

CC2564B to CC2564C Migration Guide

User's Guide



Literature Number: SWRU496A
November 2016—Revised November 2018

1	Introduction	3
2	Hardware Migration	3
3	Software Migration	3
4	Low-Energy Secure Connections	3
5	Low Energy Secure Connections API Changes	4
	5.1 Changes are in GAPAPI.h File	4
	5.2 Function Changes	4
	5.3 Structure Changes	5
	5.4 Enumeration Changes.....	7
6	HCI API Changes	8
	6.1 Function Changes	8
	6.2 Structure Changes.....	10
	6.3 Macro Changes	14
	6.4 Enumeration Changes	15
7	iSPP API Changes	16
	7.1 Function Changes	16
	7.2 Structure Changes.....	16
8	GATT API Changes	17
	8.1 Structure Changes	17
	8.2 Macro Changes	18
	8.3 Enumeration Changes	18
9	GAP API Changes	19
	9.1 Function Changes	19
	9.2 Structure Changes	20
	9.3 Enumeration Changes	22
10	L2CAP API changes	22
	10.1 Function Changes	23
	10.2 Structure Changes	24
	10.3 Enumeration changes	26
	10.4 Macro Changes	26
	Revision History	27

CC2564B to CC2564C Migration Guide

1 Introduction

The CC2564C device is TI's third generation of the CC2564 dual mode *Bluetooth*[®] and Bluetooth low energy single-chip solution. Support for both Bluetooth 4.1 and Bluetooth 4.2 support is implemented (4.2 support features integrated in TI Dual-Mode Bluetooth[®] stack). For detailed features, performance, and functionality of the CC2564C device, see the respective documents:

- Data sheet [1] [CC2564C Dual-Mode Bluetooth[®] Control](#)
- User's guides [2]:
 - [CC256xC QFN EM User's Guide](#)
 - [CC256x QFN PCB Guidelines](#)
- Application notes [3] [CC2564C Bluetooth[®] 4.1 and 4.2 Application Note](#)

2 Hardware Migration

The CC2564C device is pin compatible to the CC2564B device.

Any design (schematics and layout) for the CC2564B device is compatible with the CC2564C device.

The only BOM difference is the different generic part number for the CC2564C device is replacing the CC2564B device.

QDID links:

- Stack QDID: https://www.bluetooth.org/tpg/QLI_viewQDL.cfm?qid=32797
- Controller QDID: https://www.bluetooth.org/tpg/QLI_viewQDL.cfm?qid=32801

3 Software Migration

When migrating from CC2564B to CC2564C, the host stack must be upgraded to support CC2564C even if the BT4.2 features of CC2564C are not used. Additionally, the host stack must use the appropriate device-specific service pack for initialization.

Following are the TI dual-mode Bluetooth stack releases compatible with CC2564C.

- [CC2564CMSP432BTBLESW](#)
- [CC2564CSTBTBLESW](#)
- [TI-BT-4-2-STACK-LINUX-ADDON](#)

4 Low-Energy Secure Connections

The low-energy secure connections feature introduces a new security model. The feature uses an Elliptic Curve Diffie Hellman (ECDH) algorithm for key generation, and it uses a new pairing procedure for the key exchange. This feature raises the bar and helps solve man-in-the-middle issues and other passive eavesdropping mechanisms.

⁽¹⁾ Bluetooth is a registered trademark of Bluetooth SIG, Inc.

⁽²⁾ All other trademarks are the property of their respective owners.

5 Low Energy Secure Connections API Changes

5.1 Changes are in GAPAPI.h File

NOTE: Changes are marked in **RED** text.

5.2 Function Changes

```

int BTPSAPI GAP_LE_Request_Security (
unsigned int BluetoothStackID,
  BD_ADDR_t BD_ADDR,
  GAP_LE_Bonding_Type_t Bonding_Type,
  Boolean_t MITM,
  Boolean_t SC,
      Boolean_t Keypress,
      Boolean_t P256Debug,
  GAP_LE_Event_Callback_t GAP_LE_Event_Callback,
  unsigned long CallbackParameter)

      int BTPSAPI GAP_LE_SC_OOB_Generate_Parameters( unsigned int BluetoothStackID,
      SM_Random_Value_t *OOB_Local_Rand_Result, SM_Confirm_Value_t
*OOB_Local_Confirm_Result) int BTPSAPI GAP_LE_SC_Send_Keypress_Notification(
      unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, GAP_LE_Keypress_t
*Keypress_Notification_Type) int
      BTPSAPI GAP_LE_SC_Only_Mode( unsigned int BluetoothStackID, Boolean_t EnableSOnly)
int BTPSAPI GAP_LE_SC_Regenerate_P256_Local_Keys(
      unsigned int BluetoothStackID)

```

5.3 Structure Changes

```

typedef struct _tagGAP_LE_Pairing_Capabilities_t
{
    GAP_LE_IO_Capability_t    IO_Capability;
    Boolean_t                 OOB_Present;
    GAP_LE_Bonding_Type_t    Bonding_Type;
    Boolean_t                 MITM; Boolean_t SC; Boolean_t Keypress; Boolean_t P256DebugMode;
    Byte_t                    Maximum_Encryption_Key_Size;
    GAP_LE_Key_Distribution_t Receiving_Keys;
    GAP_LE_Key_Distribution_t Sending_Keys;
} GAP_LE_Pairing_Capabilities_t;

```

```

typedef struct _tagGAP_LE_SC_OOB_Data_t { SM_Random_Value_t RemoteRand;
    SM_Confirm_Value_t RemoteConfirmation; } GAP_LE_SC_OOB_Data_t;

```

```

typedef struct _tagGAP_LE_Authentication_Response_Information_t
{
    GAP_LE_Authentication_Response_Type_t GAP_LE_Authentication_Type;
    Byte_t                                Authentication_Data_Length;
    union
    {
        GAP_LE_Long_Term_Key_Information_t Long_Term_Key_Information;
        GAP_LE_Pairing_Capabilities_t     Pairing_Capabilities;
        GAP_LE_OOB_Data_t                 Out_Of_Band_Data;
        DWord_t                           Passkey;
        Boolean_t AcceptedNumericValue; GAP_LE_SC_OOB_Data_t SC_Out_Of_Band_Data;
        Byte_t                             Error_Code;
        GAP_LE_Encryption_Information_t    Encryption_Information;
        GAP_LE_Identity_Information_t      Identity_Information;
        GAP_LE_Signing_Information_t       Signing_Information;
    } Authentication_Data;
} GAP_LE_Authentication_Response_Information_t;

```

```

typedef struct _tagGAP_LE_Slave_Security_Information_t
{
    GAP_LE_Bonding_Type_t    Bonding_Type;
    Boolean_t                 MITM;
    Boolean_t SC; Boolean_t Keypress; Boolean_t P256DebugMode;
} GAP_LE_Slave_Security_Information_t;

```

```

typedef struct _tagGAP_LE_Encryption_Change_Event_Data_t
{
    BD_ADDR_t                BD_ADDR;
    Byte_t                   Encryption_Change_Status;
    GAP_Encryption_Mode_t    Encryption_Mode;
    Boolean_t SC; Long_Term_Key_t LTK; Boolean_t Reestablish; Byte_t Encryption_Key_Size;
} GAP_LE_Encryption_Change_Event_Data_t;

```

```

    typedef struct _tagGAP_LE_Authentication_Event_Data_t
    {
        GAP_LE_Authentication_Event_Type_t GAP_LE_Authentication_Event_Type;
        BD_ADDR_t                          BD_ADDR;
        union
        {
            GAP_LE_Key_Request_Info_t      Long_Term_Key_Request;
            GAP_LE_Pairing_Capabilities_t   Pairing_Request;
            GAP_LE_Security_Request_t      Security_Request;
            GAP_LE_Confirmation_Request_t   Confirmation_Request;
            GAP_LE_Pairing_Status_t        Pairing_Status;
            GAP_LE_Encryption_Request_Information_t Encryption_Request_Information;
            GAP_LE_Encryption_Information_t Encryption_Information;
            GAP_LE_Identity_Information_t   Identity_Information;
            GAP_LE_Signing_Information_t    Signing_Information;
            GAP_LE_Security_Establishment_Complete_t Security_Establishment_Complete;
            GAP_LE_Keypress_t Keypress_Type;
        } Authentication_Event_Data;
    } GAP_LE_Authentication_Event_Data_t;
  
```

```

    typedef struct _tagGAP_LE_Security_Request_t
    {
        GAP_LE_Bonding_Type_t Bonding_Type;
        Boolean_t             MITM;
        Boolean_t SC; Boolean_t Keypress;
    } GAP_LE_Security_Request_t;
  
```

5.4 Enumeration Changes

```

        typedef enum
    {
        larLongTermKey,
        larOutOfBandData,
        larPairingCapabilities,
        larPasskey,
        larUserConfirmation,
        larConfirmation,
        larError,
        larEncryptionInformation,
        larIdentityInformation,
        larSigningInformation
    } GAP_LE_Authentication_Response_Type_t;

```

typedef

```

        enum
    {
        latLongTermKeyRequest,
        latSecurityRequest,
        latPairingRequest,
        latKeypressNotification,
        latConfirmationRequest,
        latPairingStatus,
        latEncryptionInformationRequest,
        latIdentityInformationRequest,
        latSigningInformationRequest,
        latEncryptionInformation,
        latIdentityInformation,
        latSigningInformation,
        latSecurityEstablishmentComplete
    } GAP_LE_Authentication_Event_Type_t;

```

typedef

```

        enum
    {
        crtNone,
        crtNumeric,
        crtPasskey,
        crtDisplay,
        crtOOB
    } GAP_LE_Confirmation_Request_Type_t;

```

```

        #define GAP_LE_PAIRING_STATUS_DHKEY_CHECK_FAILED 0x0E #define
GAP_LE_PAIRING_STATUS_NUMERIC_COMPARISON_FAILED
        0x0F

```

6 HCI API Changes

NOTE: Changes are in the HCITypes.h file (changes are marked in **RED** text).

6.1 Function Changes

```
BTPSAPI_DECLARATION int BTPSAPI HCI_Read_Authenticated_Payload_Timeout(  
    unsigned int BluetoothStackID, Word_t Connection_Handle, Byte_t *StatusResult, Word_t  
*Connection_HandleResult,  
    Word_t *Authenticated_Payload_TimeoutResult);
```

```
BTPSAPI_DECLARATION int BTPSAPI HCI_Write_Authenticated_Payload_Timeout(  
    unsigned int BluetoothStackID, Word_t Connection_Handle, Word_t  
Authenticated_Payload_Timeout, Byte_t  
*StatusResult, Word_t *Connection_HandleResult);
```

```
BTPSAPI_DECLARATION int BTPSAPI HCI_Read_Extended_Page_Timeout( unsigned  
    int BluetoothStackID, Byte_t *StatusResult, Word_t *Extended_Page_TimeoutResult);
```

```
BTPSAPI_DECLARATION int BTPSAPI HCI_Write_Extended_Page_Timeout( unsigned  
    int BluetoothStackID, Word_t Extended_Page_Timeout, Byte_t *StatusResult);
```

```
BTPSAPI_DECLARATION int BTPSAPI HCI_Read_Extended_Inquiry_Length( unsigned  
    int BluetoothStackID, Byte_t *StatusResult, Word_t *Extended_Inquiry_LengthResult);
```



```
BTPSAPI_DECLARATION int BTPSAPI HCI_Write_Extended_Inquiry_Length(
    unsigned int BluetoothStackID, Word_t Extended_Inquiry_Length, Byte_t *StatusResult);

int BTPSAPI HCI_Write_Secure_Connections_Test_Mode( unsigned int BluetoothStackID,
    Word_t Connection_Handle, Byte_t DM1_ACLU_Mode, Byte_t ESCO_Loopback_Mode, Byte_t
*StatusResult, Word_t
    *Connection_HandleResult)

BTPSAPI_DECLARATION int BTPSAPI HCI_LE_Remote_Connection_Parameter_Request_Reply(
    unsigned int BluetoothStackID, Word_t Connection_Handle, Word_t Conn_Interval_Min,
Word_t Conn_Interval_Max,
    Word_t Conn_Latency, Word_t Supervision_Timeout, Word_t Minimum_CE_Length, Word_t
Maximum_CE_Length,
    Byte_t *StatusResult, Word_t *Connection_HandleResult);

BTPSAPI_DECLARATION int BTPSAPI
HCI_LE_Remote_Connection_Parameter_Request_Negative_Reply(
    unsigned int BluetoothStackID, Word_t Connection_Handle, Byte_t Reason, Byte_t
*StatusResult, Word_t
    *Connection_HandleResult);
```

6.2 Structure Changes

```

typedef __PACKED_STRUCT_BEGIN__ struct
_tagHCI_Read_Authenticated_Payload_Timeout_Command_t
{ HCI_Command_Header_t HCI_Command_Header; NonAlignedWord_t Connection_Handle; }
__PACKED_STRUCT_END__
HCI_Read_Authenticated_Payload_Timeout_Command_t;

```

```

typedef __PACKED_STRUCT_BEGIN__ struct
_tagHCI_Write_Authenticated_Payload_Timeout_Command_t
{ HCI_Command_Header_t HCI_Command_Header; NonAlignedWord_t Connection_Handle;
NonAlignedWord_t Authenticated_Payload_Timeout;
} __PACKED_STRUCT_END__ HCI_Write_Authenticated_Payload_Timeout_Command_t;

```

```

typedef __PACKED_STRUCT_BEGIN__ struct
_tagHCI_LE_Remote_Connection_Parameter_Request_Reply_Command_t
{ HCI_Command_Header_t HCI_Command_Header; NonAlignedWord_t Connection_Handle;
NonAlignedWord_t Interval_Min;
NonAlignedWord_t Interval_Max; NonAlignedWord_t Latency; NonAlignedWord_t Timeout;
NonAlignedWord_t Minimum_CE_Length;
NonAlignedWord_t Maximum_CE_Length; } __PACKED_STRUCT_END__
HCI_LE_Remote_Connection_Parameter_Request_Reply_Command_t;

```

```

typedef __PACKED_STRUCT_BEGIN__ struct
_tagHCI_LE_Remote_Connection_Parameter_Request_Negative_Reply_Command_t
{ HCI_Command_Header_t HCI_Command_Header; NonAlignedWord_t Connection_Handle; Byte_t
Reason; } __PACKED_STRUCT_END__
HCI_LE_Remote_Connection_Parameter_Request_Negative_Reply_Command_t;

```

```

typedef __PACKED_STRUCT_BEGIN__ struct
_tagHCI_LE_Remote_Connection_Parameter_Request_Event_t
HCI_LE_Meta_Event_Header_t HCI_LE_Meta_Event_Header; NonAlignedWord_t
Connection_Handle; NonAlignedWord_t
Conn_Interval_Min; NonAlignedWord_t Conn_Interval_Max; NonAlignedWord_t Conn_Latency;
NonAlignedWord_t
Supervision_Timeout; } __PACKED_STRUCT_END__
HCI_LE_Remote_Connection_Parameter_Request_Event_t;

```

```

typedef __PACKED_STRUCT_BEGIN__ struct
_tagHCI_Authenticated_Payload_Timeout_Expired_Event_t
{ HCI_Event_Header_t HCI_Event_Header; NonAlignedWord_t Connection_Handle; }
__PACKED_STRUCT_END__ HCI_Authenticated_Payload_Timeout_Expired_Event_t;

```

```

typedef __PACKED_STRUCT_BEGIN__ struct
_tagHCI_Read_Authenticated_Payload_Timeout_Command_Complete_Event_t
{ HCI_Command_Complete_Event_Header_t HCI_Event_Header; Byte_t Status;
NonAlignedWord_t Connection_Handle;

```

```
NonAlignedWord_t Authenticated_Payload_Timeout; } __PACKED_STRUCT_END__  
HCI_Read_Authenticated_Payload_Timeout_Command_Complete_Event_t;
```

```

typedef __PACKED_STRUCT_BEGIN__ struct
_tagHCI_Write_Authenticated_Payload_Timeout_Command_Complete_Event_t
{ HCI_Command_Complete_Event_Header_t HCI_Event_Header; Byte_t Status;
NonAlignedWord_t Connection_Handle;
} __PACKED_STRUCT_END__
HCI_Write_Authenticated_Payload_Timeout_Command_Complete_Event_t;

```

```

typedef __PACKED_STRUCT_BEGIN__ struct
_tagHCI_LE_Remote_Connection_Parameter_Request_Reply_Command_Complete_t
{ HCI_Command_Complete_Event_Header_t HCI_Event_Header; Byte_t Status;
NonAlignedWord_t Connection_Handle;
} __PACKED_STRUCT_END__
HCI_LE_Remote_Connection_Parameter_Request_Reply_Command_Complete_t;

```

```

typedef __PACKED_STRUCT_BEGIN__ struct
_tagHCI_LE_Remote_Connection_Parameter_Request_Negative_Reply_Command_Complete_t
{ HCI_Command_Complete_Event_Header_t HCI_Event_Header; Byte_t Status;
NonAlignedWord_t Connection_Handle;
} __PACKED_STRUCT_END__
HCI_LE_Remote_Connection_Parameter_Request_Negative_Reply_Command_Complete_t;

```

```

typedef struct _tagHCI_LE_Remote_Connection_Parameter_Request_Event_Data_t
{ Word_t Connection_Handle; Word_t Conn_Interval_Min; Word_t Conn_Interval_Max;
Word_t Conn_Latency;
Word_t Supervision_Timeout; } HCI_LE_Remote_Connection_Parameter_Request_Event_Data_t;

```

```

typedef struct _tagHCI_LE_Meta_Event_Data_t

    HCI_LE_Meta_Event_Type_t LE_Event_Data_Type;
    union
    {
        HCI_LE_Connection_Complete_Event_Data_t
HCI_LE_Connection_Complete_Event_Data;
        HCI_LE_Advertising_Report_Event_Data_t
HCI_LE_Advertising_Report_Event_Data;
        HCI_LE_Connection_Update_Complete_Event_Data_t
HCI_LE_Connection_Update_Complete_Event_Data;
        HCI_LE_Read_Remote_Used_Features_Complete_Event_Data_t
HCI_LE_Read_Remote_Used_Features_Complete_Event_Data;
        HCI_LE_Long_Term_Key_Request_Event_Data_t
HCI_LE_Long_Term_Key_Request_Event_Data;
        HCI_LE_Remote_Connection_Parameter_Request_Event_Data_t
HCI_LE_Remote_Connection_Parameter_Request_Event_Data;
    } Event_Data;
} HCI_LE_Meta_Event_Data_t;

    typedef struct _tagHCI_Authenticated_Payload_Timeout_Expired_Event_Data_t
    { Word_t Connection_Handle; } HCI_Authenticated_Payload_Timeout_Expired_Event_Data_t;

typedef struct _tagHCI_Event_Data_t
{
    HCI_Event_Type_t Event_Data_Type;
    Word_t          Event_Data_Size;
    union
    {
        HCI_Inquiry_Complete_Event_Data_t
*HCI_Inquiry_Complete_Event_Data;
        HCI_Inquiry_Result_Event_Data_t
*HCI_Inquiry_Result_Event_Data;
        .
        .
        .
        HCI_Triggered_Clock_Capture_Event_Data_t *HCI_Triggered_Clock_Capture_Event_Data;
        HCI_Synchronization_Train_Complete_Event_Data_t
*HCI_Synchronization_Train_Complete_Event_Data; HCI_Synchronization_Train_Received_Event_Data_t
*HCI_Synchronization_Train_Received_Event_Data;
        HCI_Connectionless_Slave_Broadcast_Receive_Event_Data_t
*HCI_Connectionless_Slave_Broadcast_Receive_Event_Data;
        HCI_Connectionless_Slave_Broadcast_Timeout_Event_Data_t
*HCI_Connectionless_Slave_Broadcast_Timeout_Event_Data;
        HCI_Truncated_Page_Complete_Event_Data_t *HCI_Truncated_Page_Complete_Event_Data;
        HCI_Connectionless_Slave_Broadcast_Channel_Map_Change_Event_Data_t
*HCI_Connectionless_Slave_Broadcast_Channel_Map_Change_Event_Data;
        HCI_Inquiry_Response_Notification_Event_Data_t
*HCI_Inquiry_Response_Notification_Event_Data;
        HCI_Authenticated_Payload_Timeout_Expired_Event_Data_t
*HCI_Authenticated_Payload_Timeout_Expired_Event_Data;
        HCI_Platform_Specific_Event_Data_t
*HCI_Platform_Specific_Event_Data;
        void
*HCI_Unknown_Event_Data;
    } Event_Data;
} HCI_Event_Data_t;

```

6.3 Macro Changes

```

#define HCI_ENABLE_ALL_HCI_LE_EVENTS_IN_EVENT_MASK(_x)
{
    ASSIGN_EVENT_MASK((_x), 0, 0, 0, 0, 0, 0, 0, 0);
    SET_EVENT_MASK_BIT((_x), HCI_LE_EVENT_MASK_CONNECTION_COMPLETE_BIT_NUMBER);
    SET_EVENT_MASK_BIT((_x), HCI_LE_EVENT_MASK_ADVERTISING_REPORT_BIT_NUMBER);
    SET_EVENT_MASK_BIT((_x), HCI_LE_EVENT_MASK_CONNECTION_UPDATE_COMPLETE_BIT_NUMBER);
    SET_EVENT_MASK_BIT((_x),
HCI_LE_EVENT_MASK_READ_REMOTE_USED_FEATURES_COMPLETE_BIT_NUMBER);
    SET_EVENT_MASK_BIT((_x), HCI_LE_EVENT_MASK_LONG_TERM_KEY_REQUEST_BIT_NUMBER);
    SET_EVENT_MASK_BIT((_x),
HCI_LE_EVENT_MASK_REMOTE_CONNECTION_PARAMETER_REQUEST_BIT_NUMBER);

#define HCI_DISABLE_ALL_HCI_LE_EVENTS_IN_EVENT_MASK(_x)
{
    ASSIGN_EVENT_MASK((_x), 0, 0, 0, 0, 0, 0, 0, 0);
    RESET_EVENT_MASK_BIT((_x), HCI_LE_EVENT_MASK_CONNECTION_COMPLETE_BIT_NUMBER);
    RESET_EVENT_MASK_BIT((_x), HCI_LE_EVENT_MASK_ADVERTISING_REPORT_BIT_NUMBER);
    RESET_EVENT_MASK_BIT((_x), HCI_LE_EVENT_MASK_CONNECTION_UPDATE_COMPLETE_BIT_NUMBER);
    RESET_EVENT_MASK_BIT((_x),
HCI_LE_EVENT_MASK_READ_REMOTE_USED_FEATURES_COMPLETE_BIT_NUMBER);
    RESET_EVENT_MASK_BIT((_x), HCI_LE_EVENT_MASK_LONG_TERM_KEY_REQUEST_BIT_NUMBER);
    RESET_EVENT_MASK_BIT((_x),
HCI_LE_EVENT_MASK_REMOTE_CONNECTION_PARAMETER_REQUEST_BIT_NUMBER);
}

```

6.4 Enumeration Changes

```

typedef enum
{
    etInquiry_Complete_Event,
    etInquiry_Result_Event,
    .
    .
    .
    etTriggered_Clock_Capture_Event, etSynchronization_Train_Complete_Event,
etSynchronization_Train_Received_Event,
    etConnectionless_Slave_Broadcast_Receive_Event,
etConnectionless_Slave_Broadcast_Timeout_Event, etTruncated_Page_Complete_Event,
    etSlave_Page_Response_Timeout_Event,
etConnectionless_Slave_Broadcast_Channel_Map_Change_Event, etInquiry_Response_Notification_Event,
    etAuthenticated_Payload_Timeout_Expired_Event
} HCI_Event_Type_t;

typedef enum
{
    meConnection_Complete_Event, meAdvertising_Report_Event,
    meConnection_Update_Complete_Event, meRead_Remote_Used_Features_Complete_Event,
meLong_Term_Key_Request_Event,
    meRemote_Connection_Parameter_Request_Event
} HCI_LE_Meta_Event_Type_t;

```

7 iSPP API Changes

NOTE: Changes are in the ISPPAPI.h file (changes are marked in **RED** text).

7.1 Function Changes

```
int BTPSAPI ISPP_Get_Configuration_Parameters( unsigned int BluetoothStackID,
        ISPP_Configuration_Params_t *ISPPConfigurationParams)
```

```
int BTPSAPI ISPP_Set_Configuration_Parameters( unsigned int BluetoothStackID,
        ISPP_Configuration_Params_t *ISPPConfigurationParams)
```

7.2 Structure Changes

```
typedef struct _tagISPP_Configuration_Params_t
{
    unsigned int RetransmitAttempts;
    unsigned int PacketTimeout;
    Word_t      MaximumFrameSize;
    unsigned int TransmitBufferSize;
    unsigned int ReceiveBufferSize;
    unsigned int MaxOutstandingPackets;
    unsigned int MaxCumulativeAcks; unsigned int CumulativeAckTimeout;
    unsigned int SessionTypesSupported;
    Word_t      SupportedMessages;
} ISPP_Configuration_Params_t;
```


8 GATT API Changes

NOTE: Changes are in the GATTAPI.h file.
GATTType.h file (changes are marked in **RED** text).

8.1 Structure Changes

```
typedef struct _tagGATT_UUID_t
{
    GATT_UUID_Type_t UUID_Type;
    union
    {
        UUID_16_t UUID_16;
        UUID_32_t UUID_32;
        UUID_128_t UUID_128;
    } UUID;
} GATT_UUID_t;
```

```
typedef struct _tagGATT_Primary_Service_32_Entry_t { UUID_32_t Service_UUID;
} GATT_Primary_Service_32_Entry_t;
```

```
typedef struct _tagGATT_Secondary_Service_32_Entry_t { UUID_32_t Service_UUID;
} GATT_Secondary_Service_32_Entry_t;
```

```
typedef struct _tagGATT_Characteristic_Declaration_32_Entry_t { Byte_t
    Properties; UUID_32_t Characteristic_Value_UUID; }
GATT_Characteristic_Declaration_32_Entry_t;
```

```
typedef struct _tagGATT_Characteristic_Value_32_Entry_t { UUID_32_t
    Characteristic_Value_UUID; unsigned int Characteristic_Value_Length; Byte_t
*Characteristic_Value; }
GATT_Characteristic_Value_32_Entry_t;
```

```
typedef struct _tagGATT_Characteristic_Descriptor_32_Entry_t { UUID_32_t
    Characteristic_Descriptor_UUID; unsigned int Characteristic_Descriptor_Length; Byte_t
*Characteristic_Descriptor;
} GATT_Characteristic_Descriptor_32_Entry_t;
```

8.2 Macro Changes

```

#define COMPARE_BLUETOOTH_UUID_128_TO_UUID_32_CONSTANT(_x, _a, _b,
    _c, _d) COMPARE_BLUETOOTH_UUID_128_TO_CONSTANT((_x), _a, _b, _c, _d, 0x00, 0x00,
0x10, 0x00, 0x80, 0x00,
    0x00, 0x80, 0x5F, 0x9B, 0x34,

#define COMPARE_BLUETOOTH_UUID_128_TO_BLUETOOTH_UUID_32(_x, _y)
COMPARE_BLUETOOTH_UUID_128_TO_CONSTANT((_x),
    (_y).UUID_Byte3, (_y).UUID_Byte2, (_y).UUID_Byte1, (_y).UUID_Byte0, 0x00, 0x00,

#define COMPARE_BLUETOOTH_UUID_TO_BLUETOOTH_BASE_UUID(_x)
COMPARE_BLUETOOTH_UUID_128_TO_CONSTANT((_x),
    (_x).UUID_Byte15, (_x).UUID_Byte14, (_x).UUID_Byte13, (_x).UUID_Byte12, 0x00, 0x00,

```

8.3 Enumeration Changes

```

typedef enum
{
    guUUID_16,
    guUUID_128,
    guUUID_32
} GATT_UUID_Type_t;

```

```

typedef enum
{
    aetPrimaryService16,
    aetPrimaryService128,
    aetSecondaryService16,
    aetSecondaryService128,
    aetIncludeDefinition,
    aetCharacteristicDeclaration16,
    aetCharacteristicDeclaration128,
    aetCharacteristicValue16,
    aetCharacteristicValue128,
    aetCharacteristicDescriptor16,
    aetCharacteristicDescriptor128,
    aetPrimaryService32, aetSecondaryService32, aetCharacteristicDeclaration32,
    aetCharacteristicValue32,
    aetCharacteristicDescriptor32
} GATT_Service_Attribute_Entry_Type_t;

```

9 GAP API Changes

NOTE: Changes are in the GAPAPI.h file (changes are marked in **RED** text).

9.1 Function Changes

```

BTPSAPI_DECLARATION int BTPSAPI GAP_Query_Local_Extended_Out_Of_Band_Data(
    unsigned int BluetoothStackID, GAP_Extended_Out_Of_Band_Data_t
    *ExtendedOutOfBandData);

```

```

BTPSAPI_DECLARATION int BTPSAPI GAP_Set_Authenticated_Payload_Timeout(
    unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, Word_t AuthenticatedPayloadTimeout);

```

```

BTPSAPI_DECLARATION int BTPSAPI GAP_Query_Authenticated_Payload_Timeout(
    unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, Word_t
    *AuthenticatedPayloadTimeout);

```

```

BTPSAPI_DECLARATION int BTPSAPI GAP_LE_Set_Authenticated_Payload_Timeout(
    unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, Word_t AuthenticatedPayloadTimeout);

```

```

BTPSAPI_DECLARATION int BTPSAPI GAP_LE_Query_Authenticated_Payload_Timeout(
    unsigned int BluetoothStackID, BD_ADDR_t BD_ADDR, Word_t
    *AuthenticatedPayloadTimeout);

```

Affected functions

```

int BTPSAPI GAP_Authentication_Response(
    unsigned int BluetoothStackID,
    BD_ADDR_t BD_ADDR,
    GAP_Authentication_Information_t *GAP_Authentication_Information)

```

9.2 Structure Changes

```

typedef struct _tagGAP_Extended_Out_Of_Band_Data_t { Simple_Pairing_Hash_t
    Simple_Pairing_Hash_192; Simple_Pairing_Randomizer_t Simple_Pairing_Randomizer_192;
Simple_Pairing_Hash_t
    Simple_Pairing_Hash_256; Simple_Pairing_Randomizer_t Simple_Pairing_Randomizer_256; }
GAP_Extended_Out_Of_Band_Data_t;

```

```

typedef struct _tagGAP_IO_Capabilities_t
{
    GAP_IO_Capability_t          IO_Capability;
    Boolean_t                    OOB_Data_Present;
    Boolean_t                    MITM_Protection_Required;
    GAP_IO_Capability_Bonding_Type_t Bonding_Type;
    Boolean_t OOB_256_Data_Present;
} GAP_IO_Capabilities_t;

```

```

typedef struct _tagGAP_Authentication_Information_t
{
    GAP_Authentication_Type_t    GAP_Authentication_Type;
    Byte_t                      Authentication_Data_Length;
    union
    {
        PIN_Code_t              PIN_Code;
        Link_Key_t              Link_Key;
        Boolean_t                Confirmation;
        DWord_t                  Passkey;
        GAP_Keypress_t           Keypress;
        GAP_Out_Of_Band_Data_t   Out_Of_Band_Data;
        GAP_Extended_Out_Of_Band_Data_t Extended_Out_Of_Band_Data;
        GAP_IO_Capabilities_t    IO_Capabilities;
    } Authentication_Data;
} GAP_Authentication_Information_t;

```

```

typedef struct _tagGAP_Authenticated_Payload_Timeout_Event_Data_t {
    BD_ADDR_t BD_ADDR; } GAP_Authenticated_Payload_Timeout_Event_Data_t;

```

```

typedef struct _tagGAP_Event_Data_t
{
    GAP_Event_Type_t Event_Data_Type;
    Word_t           Event_Data_Size;
    union
    {
        GAP_Inquiry_Event_Data_t          *GAP_Inquiry_Event_Data;
        .
        GAP_Authenticated_Payload_Timeout_Event_Data_t
    } Event_Data;
} GAP_Event_Data_t;

```

```

typedef struct _tagGAP_LE_Authenticated_Payload_Timeout_Expired_Event_Data_t
{ BD_ADDR_t BD_ADDR; } GAP_LE_Authenticated_Payload_Timeout_Expired_Event_Data_t;

```

```
typedef struct _tagGAP_LE_Event_Data_t
{
    GAP_LE_Event_Type_t Event_Data_Type;
    Word_t              Event_Data_Size;
    union
    {
        GAP_LE_Remote_Features_Event_Data_t
    *GAP_LE_Remote_Features_Event_Data;
        .
        .
        GAP_LE_Authenticated_Payload_Timeout_Expired_Event_Data_t
    *GAP_LE_Authenticated_Payload_Timeout_Expired_Event_Data;
    } Event_Data;
} GAP_LE_Event_Data_t;
```

9.3 Enumeration Changes

```

typedef enum
{
    emDisabled,
    emEnabled,
    emEnabled_AES
} GAP_Encryption_Mode_t

typedef enum
{
    etInquiry_Result,
    etEncryption_Change_Result,
    etAuthentication,
    etRemote_Name_Result,
    etInquiry_Entry_Result,
    etInquiry_With_RSSI_Entry_Result,
    etExtended_Inquiry_Entry_Result,
    etEncryption_Refresh_Complete,
    etRemote_Features_Result,
    etRemote_Version_Information_Result,
    etAuthenticated_Payload_Timeout
} GAP_Event_Type_t;

typedef enum
{
    lcmNonConnectable,
    lcmConnectable,
    lcmDirectConnectable,
    lcmLowDutyCycleDirectConnectable
} GAP_LE_Connectability_Mode_t;

typedef enum
{
    etLE_Remote_Features_Result,
    etLE_Advertising_Report,
    etLE_Connection_Complete,
    etLE_Disconnection_Complete,
    etLE_Encryption_Change,
    etLE_Encryption_Refresh_Complete,
    etLE_Authentication,
    etLE_Connection_Parameter_Update_Request,
    etLE_Connection_Parameter_Update_Response,
    etLE_Connection_Parameter_Updated,
    etLE_Authenticated_Payload_Timeout_Expired
} GAP_LE_Event_Type_t;

```

10 L2CAP API changes

NOTE: Changes are in the L2CAPTyp.h file and the L2CAPAPI.h file (changes are marked in **RED** text).

10.1 Function Changes

```
BTPSAPI_DECLARATION int BTPSAPI L2CA_Register_LE_PSM( unsigned int
    BluetoothStackID, Word_t LE_PSM, L2CA_Event_Callback_t L2CA_Event_Callback, unsigned
long CallbackParameter);
```

```
BTPSAPI_DECLARATION int BTPSAPI L2CA_Un_Register_LE_PSM( unsigned int
    BluetoothStackID, unsigned int L2CAP_PSMID);
```

```
BTPSAPI_DECLARATION int BTPSAPI L2CA_LE_Connect_Request( unsigned int
    BluetoothStackID, BD_ADDR_t BD_ADDR, Word_t LE_PSM, L2CA_LE_Channel_Parameters_t
*ChannelParameters,
    L2CA_Event_Callback_t L2CA_Event_Callback, unsigned long CallbackParameter);
```

```
BTPSAPI_DECLARATION int BTPSAPI L2CA_LE_Connect_Response( unsigned
    int BluetoothStackID, BD_ADDR_t BD_ADDR, Byte_t Identifier, Word_t LCID, Word_t
Result, L2CA_LE_Channel_Parameters_t
*ChannelParameters);
```

```
BTPSAPI_DECLARATION int BTPSAPI L2CA_LE_Disconnect_Request( unsigned
    int BluetoothStackID, Word_t LCID);
```

```
BTPSAPI_DECLARATION int BTPSAPI L2CA_LE_Disconnect_Response( unsigned
    int BluetoothStackID, Word_t LCID);
```

```
BTPSAPI_DECLARATION int BTPSAPI L2CA_Enhanced_Dynamic_Channel_Data_Write(
    unsigned int BluetoothStackID, Word_t LCID, L2CA_Queueing_Parameters_t
*QueueingParameters, Word_t Data_Length,
    Byte_t *Data);
```

```
BTPSAPI_DECLARATION int BTPSAPI L2CA_LE_Flush_Channel_Data( unsigned
    int BluetoothStackID, Word_t CID);
```

```
BTPSAPI_DECLARATION int BTPSAPI L2CA_LE_Grant_Credits( unsigned int
    BluetoothStackID, Word_t CID, Word_t CreditsToGrant);
```

10.2 Structure Changes

```

typedef __PACKED_STRUCT_BEGIN__ struct _tagL2CAP_LE_Credit_Based_Connection_Request_t
{ L2CAP_Signal_Command_Header_t CommandHeader; NonAlignedWord_t LE_PSM;
NonAlignedWord_t SCID; NonAlignedWord_t
    MTU; NonAlignedWord_t MPS; NonAlignedWord_t InitialCredits; } __PACKED_STRUCT_END__
L2CAP_LE_Credit_Based_Connection_Request_t;

```

```

typedef __PACKED_STRUCT_BEGIN__ struct _tagL2CAP_LE_Credit_Based_Connection_Response_t
{ L2CAP_Signal_Command_Header_t CommandHeader; NonAlignedWord_t DCID;
NonAlignedWord_t MTU; NonAlignedWord_t
    MPS; NonAlignedWord_t InitialCredits; NonAlignedWord_t Result; }
__PACKED_STRUCT_END__ L2CAP_LE_Credit_Based_Connection_Response_t;

```

```

typedef __PACKED_STRUCT_BEGIN__ struct _tagL2CAP_LE_Flow_Control_Credit_t
{ L2CAP_Signal_Command_Header_t CommandHeader; NonAlignedWord_t DCID;
NonAlignedWord_t Credits; } __PACKED_STRUCT_END__
L2CAP_LE_Flow_Control_Credit_t;

```

```

typedef __PACKED_STRUCT_BEGIN__ struct _tagL2CAP_LE_Frame_Packet_t
{ L2CAP_Data_Packet_Header_t L2CAP_Data_Packet_Header; NonAlignedWord_t SDU_Length;
Byte_t Variable_Data[1];
} __PACKED_STRUCT_END__ L2CAP_LE_Frame_Packet_t;

```

```

typedef struct _tagL2CA_LE_Channel_Parameters_t { Word_t ChannelFlags;
    Word_t MaxSDUSize; Word_t MaxPDUSize; Word_t PDUQueueDepth; Word_t MaximumCredits; }
L2CA_LE_Channel_Parameters_t;

```

```

typedef struct _tagL2CA_LE_Connect_Indication_t { Word_t PSM; Word_t
    LCID; Byte_t Identifier; BD_ADDR_t BD_ADDR; Word_t MaxSDUSize; Word_t MaxPDUSize;
Word_t InitialCredits;
} L2CA_LE_Connect_Indication_t;

```



```
typedef struct _tagL2CA_LE_Connect_Confirmation_t { Word_t LCID; Word_t
    Result; Word_t MaxSDUSize; Word_t MaxPDUSize; Word_t InitialCredits; }
L2CA_LE_Connect_Confirmation_t;
```

```
typedef struct _tagL2CA_LE_Disconnect_Indication_t { Word_t LCID; Byte_t
    Reason; } L2CA_LE_Disconnect_Indication_t;
```

```
typedef struct _tagL2CA_LE_Disconnect_Confirmation_t { Word_t LCID;
    Word_t Result; } L2CA_LE_Disconnect_Confirmation_t;
```

```
typedef struct _tagL2CA_LE_Data_Indication_t { Word_t CID; Word_t CreditsConsumed;
    Word_t Data_Length; Byte_t *Variable_Data; } L2CA_LE_Data_Indication_t;
```

```
typedef struct _tagL2CA_LE_Channel_Buffer_Empty_Indication_t { Word_t
    CID; } L2CA_LE_Channel_Buffer_Empty_Indication_t;
```

```
typedef struct _tagL2CA_Event_Data_t
{
    L2CA_Event_Type_t L2CA_Event_Type;
    DWord_t          Event_Data_Length;
    union
    {
        L2CA_Connect_Indication_t          *L2CA_Connect_Indication;
        L2CA_Connect_Confirmation_t        *L2CA_Connect_Confirmation;
        .
        .
        L2CA_LE_Connect_Indication_t        *L2CA_LE_Connect_Indication;
    } Event_Data;
} L2CA_Event_Data_t;
```

10.3 Enumeration changes

```
typedef enum
{
    etConnect_Indication,
    etConnect_Confirmation,
    .
    .
    .
    etDisconnect_Confirmation,
    etLE_Connect_Indication, etLE_Connect_Confirmation, etLE_Disconnect_Indication,
    etLE_Disconnect_Confirmation, etLE_Data_Indication,
    etLE_Channel_Buffer_Empty_Indication
} L2CA_Event_Type_t;
```

10.4 Macro Changes

```
#define L2CAP_LE_PSM_VALID_PSM(_x) (((_x)>=L2CAP_LE_PSM_MINIMUM_PSM)
    && ((_x)<=L2CAP_LE_PSM_MAXIMUM_PSM))
```

Revision History

Changes from Original (November 2016) to A Revision**Page**

-
- Updated Software Migration section. **3**
-

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2018, Texas Instruments Incorporated