EtherMind GATT DB Customization

1 Customizing GATT Database

To allow vendor defined GATT database or to extend existing GATT database, `GATT_DB_HAVE_STATIC_CONFIGURATION` should not be defined in the EtherMind stack library build.

This will ensure following are defined in the respective services gatt_db.c file.

```
--
#ifndef GATT_DB_HAVE_STATIC_CONFIGURATION
DECL_CONST UCHAR  gatt_service_count = GATT_SERVICE_COUNT;
DECL_CONST UCHAR  gatt_characteristic_count = GATT_CHARACTERISTIC_COUNT;

DECL_CONST UCHAR  gatt_db_max_type_count = GATT_DB_MAX_TYPE_COUNT;
DECL_CONST UCHAR  gatt_db_peer_val_arr_size = GATT_DB_PEER_VALUE_ARRAY_SIZE;
#endif /* GATT_DB_HAVE_STATIC_CONFIGURATION */
--
```

These are defined in the application (in gatt_db.c file) and used as externs in the EtherMind stack library. Linker takes care of using the appropriate definition, based on the application (and associated gatt_db.c) used.

Apart from these, following symbols/arrays are also defined by the application (typically in gatt_db.c file), which are also externally linked from the library.

```
      gatt_const_uuid_arr
      gatt_value_arr
      gatt_const_value_arr
      gatt_db_peer_specific_val_arr
      gatt_db_const_peer_specific_val_arr
      gatt_type_table
      gatt_db_attr_table
      gatt_characteristic
      gatt_service
```

Once application decides the content of the GATT database, most of its element remains same.
Accordingly wherever possible, for these count/size related variables and data structure arrays 'const' keyword is used.
'const' keywords are not used only for the ones, whose value can be changed locally or remotely.

Significance of these data structure/arrays in GATT Database are described below.

- GATT Database is a collection of services and list of services are realized through following data structure DECL_CONST GATT_DB_SERVICE gatt_service[GATT_SERVICE_COUNT]

  - "list" parameter describes the number of attributes in the service (which is typically 1).
  - "desc" parameter describes if the service requires security or not (GATT_DB_SER_NO_SECURITY_PROPERTY,GATT_DB_SER_SECURITY_LEVEL1,GATT_DB_SER_SECURITY_MODE1,GATT_DB_SER_ENCRYPT_KEY_SIZE_16 etc), if the service uuid is 16-bit or 128-bit format (GATT_DB_SER_128_BIT_UUID_FORMAT, default is 16-bit), if the service is to be used as a primary service or secondary service(GATT_DB_SER_SECONDARY_SERVICE_PROPERTY, default is primary service), if the services is discoverable only on a particular transport or it is discoverable over any

transport(GATT_DB_SER_SUPPORT_ANY_LINK_TYPE,GATT_DB_SER_SUPPORT_BR_LINK_TYPE,GATT_DB_SER_SUPPORT_LE_LINK_TYPE etc).

  - Start and End attribute handles of the service. There is no hole in the attribute handles in this new GATT database. And the first attribute handle of the first service record starts from 0x0001.
  - One service record is a collection of multiple characteristics. "char_start_index" start index of the characteristic grouped under this service with reference to "gatt_characteristic" data structure
  - "char_end_index" end index of the characteristic grouped under this service with reference to "gatt_characteristic" data structure

One or more characteristics are part of a service.

- DECL_CONST GATT_DB_CHARACERISTIC gatt_characteristic[GATT_CHARACTERISTIC_COUNT]

  - This data structure holds the characteristic specific information;
  - "list" parameter describes the number of attributes in the characteristics including the characteristic definition
  - "start_handle" parameter describes the start handle of the characteristic
  - "service_id" parameter describes the service to which this characteristic belong to with reference to the "gatt_service" data structure.

All the characteristics and services are collection of attributes. Each attribute is identified by attribute handle and have specific properties and associated values.

- DECL_CONST GATT_DB_DESC_DATA gatt_db_attr_table[GATT_DB_MAX_ATTRIBUTES]

  - This data structure holds the attribute information of all the services, characteristics and characteristic descriptors of the profile.
  - The attribute handle is realized using the index of this structure.
  - For a given service, with an attribute handle "X"
    - Attribute Handle X contains Service Declaration
    - Attribute Handle X + 1 contains Characteristic Declaration
    - Attribute Handle X + 2 contains Characteristic Value
    - Attribute Handle X + 3 contains Characteristic Descriptors if any,

  - "desc_property" parameter describes the attribute permissions (Ex. Read, Write, Write Without Response, Notification, Indication).
  - "aux_property" parameter describes the auxiliary permissions (Ex, Fixed length, Peer specific value, 128bit UUID, Authorization requirements etc.). Refer NOTE section below for further details.
  - "length" parameter describes the length of the attibute value. The length of the attribute value depends on the "aux property" and also the "data offset" (which points to the reference of location or placement of the attribute value). If the "aux property" does not have GATT_DB_FIXED_LENGTH_PROPERTY associated with it, then an additional 1 Byte is added to the "length" of the attribute. This "additional byte" which is located at the first byte of the attribute value (which is pointed by "data offset") holds the current length of the attribute value. If the "data offset" field is set to "NULL" then the "length" is set to "0". Refer NOTE section below for further details.
  - "next" parameter describes the next location/index of the current attribute type in the "gatt_db_attr_table"
  - "uuid_offset" parameter describes the offset of the UUID
  - "data_offset" parameter gives the offset for the attribute value with reference to gatt_value_arr/gatt_const_value_arr/gatt_db_peer_specific_val_arr/NULL. Refer NOTE section below for further details.

NOTE:

Each attribute value can be of the following types:
1. The Value of which is always constant - Category A.
2. The Value of which is modifiable (by either local application or peer device) - Category B.
3. The Value of which has to be maintained per Peer device instance - Category C.
4. The Value of which is maintained directly at the local application - Category D.

Depending on the category into which a given attribute fits, its value will be allotted position in corresponding data structures i.e. its "data offset" will be in:
1. "gatt_const_value_arr" for all Category A attribute.
2. "gatt_value_arr" for all Category B attribute.
3. "gatt_db_peer_specific_val_arr" for all Category C attribute.
4. For all Category D attribute, the value will not be placed in GATT Database, the above application needs to manage these attribute/attribute values.

Also, depending on the category into which a given attribute fits, following rules apply:

1. For all Category A attributes "GATT_DB_FIXED_LENGTH_PROPERTY" as "aux property" and "GATT_DB_READ" as "desc property" is mandatory. The "desc property" cannot be "GATT_DB_WRITE".

2. For a Category B and C attribute, if "GATT_DB_FIXED_LENGTH_PROPERTY" is not present in "aux property" then "length" should be incremented by 1 to accommodate the header Byte. Correspondingly, the "Header Byte" of the attribute value in the associated data structure (gatt_value_arr/gatt_db_peer_specific_val_arr) location should be updated with current length of the attribute. Also, "desc property" shall be "GATT_DB_WRITE" if the attribute value is writeable by peer device otherwise, it is only “GATT_DB_READ”. These are both readable and writeable irrespective of "desc property" for the local application.

3. For a Category D attribute, the "length" field is always "0". The “desc property” can be “GATT_DB_WRITE” or “GATT_DB_READ” depending on the use case specified by the higher layer/application.

Mapping of attribute types or UUIDs with their occurrence in attribute handles for faster search.

- DECL_CONST GATT_ATTR_TYPE gatt_type_table[GATT_DB_MAX_TYPE_COUNT]

  - This data structure holds the first and last occurrence/index information of the attribute type/UUID.
  - "uuid_offset" parameter is the reference for the UUID defined in the "gatt_const_uuid_arr" data structure.
  - "fo" parameter is the reference for the first occurrence of the above Atribute type/UUID in the "gatt_db_attr_table" data structure.
  - "lo" parameter is the reference for the last occurrence of the above Attribute type/UUID in the "gatt_db_attr_table" data structure.

If there is only one occurrence of the attribute type/UUID in the "gatt_db_attr_table" then this parameter will be set to "ATT_INVALID_ATTR_HANDLE_VAL".

Flattened list of UUID values (both 16 and 128 bit), so that these UUID values can be mapped/indexed from other data structures.
Also removing the need to use the same 16-bit or 128-bit UUIDs multiple times, rather use shorted indexed value in place of those.

- DECL_CONST UCHAR gatt_const_uuid_arr[GATT_UUID_ARRAY_SIZE]

  - This data structure holds the 16bit or 128bit UUIDs of the Services, Characteristics and Characteristic Descriptors of a profile.

  - The data structure is defined as an unsigned char array to easily refer the specific UUID in other GATT data structures.
  - The UUIDs below are present across all the GATT DBs followed by the Profile specific UUIDs.

```
/* 0 - Primary Service UUID */
0x00, 0x28,
/* 2 - Secondary Service UUID */
0x01, 0x28,
/* 4 - Include UUID */
0x02, 0x28,
/* 6 - Characteristic UUID */
0x03, 0x28,
/* 8 - Characteristic Extended Properties UUID */
0x00, 0x29,
/* 10 - Characteristic User Description UUID */
0x01, 0x29,
/* 12 - Client Configuration UUID */
0x02, 0x29,
/* 14 - Server Configuration UUID */
0x03, 0x29,
/* 16 - Characteristic Format UUID */
0x04, 0x29,
/* 18 - Characteristic Aggregate Format UUID */
0x05, 0x29,
```

  - Adding custom service/characteristics UUIDs to the gatt_const_uuid_arr data structure

```
/* 20 - 16Bit Custom UUID 0x1234 */
0x34, 0x12,
/* 22 - 128Bit Custom UUID 0x1234567890ABCDEF1234567890ABCDEF */
0xEF, 0xCD, 0xAB, 0x90, 0x78, 0x56, 0x34, 0x12,
0xEF, 0xCD, 0xAB, 0x90, 0x78, 0x56, 0x34, 0x12,
/* 38 - 16Bit Custom UUID 0xABCD */
0xCD, 0xAB,
```

- ** DECL_CONST UCHAR gatt_db_const_peer_specific_val_arr[GATT_DB_PEER_VALUE_ARRAY_SIZE]

  - This data structure holds the characteristic and characteristic descriptor attribute values which are readable, writeable and peer specific.
  - This data structure is populated for a single instance of peer device and based on the number of peer/remote devices supported by stack the "UCHAR gatt_db_peer_specific_val_arr[GATT_DB_MAX_PEER_CONFIGURATION]" data structure is populated during the GATT database initialization.
  - After this initial population of peer specific values with this specific format, content of gatt_db_peer_specific_val_arr[] changes based on different values configured by different peer devices.
  - For attribute values which point to attributes in gatt_db_attr_table, if "GATT_DB_FIXED_LENGTH_PROPERTY" is not present in "aux property" in the corresponding entries in gatt_db_attr_table, the "Header Byte" of the attribute value should be updated with current length.

- UCHAR gatt_value_arr[GATT_VALUE_ARRAY_SIZE]

  - This data structure holds the characteristic and characteristic descriptor attribute values which are readable and writeable but not peer/client specific.

- For attribute values which point to attributes in gatt_db_attr_table, if "GATT_DB_FIXED_LENGTH_PROPERTY" is not present in "aux property" in the corresponding entries in gatt_db_attr_table, the "Header Byte" of the attribute value should be updated with current length.

1.1 Illustration with example

Mapping of the GATT Database structures for the mandatory GATT, GAP Services and optional Battery Service.

The mapping of the data structures is realized using the attribute information added as comment at the start of each GATT database.

```
Handle  |      ATT_Type      |    Permission    |  ATT_VALUE
========+====================+==================+=====================

**********************************************************************
GAP Service Declaration
========+====================+==================+=====================
0x0001  |       0x2800       | Rd               | 0x1800               --> GAP Service Start Handle
--------+--------------------+------------------+---------------------
Device Name Characteristic Declaration
--------+--------------------+------------------+---------------------
0x0002  |       0x2803       | Rd               | 0x02, 0x0003, 0x2A00 --> Device Name
Characteristic Start Handle
--------+--------------------+------------------+---------------------
Device Name Characteristic Value
--------+--------------------+------------------+---------------------
0x0003  |       0x2A00       | Rd               | MindtreePhone
--------+--------------------+------------------+---------------------
Appearance Characteristic Declaration
--------+--------------------+------------------+---------------------
0x0004  |      0x2803        | Rd               | 0x02, 0x0005, 0x2A01  --> Appearance
Characteristic Start handle
--------+--------------------+------------------+---------------------
Appearance Characteristic Value
--------+--------------------+------------------+---------------------
0x0005  |       0x2A01       | Rd               | 64.0                 --> GAP Service End Handle
**********************************************************************

**********************************************************************
GATT Service Declaration
========+====================+==================+=====================
0x0006  |       0x2800       | Rd               | 0x1801               --> GATT Service Start
Handle
--------+--------------------+------------------+---------------------
Service Changed Characteristic Declaration
--------+--------------------+------------------+---------------------
0x0007  |       0x2803       | Rd               | 0x20, 0x0008, 0x2A05 --> Service Changed
Characteristic Start Handle
--------+--------------------+------------------+---------------------
Service Changed Characteristic Value
--------+--------------------+------------------+---------------------
0x0008  |       0x2A05       | Ind              | N/A
--------+--------------------+------------------+---------------------
Service Changed CCCD (Client Characteristic Configuration Descriptor)
--------+--------------------+------------------+---------------------
0x0009  |       0x2902       | Rd, Wr           | 0x0000               --> GATT Service End Handle
**********************************************************************

**********************************************************************
Battery Service Declaration
========+====================+==================+=====================
0x000A  |       0x2800       | Rd                | 0x180F               --> Battery Service Start
Handle
--------+--------------------+-------------------+---------------------
Battery Level Characteristic Declaration
--------+--------------------+-------------------+---------------------
0x000B  |       0x2803       | Rd                | 0x12, 0x000C, 0x2A19 --> Battery Level
Characteristic Start Handle
--------+--------------------+-------------------+---------------------
Battery Level Characteristic Value
--------+--------------------+-------------------+---------------------
0x000C  |       0x2A19       | Rd, Ntf           | 100.0
--------+--------------------+-------------------+---------------------
Battery Level CCCD (Client Characteristic Configuration Descriptor)
--------+--------------------+-------------------+---------------------
0x000D  |       0x2902       | Rd, Wr            | 0x0000
--------+--------------------+-------------------+---------------------
Battery Level Presentation Format Descriptor
--------+--------------------+-------------------+---------------------
0x000E  |       0x2904       | Rd                | VALUE                --> Battery Service End
Handle
**********************************************************************

GATT UUID array from the above attribute table
----------------------------------------------
DECL_CONST UCHAR gatt_const_uuid_arr[GATT_UUID_ARRAY_SIZE] =
{
------------------------------------------------------------
    /* 0 - Primary Service UUID */                            |
    0x00, 0x28,                                               |
                                                              |
    /* 2 - Secondary Service UUID */                          |
    0x01, 0x28,                                               |
                                                              |
    /* 4 - Include UUID */                                    |
    0x02, 0x28,                                               |
                                                              |
    /* 6 - Characteristic UUID */                             |
    0x03, 0x28,                                               |
                                                              |
    /* 8 - Characteristic Extended Properties UUID */         |
    0x00, 0x29,                                               |    These set of UUIDs are present
across all databases
                                                              |
    /* 10 - Characteristic User Description UUID */           |
    0x01, 0x29,                                               |
                                                              |
    /* 12 - Client Configuration UUID */                      |
    0x02, 0x29,                                               |
                                                              |
    /* 14 - Server Configuration UUID */                      |
    0x03, 0x29,                                               |
                                                              |
    /* 16 - Characteristic Format UUID */                     |
    0x04, 0x29,                                               |
                                                              |
    /* 18 - Characteristic Aggregate Format UUID */           |
    0x05, 0x29,                                               |
------------------------------------------------------------
    /* GAP Service UUID Index "20" - GAP Service UUID */
    0x00, 0x18,

    /* DeviceName Characteristic UUID Index "22" - DeviceName Characteristic UUID */
    0x00, 0x2A,

    /* Appearance Characteristic UUID Index "24" - Appearance Characteristic UUID */
    0x01, 0x2A,

    /* GATT Service UUID Index "26" - GATT Service UUID */
    0x01, 0x18,

    /* Service Changed Characteristic UUID Index "28" - Service Changed Characteristic UUID */
    0x05, 0x2A,

    /* Battery Service UUID Index "30" - Battery Service UUID */
    0x0F, 0x18,

    /* BatteryLevel Characteristic UUID Index "32" - BatteryLevel Characteristic UUID */
    0x19, 0x2A,
}
```

Mapping of the "gatt_service" data structure with the above attributes table
----

```
DECL_CONST GATT_DB_SERVICE gatt_service[GATT_SERVICE_COUNT] =
{

    /* GAP Service Index "0" - GAP */
    {
        {
            /* Number of attributes in Service */
            1, (Fixed to "1" for all services, added for future use and has no significance for
now)
        },

        /* Service Description */
        GATT_DB_SER_NO_SECURITY_PROPERTY |
        GATT_DB_SER_SUPPORT_ANY_LINK_TYPE, (No Security permissions and service accessible on LE or
BREDR link)

        /* Service Start Handle */
        0x0001, (GAP Service Start Handle)

        /* Service End Handle */
        0x0005, (GAP Service End Handle)

        /* Characteristic Start Index */
        0, (Device Name Characteristic Index)

        /* Characteristic End Index */
        1 (Appearance Characteristic Index)
    },

    /* GATT Service Index "1" - GATT */
    {
        {
            /* Number of attributes in Service */
            1, (Fixed to "1" for all services, added for future use and has no significance for
now)
        },

        /* Service Description */
        GATT_DB_SER_NO_SECURITY_PROPERTY |
        GATT_DB_SER_SUPPORT_ANY_LINK_TYPE, (No Security permissions and service accessible on LE or
BREDR link)

        /* Service Start Handle */
        0x0006, (GATT Service Start Handle)

        /* Service End Handle */
        0x0009, (GATT Service End Handle)

        /* Characteristic Start Index */
        2, (Service Changed Characteristic Index)

        /* Characteristic End Index */
        2 (Service Changed Characteristic Index)
    },

    /* Battery Service Index "2" - Battery */
    {
        {
            /* Number of attributes in Service */
            1, (Fixed to "1" for all services, added for future use and has no significance for
now)
        },

        /* Service Description */
        GATT_DB_SER_NO_SECURITY_PROPERTY |
        GATT_DB_SER_SUPPORT_ANY_LINK_TYPE, (No Security permissions and service accessible on LE or
BREDR link)

        /* Service Start Handle */
        0x000A, (Battery Service Start Handle)

        /* Service End Handle */
        0x000E, (Battery Service End Handle)

        /* Characteristic Start Index */
        3, (Battery Level Characteristic Index)

        /* Characteristic End Index */
        3 (Battery Level Characteristic Index)
    },
}
```

Mapping of "gatt_characteristic" data structure with above attributes table

```
-------------------------------------------------------------------------------------------
DECL_CONST GATT_DB_CHARACERISTIC gatt_characteristic[GATT_CHARACTERISTIC_COUNT] =
{

    /* Device Name Characteristic Index "0" - DeviceName */
    {
        {
            /* Number of attributes in characteristic including the definition */
            2, (Device Name Characteristic Declaration + Device Name Characteristic Value)
        },

        /* Characteristic Start Handle */
        0x0002, (Device Name Characteristic Start Handle)

        /* Characteristic Service Index */
        0, (GAP Service Index)
    },

    /* Appearance Characteristic Index "1" - Appearance */
    {
        {
            /* Number of attributes in characteristic including the definition */
            2, (Appearance Characteristic Declaration + Appearance Characteristic Value)
        },

        /* Characteristic Start Handle */
        0x0004, (Appearance Characteristic Start Handle)

        /* Characteristic Service Index */
        0, (GAP Service Index)
    },

    /* Service Changed Characteristic Index "2" - Service Changed */
    {
        {
            /* Number of attributes in characteristic including the definition */
            3, (Service Changed Characteristic Declaration + Service Changed Characteristic Value +
Service Changed CCCD)
        },

        /* Characteristic Start Handle */
        0x0007, (Service Changed Characteristic Start Handle)

        /* Characteristic Service Index */
        1, (GATT Service Index)
    },
    /* 3 - BatteryLevel */
    {
        {
            /* Number of attributes in characteristic including the definition */
            4, (Battery Level Characteristic Declaration + Battery Level Characteristic Value +
Battery Level CCCD + Battery Level Presentation Format)
        },

        /* Characteristic Start Handle */
        0x000B, (Battery Level Characteristic Start Handle)

        /* Characteristic Service Index */
        2, (Battery Servie Index)
    },
}
```

Mapping of "gatt_db_attr_table" data structure with above attributes table

```
---------------------------------------------------------------------------------------------
DECL_CONST GATT_DB_DESC_DATA gatt_db_attr_table[GATT_DB_MAX_ATTRIBUTES] =
{
    The Attribute Handle is realized using the index of gatt_db_attr_table,
    since the valid range for attribute handle is 0x0001 to 0xFFFF therfore
    the first index "0x0000" is not used and filled up with dummy values.

    /* Dummy */
    {
        /* Property */
        GATT_DB_READ,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,

        /* Value Length */
        ATT_INVALID_ATTR_HANDLE_VAL,

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL,

        /* UUID Offset */
        ATT_INVALID_ATTR_HANDLE_VAL,

        /* Value */
        NULL,
    },

    /* Handle - 0x0001 */
    /* GAP Service Index "0" - GAP - Service Declaration */
    {
        /* Property */
        GATT_DB_READ, (Read only attribute permission for Service Definition)

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY, (2/16 Bytes fixed attribute value)

        /* Value Length */
        2, (Value length of the attribute value i.e. the service specific UUID 2bytes for 16Bit
UUID and 16bytes for 128Bit UUID)

        /* Next Attribute Type Index */
        6, (Next Primary Service Declaration index at Handle "0x0006" for the UUID offset 0 in this
data structure)

        /* UUID Offset */
        0, (Primary Service UUID index from "gatt_const_uuid_arr" array)

        /* Value */
        (UCHAR *)(gatt_const_uuid_arr + 20), ("20" GAP Service UUID index from
"gatt_const_uuid_arr" array)
    },
    /* Handle - 0x0002 */
    /* Device Name Characteristic Index "0" - DeviceName - Characteristic Declaration */
    {
        /* Property */
        GATT_DB_READ, (Read only attribute permission for Characteristic Definition)

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY, (5/19 Bytes fixed attribute value)

        /* Value Length */
        5, (1Byte of Characteristic Property + 2Bytes of Characteristic Value Handle + 2/16bBytes
of Characteristic UUID)

        /* Next Attribute Type Index */
        4, (Next Characteristic Declaration index at Handle "0x0004" for the UUID Offset "6" in
this data structure)

        /* UUID Offset */
        6, (Characteristic UUID index from "gatt_const_uuid_arr" array)

        /* Value */
        (UCHAR *)(gatt_const_value_arr + 0), (Since the value is only readable and not peer
specific the value is stored in gatt_const_value_arr and referred
        accordingly)
    },

    /* Handle - 0x0003 */
    /* 0 - DeviceName - Characteristic Value */
    {
        /* Property */
        GATT_DB_READ, (Attribute property)

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY, (Fixed length property)

        /* Value Length */
        13, ("MindtreePhone" attribute value length)

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL, (No more occurrence of Device Name Characteristic UUID in this
table)

        /* UUID Offset */
        22, (Device Name Characteristic UUID index from "gatt_const_uuid_arr" array)

        /* Value */
        (UCHAR *)(gatt_const_value_arr + 5), (Since the value is only readable and not peer
specific the value is stored in gatt_const_value_arr and referred
        accordingly)
    },

    /* Handle - 0x0004 */
    /* 1 - Appearance - Characteristic Declaration */
    {
        /* Property */
        GATT_DB_READ,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,

        /* Value Length */
        5,

        /* Next Attribute Type Index */
        7,

        /* UUID Offset */
        6,

        /* Value */
        (UCHAR *)(gatt_const_value_arr + 19),
    },

    /* Handle - 0x0005 */
    /* 1 - Appearance - Characteristic Value */
    {
        /* Property */
        GATT_DB_READ,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,

        /* Value Length */
        2,

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL,

        /* UUID Offset */
        24,

        /* Value */
        (UCHAR *)(gatt_const_value_arr + 24),
    },

    /* Handle - 0x0006 */
    /* 1 - GATT - Service Declaration */
    {
        /* Property */
        GATT_DB_READ,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,

        /* Value Length */
        2,

        /* Next Attribute Type Index */
        10,

        /* UUID Offset */
        0,

        /* Value */
        (UCHAR *)(gatt_const_uuid_arr + 26),
    },

    /* Handle - 0x0007 */
    /* 2 - Service Changed - Characteristic Declaration */
    {
        /* Property */
        GATT_DB_READ,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,

        /* Value Length */
        5,

        /* Next Attribute Type Index */
        11,

        /* UUID Offset */
        6,

        /* Value */
        (UCHAR *)(gatt_const_value_arr + 26),
    },

    /* Handle - 0x0008 */
    /* 2 - Service Changed - Characteristic Value */
    {
        /* Property */
        GATT_DB_CHAR_INDICATE_PROPERTY,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,

        /* Value Length */
        0, (Since this is to be handled/maintained by the application)

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL,

        /* UUID Offset */
        28,

        /* Value */
        NULL, (Since this is to be handled/maintained by the application)
    },

    /* Handle - 0x0009 */
    /* 2 - Service Changed - CCD */
    {
        /* Property */
        GATT_DB_READ |
        GATT_DB_WRITE,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY |
        GATT_DB_PEER_SPECIFIC_VAL_PROPERTY,

        /* Value Length */
        2,

        /* Next Attribute Type Index */
        13,

        /* UUID Offset */
        12,

        /* Value */
        gatt_db_peer_specific_val_arr + 0,
    },

    /* Handle - 0x000A */
    /* 2 - Battery - Service Declaration */
    {
        /* Property */
        GATT_DB_READ,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,

        /* Value Length */
        2,

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL, (No more Primary Service Declaration after the Battery
Service)

        /* UUID Offset */
        0,

        /* Value */
        (UCHAR *)(gatt_const_uuid_arr + 30),
    },

    /* Handle - 0x000B */
    /* 3 - BatteryLevel - Characteristic Declaration */
    {
        /* Property */
        GATT_DB_READ,
        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,

        /* Value Length */
        5,

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL, (No more Characteristic declaration after the battery level
Characteristic)

        /* UUID Offset */
        6,

        /* Value */
        (UCHAR *)(gatt_const_value_arr + 31),
    },

    /* Handle - 0x000C */
    /* 3 - BatteryLevel - Characteristic Value */
    {
        /* Property */
        GATT_DB_READ |
        GATT_DB_CHAR_NOTIFY_PROPERTY,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY |
        GATT_DB_PEER_SPECIFIC_VAL_PROPERTY,

        /* Value Length */
        1,

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL,

        /* UUID Offset */
        32,

        /* Value */
        gatt_db_peer_specific_val_arr + 2,
    },

    /* Handle - 0x000D */
    /* 3 - BatteryLevel - CCD */
    {
        /* Property */
        GATT_DB_READ |
        GATT_DB_WRITE,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY |
        GATT_DB_PEER_SPECIFIC_VAL_PROPERTY,

        /* Value Length */
        2,

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL, (No more client characteristic descriptor after this)

        /* UUID Offset */
        12,

        /* Value */
        gatt_db_peer_specific_val_arr + 3,
    },

    /* Handle - 0x000E */
    /* Characteristic 3 - BatteryLevel - Presentation Format */
    {
        /* Property */
        GATT_DB_READ,
        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,

        /* Value Length */
        7,

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL,

        /* UUID Offset */
        16,

        /* Value */
        (UCHAR *)(gatt_const_value_arr + 36),
    },
}
```

Mapping of "gatt_type_table" data structure with above attribute table

```
-----------------------------------------------------------------------
DECL_CONST GATT_ATTR_TYPE gatt_type_table[GATT_DB_MAX_TYPE_COUNT] =
{
    {
        /* UUID Offset */
        0, (Primary Servie UUID index from the gatt_const_uuid_arr array)

        /* First Occurrence for Type */
        1, (First occurrence of the primary service attribute type in gatt_db_attr_table data
structure)

        /* Last Occurrence for Type */
        10, (Last occurrence of the primary service attribute type in gatt_db_attr_table data
structure)

    },
    {
        /* UUID Offset */
        6,

        /* First Occurrence for Type */
        2,

        /* Last Occurrence for Type */
        11,

    },
    {
        /* UUID Offset */
        12,

        /* First Occurrence for Type */
        9,

        /* Last Occurrence for Type */
        13,

    },
    {
        /* UUID Offset */
        16,

        /* First Occurrence for Type */
        14,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        20,
        /* First Occurrence for Type */
        1,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        22,

        /* First Occurrence for Type */
        3,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        24,

        /* First Occurrence for Type */
        5,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        26,

        /* First Occurrence for Type */
        6,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        28,

        /* First Occurrence for Type */
        8,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        30,

        /* First Occurrence for Type */
        10,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        32,

        /* First Occurrence for Type */
        12,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
};
```

The below attribute table contains a vendor specific service with a single characteristic and client characteristic descriptor.
Now we will capture the changes in the above GATT data structures/array after appending the Vendor Specific service to the above attribute table consisting of GATT, GAP and Battery Service.

```
Tx Service Declaration
========+====================+==================+=====================
0x000F  |       0x2800       | Rd               | 0x1234567890ABCDEF   --> Tx Service Start Handle
        |                    |                  |   1234567890ABCDEF
--------+--------------------+------------------+---------------------
Tx Data Port Characteristic Declaration
--------+--------------------+------------------+---------------------
0x0010  |       0x2803       | Rd               | 0x20, 0x0011,
        |                    |                  | 0x1122334455667788   --> Tx Data Port
Characteristic Start Handle
        |                    |                  |   9900AABBCCDDEEFF
--------+--------------------+------------------+---------------------
Tx Data Port Characteristic Value
--------+--------------------+------------------+---------------------
0x0011  | 0x1122334455667788 | Ind              | N/A
        |  99000AABBCCDDEEFF |                  |
--------+--------------------+------------------+---------------------
Tx Data Port CCCD
--------+--------------------+------------------+---------------------
0x0012  |       0x2902       | Rd, Wr           | 0x0000               --> Tx Service End Handle
```

The Mapping of the GATT data structures will be from the point where changes are required and if there are no changes required the new values will just be appended to the previous GATT data structures/arrays.

The following GATT data structures can be just appended with the new attribute information
- gatt_const_uuid_arr
- gatt_service
- gatt_characteristic
- gatt_value_arr
- gatt_const_value_arr
- gatt_db_const_peer_specific_val_arr

```
DECL_CONST UCHAR gatt_const_uuid_arr[GATT_UUID_ARRAY_SIZE] =
{

    + Appending the new UUIDs at the end.

    /* Tx Service UUID index "34" - Vendor Specific - Tx Service UUID */
    0xEF, 0xCD, 0xAB, 0x90, 0x78, 0x56, 0x34, 0x12,
    0xEF, 0xCD, 0xAB, 0x90, 0x78, 0x56, 0x34, 0x12

    /* Tx Data Port Characteristic UUID index "50" - Vendor Specific - Tx Data Port Characteristic
UUID */
    0xFF, 0xEE, 0xDD, 0xCC, 0xBB, 0xAA, 0x00, 0x99,
    0x88, 0x77, 0x66, 0x55, 0x44, 0x33, 0x22, 0x11
}

DECL_CONST GATT_DB_SERVICE gatt_service[GATT_SERVICE_COUNT] =
{
    + Appending new service information

    /* Tx Service Index "3" - Vendor Specific - Tx */
    {
        {
            /* Number of attributes in Service */
            1, (Fixed to "1" for all services, added for future use and has no significance for
now)
        },

        /* Service Description */
        GATT_DB_SER_NO_SECURITY_PROPERTY |
        GATT_DB_SER_SUPPORT_ANY_LINK_TYPE, (No Security permissions and service accessible on LE or
BREDR link)

        /* Service Start Handle */
        0x000F, (Tx Service Start Handle)

        /* Service End Handle */
        0x0012, (Tx Service End Handle)

        /* Characteristic Start Index */
        4, (Tx Data Port Characteristic Index)

        /* Characteristic End Index */
        4 (Tx Data Port Characteristic Index)
    },
}

DECL_CONST GATT_DB_CHARACERISTIC gatt_characteristic[GATT_CHARACTERISTIC_COUNT] =
{
    + Appending new characteristic information

    /* Tx Data Port Characteristic Index "4" - Vendor Specific - Tx Data Port */
    {
        {
            /* Number of attributes in characteristic including the definition */
            3, (Tx Data Port Characteristic Declaration + Tx Data Port Characteristic Value + Tx
Data Port CCCD)
        },

        /* Characteristic Start Handle */
        0x0010, (Tx Data Port Characteristic Start Handle)

        /* Characteristic Service Index */
        3, (Tx Service Index)
    },
}
```

The follwoing GATT data structure needs to be updated when adding new Attribute information
- gatt_db_attr_table
- gatt_type_table

```
DECL_CONST GATT_DB_DESC_DATA gatt_db_attr_table[GATT_DB_MAX_ATTRIBUTES] =
{
```

No changes required til the handle/index 0x0009 in this data strucutre. The last service (i.e. Battery in this case) needs to be updated accordingly before adding new attribute information.
The "next" is the only parameter that requires an update as there can be similar attribute types previously defined including Primary Service, Characteristic declarations and Characteristic descriptor if any.

```
    /* Handle - 0x000A */
    /* 2 - Battery - Service Declaration */
    {
        /* Property */
        GATT_DB_READ,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,
        /* Value Length */
        2,

***************************************************************************************************
***
        /* Next Attribute Type Index */
        15, (Update the next Service declaration handle/index in place of
ATT_INVALID_ATTR_HANDLE_VAL)
***************************************************************************************************
***

        /* UUID Offset */
        0,

        /* Value */
        (UCHAR *)(gatt_const_uuid_arr + 30),
    },

    /* Handle - 0x000B */
    /* 3 - BatteryLevel - Characteristic Declaration */
    {
        /* Property */
        GATT_DB_READ,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,

        /* Value Length */
        5,

***************************************************************************************************
******
        /* Next Attribute Type Index */
        16, (Update next Characteristic declaration handle/index in place of
ATT_INVALID_ATTR_HANDLE_VAL)
***************************************************************************************************
******

        /* UUID Offset */
        6,

        /* Value */
        (UCHAR *)(gatt_const_value_arr + 31),
    },

    /* Handle - 0x000C */
    /* 3 - BatteryLevel - Characteristic Value */
    {
        /* Property */
        GATT_DB_READ |
        GATT_DB_CHAR_NOTIFY_PROPERTY,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY |
        GATT_DB_PEER_SPECIFIC_VAL_PROPERTY,

        /* Value Length */
        1,

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL,

        /* UUID Offset */
        32,

        /* Value */
        gatt_db_peer_specific_val_arr + 2,
    },

    /* Handle - 0x000D */
    /* 3 - BatteryLevel - CCD */
    {
        /* Property */
        GATT_DB_READ |
        GATT_DB_WRITE,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY |
        GATT_DB_PEER_SPECIFIC_VAL_PROPERTY,

        /* Value Length */
        2,

******************************************************************************************
        /* Next Attribute Type Index */
        18, (Update next CCCD handle/index in place of ATT_INVALID_ATTR_HANDLE_VAL)
******************************************************************************************
        /* UUID Offset */
        12,

        /* Value */
        gatt_db_peer_specific_val_arr + 3,
    },

    /* Handle - 0x000E */
    /* Characteristic 3 - BatteryLevel - Presentation Format */
    {
        /* Property */
        GATT_DB_READ,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,

        /* Value Length */
        7,

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL,

        /* UUID Offset */
        16,

        /* Value */
        (UCHAR *)(gatt_const_value_arr + 36),
    },
***************************************************
New Service Attribute information starts from here
***************************************************
    /* Handle - 0x000F */
    /* 3 - Vendor Specific - Tx - Service Declaration */
    {
        /* Property */
        GATT_DB_READ,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,

        /* Value Length */
        2,

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL, (Last Primary Service Declaration)

        /* UUID Offset */
        0, (Primary Servie UUID index)

        /* Value */
        (UCHAR *)(gatt_const_uuid_arr + 34), ("34" Tx Service UUID index in the
gatt_const_uuid_arr)
    },

    /* Handle - 0x0010 */
    /* 4 - Vendor Specific - Tx Data Port - Characteristic Declaration */
    {
        /* Property */
        GATT_DB_READ,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY,

        /* Value Length */
        5,

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL, (Last Characteristic Declaration)

        /* UUID Offset */
        6, (Characteristic UUID index)

        /* Value */
        (UCHAR *)(gatt_const_value_arr + 43),
    },

    /* Handle - 0x0011 */
    /* 4 - Vendor Specific - Tx Data Port - Characteristic Value */
    {
        /* Property */
        GATT_DB_CHAR_INDICATE_PROPERTY,

        /* Auxiliary Property */
        GATT_DB_PEER_SPECIFIC_VAL_PROPERTY,

        /* Value Length */
        (1 + 1), (Max Allocated length + "Header Byte" header length)

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL,

        /* UUID Offset */
        50, (Tx Data Port Characteristic UUID index)

        /* Value */
        gatt_db_peer_specific_val_arr + 5,
    },

    /* Handle - 0x0012 */
    /* 4 - Vendor Specific - Tx Data Port - CCD */
    {
        /* Property */
        GATT_DB_READ |
        GATT_DB_WRITE,

        /* Auxiliary Property */
        GATT_DB_FIXED_LENGTH_PROPERTY |
        GATT_DB_PEER_SPECIFIC_VAL_PROPERTY,

        /* Value Length */
        2,

        /* Next Attribute Type Index */
        ATT_INVALID_ATTR_HANDLE_VAL,

        /* UUID Offset */
        12, (Client Characteristic Configuration Descriptor UUID index)

        /* Value */
        gatt_db_peer_specific_val_arr + 7, (Since the value is readable, writeable and also peer
specific it is stored in the gatt_db_peer_specific_val_arr
        with proper indexing)
    },
}

DECL_CONST GATT_ATTR_TYPE gatt_type_table[GATT_DB_MAX_TYPE_COUNT] =
{
    {
        /* UUID Offset */
        0,

        /* First Occurrence for Type */
        1,
**************************************************************************
        /* Last Occurrence for Type */
        15, (Update the last occurrence of the Primary Service Declaration)
**************************************************************************
    },
    {
        /* UUID Offset */
        6,

        /* First Occurrence for Type */
        2,
**************************************************************************
        /* Last Occurrence for Type */
        16, (Update the last occurrence of the Characteristic Declaration)
**************************************************************************
    },
    {
        /* UUID Offset */
        12,

        /* First Occurrence for Type */
        9,
***************************************************************************
        /* Last Occurrence for Type */
        18, (Update the last occurrence of the CCCD)
***************************************************************************
    },
    {
        /* UUID Offset */
        16,

        /* First Occurrence for Type */
        14,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        20,

        /* First Occurrence for Type */
        1,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        22,

        /* First Occurrence for Type */
        3,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        24,
        /* First Occurrence for Type */
        5,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        26,

        /* First Occurrence for Type */
        6,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        28,

        /* First Occurrence for Type */
        8,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        30,

        /* First Occurrence for Type */
        10,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
    {
        /* UUID Offset */
        32,

        /* First Occurrence for Type */
        12,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL,

    },
*********************************************************************
    {
        /* UUID Offset */
        34,

        /* First Occurrence for Type */
        15,

        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL, (If only one instance of the Tx Service
        UUID then ATT_INVALID_ATTR_HANDLE_VAL else update the handle/index
        value of the next instance from the gatt_db_attr_table)

    },
    {
        /* UUID Offset */
        50,

        /* First Occurrence for Type */
        17,
        /* Last Occurrence for Type */
        ATT_INVALID_ATTR_HANDLE_VAL, (If only one instance of the Tx Data Port
        Characteristic UUID then ATT_INVALID_ATTR_HANDLE_VAL else update the
        handle/index value of the next instance from the gatt_db_attr_table)

    },
*********************************************************************
};
```