (num_of_records) with above format] <u>Full Mode:</u> Address[0]: 6 octets Address_Type[0]: 1 octet Tx_Pwr[0]: 1 octet RSSI[0]: 1 octet Timestamp[0]: 2 octets Adv packet_len[0]: 1 octet

		Adv_packet[0]: Adv_packet_len octets Scan_data_resp_len[0]: 1 octet Scan_data_resp[0]: Scan_data_resp octets [multiple records with above format (num_of_records)]
--	--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

7. Advertisement Packet Content filter

Used to enable/disable/setup the Advertising Packet Content Filter (APCF) in the controller.

7.1 LE_APCF_Command

OCF: 0x157

Command Parameter	Size	Purpose
APCF_opcode	1 octet	0x00 - APCF Enable 0x01 - APCF Set Filtering parameters 0x02 - APCF Broadcaster Address 0x03 - APCF Service UUID 0x04 - APCF Service Solicitation UUID 0x05 - APCF Local Name 0x06 - APCF Manufacturer Data 0x07 - APCF Service Data

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Return status
APCF_opcode	1 octet	0x00 - APCF Enable 0x01 - APCF Set Filtering parameters 0x02 - APCF Broadcaster Address 0x03 - APCF Service UUID 0x04 - APCF Service Solicitation UUID 0x05 - APCF Local Name 0x06 - APCF Manufacturer Data 0x07 - APCF Service Data

7.1.1 LE_APCF_Command: Enable_sub_cmd

Sub OCF: 0x00

Sub-command Parameter	Size	Purpose
APCF_enable	1 octet	0x01 - Enable APCF feature 0x00 - Disable APCF feature

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
APCF_opcode	1 octet	0x0 - APCF Enable
APCF_Enable	1 octet	Enable/disable is set via APCF_enable

7.1.2 LE_APCF_Command: set_filtering_parameters_sub_cmd

This sub-command is used to add or delete a filter specification or clear a filter list for on-chip filtering.

Sub OCF: 0x01

Sub-command Parameter	Size	Purpose
APCF_Action	1 octet	0x00 - Add 0x01 - Delete 0x02 - Clear Delete will clear the specific filter along with associated feature entries in other tables. Clear will clear all the filters and associated entries in other tables.
APCF_Filter_Index	1 octet	Filter index (0, max_filter-1)
APCF_Feature_Selection	2 octets	Bit masks for the selected features: Bit 0: Set to enable Broadcast Address filter Bit 1: Set to enable Service Data Change filter Bit 2: Set to enable Service UUID check Bit 3: Set to enable Service Solicitation UUID check Bit 4: Set to enable Local Name check Bit 5: Set to enable Manufacturer Data Check Bit 6: Set to enable Service Data Check
APCF_List_Logic_Type	2 octets	Logic operation for each feature selection (per bit position) specified in APCF_Feature_Selection. Valid only when a feature is enabled. Bit position value:

		<p>0: OR 1: AND</p> <p>If "AND" logic is selected, an ADV packet will pass the filter only if it contains ALL of the entries in the list.</p> <p>If "OR" logic is selected, an ADV packet will pass the filter if it contains any of the entries in the list.</p>
APCF_Filter_Logic_Type	1 octet	<p>0x00: OR 0x01: AND</p> <p>Note: The logic type is N/A for the first three fields of APCF_Feature_Selection, which is always "AND" logic. They are only applicable for (Bit 3 Bit 6) four fields of APCF_Feature_Selection.</p>
rssi_high_thresh	1 octet	[In dBm] the advertiser is deemed seen only if the signal is higher than the RSSI high threshold. Otherwise, the firmware must behave as if it never saw it.
delivery_mode	1 octet	<p>0x00 - immediate 0x01 - on_found 0x02 - batched</p>
onfound_timeout	2 octets	(Valid only if <code>delivery_mode</code> is <code>on_found</code>) [in milliseconds] Time for firmware to linger and collect additional advertisements before reporting.
onfound_timeout_cnt	1 octet	Valid only if <code>delivery_mode</code> is <code>on_found</code> [in milliseconds] If an advertisement in <code>onFound</code> lingers in firmware for the <code>onfound_timeout</code> duration, it will collect a few advertisements and the count is checked. If the count exceeds <code>onfound_timeout_cnt</code> , it's reported <code>OnFound</code> , immediately thereafter.
rssi_low_thresh	1 octet	Valid only if <code>delivery_mode</code> is <code>on_found</code> [in dBm]. The advertiser packet is considered as not seen, if the RSSI of the received packet is not above the RSSI low threshold.
onlost_timeout	2 octets	Valid only if <code>delivery_mode</code> is <code>on_found</code>

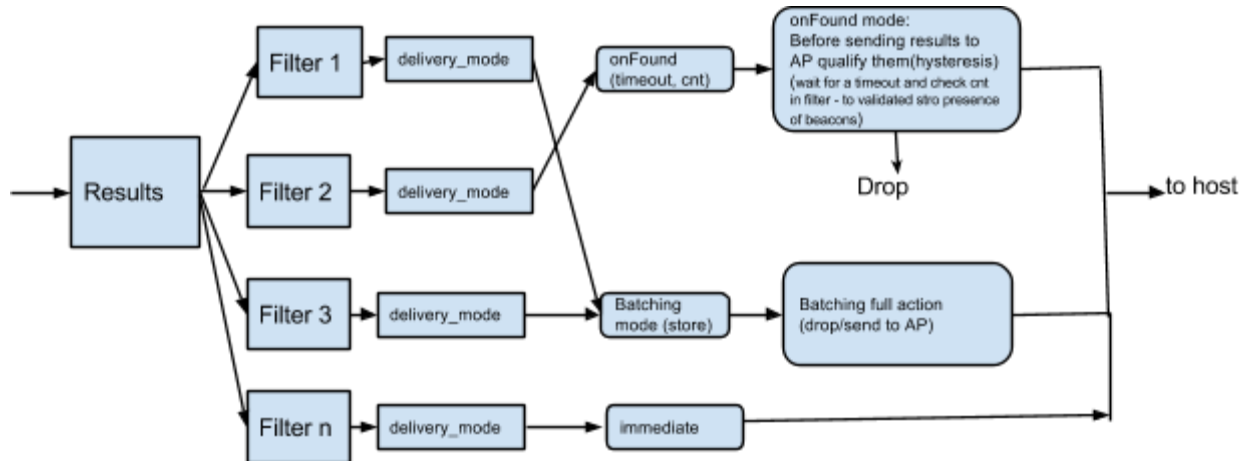
	<p>[in milliseconds]. If an advertisement, after being found, is not seen contiguously for the <code>lost_timeout</code> period, it will be reported lost. Reporting of lost is immediate.</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

RSSI values must use 2's complement of representation to represent negative values.

Host shall be able to configure multiple filters with `APCF_Application_Address_type` set to 0x02 (for all broadcaster addresses) to manage various filter combinations.

Filtering, batching and reporting are inter-related concepts. Every advertisement and related scan response will have to go through all the filters, one after the other. Thus, resulting actions (`delivery_mode`) are closely tied to filtering. The delivery modes are the following: `report_immediately`, `batch` and `onFound`. The `OnLost` value is related to `OnFound`, in the sense that it will come after `OnFound`, when lost.

The following processing flow picture will help articulate the conceptual model.



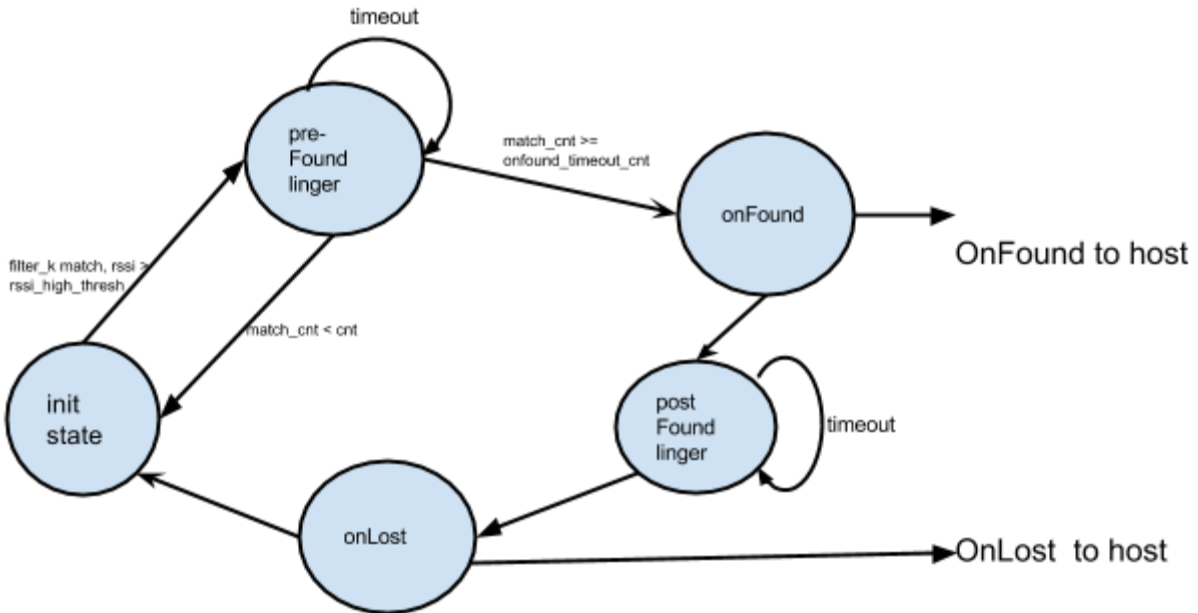
When an advertisement (or scan response) frame is received, it is applied to all the filters in serial order. It's possible that an advertisement can cause immediate reporting based on one filter and batching of the same due to a different filter action.

RSSI level thresholds (high and low) give the ability to control when the frame is visible for filter processing, even when a valid packet is received by the controller. In case of delivery mode being set to immediate or batched, the RSSI of an advertisement has to exceed or equal `rss_i_high_thresh` to be considered for further controller processing. Different apps need different reporting and batching behavior. This allows multiple apps to have direct reporting and/or batching of results in firmware, concurrently. An example is a case when a batch scan is active from one app and later a regular LE scan is issued by another app. Before a batch scan is issued, the framework/app has set appropriate filters. Later, when the second app issues a regular scan, previous batching shall continue. However, due to the regular scan, it is akin to conceptually adding a null filter (along with all the existing filters)

along with the LE scan command. The LE scan command parameters take precedence when active. When the regular LE scan is disabled, the controller will revert back to a previous batch scan, if it existed.

The OnFound delivery mode is based on configured filters. A combination that triggers a filter's action to succeed is considered the entity to track for onLost.

The OnFound/OnLost transition for a filter (if enabled) will look like the following:



A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
APCF_opcode	1 octet	0x02 - APCF Set Filtering Parameters
APCF_Action	1 octet	Echo back command's APCF_Action
APCF_AvailableSpaces	1 octet	Number of available entries in the filters table

7.1.3 LE_APCF_Command: broadcast_address_sub_cmd

This sub-command is used to add or delete an advertiser address or to clear the advertiser address list for on-chip filtering.

Sub OCF: 0x02

Sub-command Parameter	Size	Purpose
APCF_Action	1 octet	0x00 - Add 0x01 - Delete 0x02 - Clear Delete will delete the specified broadcaster address in the specified filter. Clear will clear all the broadcaster addresses in the specified filter.
APCF_Filter_Index	1 octet	Filter index (0, max_filter-1)
APCF_Broadcaster_Address	6 octet	6-byte device address to add to or delete from the broadcaster address list
APCF_Application_Address_type	1 octet	0x00: Public 0x01: Random 0x02: NA (addresses type not applicable)

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
APCF_opcode	1 octet	0x02 - APCF Broadcaster Address
APCF_Action	1 octet	Echo back command's APCF_Action
APCF_AvailableSpaces	1 octet	Number of free entries still available in the Broadcast Address table

7.1.4 LE_APCF_Command: service_uuid_sub_cmd

This sub-command is used to add or delete a service UUID or to clear a service UUID list for on-chip filtering.

Sub OCF: 0x03

Sub-command Parameter	Size	Purpose
APCF_Action	1 octet	0x00 - Add 0x01 - Delete 0x02 - Clear

		Delete will delete the specified service UUID address in the specified filter. Clear will clear all the service UUIDs in the specified filter.
APCF_Filter_Index	1 octet	Filter index (0, max_filter-1)
APCF_UUID	2,4,16 octet	The Service UUID (16-bit, 32-bit, or 128-bit) for adding to, or deleting from, the list.
APCF_UUID_MASK	2,4,16 octet	The Service UUID Mask (16-bit, 32-bit, or 128-bit) to add to the list. It should have the same length as APCF_UUID.

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
APCF_opcode	1 octet	0x03 - APCF Service UUID
APCF_Action	1 octet	Echo back command's APCF_Action
APCF_AvailableSpaces	1 octet	Number of free entries still available in the Service UUID table

7.1.5 LE_APCF_Command: solicitation_uuid_sub_cmd

This sub-command is used to add or delete a solicitation UUID or to clear a solicitation UUID list for on-chip filtering.

Sub OCF: 0x04

Sub-command Parameter	Size	Purpose
APCF_Action	1 octet	0x00 - Add 0x01 - Delete 0x02 - Clear Delete will delete the solicitation UUID address in the specified filter. Clear will clear all the solicitation UUIDs in the specified filter.
APCF_Filter_Index	1 octet	Filter index (0, max_filter-1)
APCF_UUID	2,4,16 octet	The Solicitation UUID (16-bit, 32-bit, or 128-bit) to add to or delete from the list.

APCF_UUID_MASK	2,4,16 octet	The Solicitation UUID Mask (16-bit, 32-bit, or 128-bit) to add to the list. It should have the same length as the APCF_UUID.
----------------	--------------	------------------------------------------------------------------------------------------------------------------------------

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
APCF_opcode	1 octet	0x04 - APCF Solicitation UUID
APCF_Action	1 octet	Echo back command's APCF_Action
APCF_AvailableSpaces	1 octet	Number of free entries still available in the Solicitation UUID table

7.1.6 LE_APCF_Command: local_name_sub_cmd

This sub-command is used to add or delete a local name string or to clear the local name string list for on-chip filtering.

Sub OCF: 0x05

Sub-command Parameter	Size	Purpose
APCF_Action	1 octet	0x00 - Add 0x01 - Delete 0x02 - Clear Delete will delete the specified local name string in the specified filter. Clear will clear all the local name strings in the specified filter.
APCF_Filter_Index	1 octet	Filter index (0, max_filter-1)
APCF_LocName_Mandata_or_SerData	Variable size	A character string for local name. Notes: i) Currently the max number of characters in a local name string is 29 ii) Not applicable when action is "Clear" (0x2)

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status

APCF_opcode	1 octet	0x05 - APCF Local Name
APCF_Action	1 octet	Echo back command's APCF_Action
APCF_AvailableSpaces	1 octet	Number of free entries still available in the Local name table.

7.1.7 LE_APCF_Command: manf_data_sub_cmd

This sub-command is used to add or delete a manufacturer data string or to clear the manufacturer data string list for on-chip filtering.

Sub OCF: 0x06

Sub-command Parameter	Size	Purpose
APCF_Action	1 octet	0x00 - Add 0x01 - Delete 0x02 - Clear Delete will delete the specified manufacturer data string in the specified filter. Clear will clear all the manufacturer data strings in the specified filter.
APCF_Filter_Index	1 octet	Filter Index (0, max_filter-1)
APCF_LocName_Mandata_or_SerData	Variable size	A character string for manufacturer data. Notes: i) Currently the max number of characters in a local name string is 29 ii) Not applicable when action is "Clear" (0x2)
APCF_ManData_Mask	Variable size	The manufacture data mask to add to the list. It should have the same length as APCF_LocName_or_ManData_or_SerData.

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
APCF_opcode	1 octet	0x06 - APCF Manufacturer Data
APCF_Action	1 octet	Echo back command's APCF_Action

APCF_AvailableSpaces	1 octet	Number of free entries still available in the Manufacturer Data table.
----------------------	---------	------------------------------------------------------------------------

7.1.8 LE_APCF_Command: service_data_sub_cmd

This sub-command is used to add or delete a service data string or to clear the service data string list for on-chip filtering.

Sub OCF: 0x07

Sub-command Parameter	Size	Purpose
APCF_Action	1 octet	0x00 - Add 0x01 - Delete 0x02 - Clear Delete will delete the specified service data string in the specified filter. Clear will clear all the service data strings in the specified filter.
APCF_Filter_Index	1 octet	Filter Index (0, max_filter-1)
APCF_LocName_Mandata_or_SerData	Variable size	A character string for service data. Notes: i) Currently the max number of characters in a local name string is 29 ii) Not applicable when action is "Clear" (0x2)
APCF_LocName_Mandata_or_SerData_Mask	Variable size	The service data mask to add to the list. It should have the same length as APCF_LocName_or_ManData_or_SerData.

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
APCF_opcode	1 octet	0x07 - APCF Service Data
APCF_Action	1 octet	Echo back command's APCF_Action
APCF_AvailableSpaces	1 octet	Number of free entries still available for Service Data table.

8. Controller Activity and Energy Information Command

The objective of this information is for higher host system functions to analyze the overall gross activities of all components including the BT controller and its macro state, in conjunction with what is happening in the apps and framework. To that end, the following information is required from the BT stack and the controller:

- BT stack: Reporting the current macro-operational state of the controller
- Firmware: Reporting aggregate activity and energy information

BT host stack macro states, as determined at the user level:

- Idle: [page scan, LE advt, inquiry scan]
- Scan: [paging/inquiry/trying to connect]
- Active: [ACL link on, SCO link ongoing, sniff mode]

The activities that the controller keeps track of over its life are Tx time, Rx time, idle time, and total energy consumed. They are cleared when read from the host.

Vendor Specific command: `LE_Get_Controller_Activity_Energy_Info`
OCF: 0x159

Sub-command Parameter	Size	Purpose
	NA	Empty command params

A Command Complete event will be generated for this command.

Return Parameter	Size	Purpose
Status	1 octet	Command Complete status
total_tx_time_ms	4 octet	Total time performing Tx
total_rx_time_ms	4 octet	Total time performing Rx
total_idle_time_ms	4 octet	Total time in idle (including all low power states)
total_energy_used	4 octet	Total energy used [product of current (mA), voltage (V) and time (ms)]

9. HCI Event [Vendor-specific]

Vendor-specific HCI Events are required in some cases. Refer to Figure 5.4 on page 486 of the Bluetooth Core 4.1 Specification. Event parameter 0 will always contain the first sub-event code, based on which the rest of the HCI event is decoded.

Event Parameter	Size	Purpose
HCI_vendor_specific_event_code	1 octet	0xFF
sub_event_code	1 octet	Sub-event code will be 1 octet in size, the byte immediately following Parameter Length in the HCI event packet

9.1 Storage Threshold Breach Sub-event

This event indicates that the storage threshold has been breached.

Sub-event code = 0x54

Sub-event Parameter	Size	Purpose
None		

9.2 LE Multi-Advertising State Change Sub-event

This event indicates that an advertising instance has changed its state. At this time, this event is only used to indicate which advertising instance was stopped as a result of a connection.

Sub-event code = 0x55

Sub-event Parameter	Size	Purpose
Advertising_instance	1 octet	Identifies the specific advertising instance. Valid values are 0 through (max_advt_instances - 1)
State_change_Reason	1 octet	0x00: Connection received
Connection_handle	2 octet	Identifies the connection that caused the advt instance to be disabled (0xFFFF if invalid)