make_gatt_services

make_gatt_services 工具使用 用户手册

Rev 1.10

This translated version is for reference only, and the English version shall prevail in case ofany discrepancy between the translated and English versions.

修改日志

文件版本	日期	描述
1.00	2018 / 9 / 05	工具使用说明，gatt_inc_generator.exe 版本 1.0.0.3
1.01	2019 / 1 / 16	完善工具使用说明
1.02	2019 / 12 / 03	完善工具使用说明
1.03	2020 / 03 / 03	完善工具使用说明，添加 128bit UUID 格式例子参考
1.04	2021 / 01 / 08	更新工具版本 1.0.0.4，增加关键字 GATT_HANDLE_BEGIN
1.10	2021 / 09 / 10	增加关键字 CHARACTERISTIC_USER_DESCRIPTION， 整理文档更新说明

1.1 目录结构

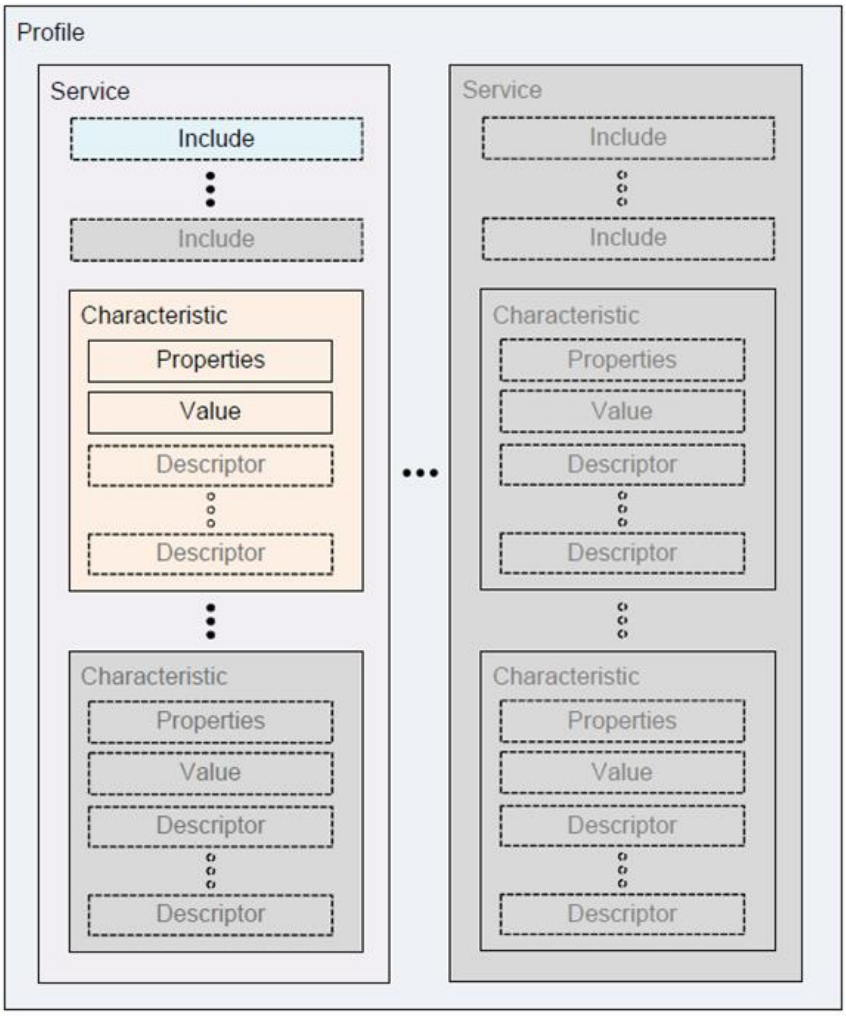
注意两个文件必须在同一级目录下才可使用。

1.2 使用说明

1.2.1 Profile 格式定义

make_gatt_services 工具主要用 SERVER 角色生成指定的 GATT Services profile，GATT 指定了数据交互的结构(Struct)；这个结构体定义了一些基本元素，如 Service、Characteristic 这些元素存在于 Attribute 中，具体内容可参照蓝牙官网描述，GATT Services profile 结构如下图：

GATT Profile Hierarchy



使用 make_gatt_services 工具生成 GATT profile 的步骤如下：

1、打开 gatt_profile.cfg 文件如下图，以数传 trans_data 应用的例子示例如下：

```
1.  PRIMARY_SERVICE, 1800
2.  CHARACTERISTIC,  2a00, READ | WRITE | DYNAMIC,
3.
4.  PRIMARY_SERVICE, ae30
5.  CHARACTERISTIC,  ae01, WRITE_WITHOUT_RESPONSE | DYNAMIC,
6.  CHARACTERISTIC,  ae02, NOTIFY,
7.
8.  CHARACTERISTIC,  ae03, WRITE_WITHOUT_RESPONSE | DYNAMIC,
9.  CHARACTERISTIC,  ae04, NOTIFY,
```

```
10.
11. CHARACTERISTIC,  ae05, INDICATE,
12.
13. CHARACTERISTIC,  ae10, READ | WRITE | DYNAMIC,
14.
15. PRIMARY_SERVICE, ae3a
16. CHARACTERISTIC,  ae3b, WRITE_WITHOUT_RESPONSE | DYNAMIC,
17. CHARACTERISTIC,  ae3c, NOTIFY,
```

上述图中配置 profile 描述：

第 1 行关键字 PRIMARY_SERVICE 声明了第 1 个服务 Service，服务的 uuid 是 1800；第 2 行是关键字 CHARACTERISTIC 声明这个服务带有 1 个属性 Characteristic，它的 uuid 是 2A00，带有属性的特性值是读 READ 和写 WRITE；另外 DYNAMIC 关键字是私有特性值，后面会有说明。

第 4 行关键字 PRIMARY_SERVICE 声明了第 2 个服务 Service，服务的 uuid 是 AE30；后面跟着关键字 CHARACTERISTIC 声明的多个属性 Characteristic，以及具有那些特性值。

同理，第 15 行再声明了第 3 个服务 Service 以及带的属性 Characteristic。

声明 Service 的特性值默认只有 READ，不需要额外配置。

属性 Characteristic 如果带有 NOTIFY 或者 INDICATE，工具会默认生成其对应的端属性配置 Client Characteristic Configuration(简称 CCC)数据，所以不用额外添加 CCC 配置。

2、用户想使用声明 128bit 的 UUID 的服务和属性，参考 UUID 的输入格式如下：

```
1.  PRIMARY_SERVICE, 1800
2.  CHARACTERISTIC,  2a00, READ | DYNAMIC,
3.
4.  PRIMARY_SERVICE, 0000F530-1212-EFDE-1523-785FEABCD123
5.  CHARACTERISTIC,  0000F531-1212-EFDE-1523-785FEABCD123, NOTIFY,
6.  CHARACTERISTIC,  0000F532-1212-EFDE-1523-785FEABCD123, WRITE_WITHOUT_RESPONSE | DYNAMIC,
7.  CHARACTERISTIC,  0000F534-1212-EFDE-1523-785FEABCD123, READ | WRITE | DYNAMIC,
```

3、关键字 GATT_HANDLE_BEGIN，可指定起始分配的 handle 值，要以服务 Service 为起始指定，示例如下：

！！！注意：指定 handle 时，整个 profile 分配生成的 handle 不能出现有重复

```
1.  //set handle start 0x0080
2.  GATT_HANDLE_BEGIN, 0080
3.
4.  PRIMARY_SERVICE, ae30
5.  CHARACTERISTIC,  ae01, WRITE_WITHOUT_RESPONSE | DYNAMIC,
6.  CHARACTERISTIC,  ae02, NOTIFY,
```

1.2.2 Service 和 Characteristic

服务 Service、特征 Characteristic 等定义解析可以参考蓝牙官网的说明，这里结合 trans_data 例子描述一下：

Declarations

Declarations are defined GATT profile attribute types.

The material contained on this page is informative only. Authoritative compliance information is contained in the applicable *Bluetooth*® specification.

Name	Uniform Type Identifier	Assigned Number	Specification
Characteristic Declaration	org.bluetooth.attribute.gatt.characteristic_declaration	0x2803	GCD
Include	org.bluetooth.attribute.gatt.include_declaration	0x2802	GCD
Primary Service	org.bluetooth.attribute.gatt.primary_service_declaration	0x2800	GCD
Secondary Service	org.bluetooth.attribute.gatt.secondary_service_declaration	0x2801	GCD

Service 以 PRIMARY_SERVICE 开始，后续跟 Service 的 Assigned Number，例如例子中的1800，代表 Generic Access，其具体定义可参考蓝牙官网 GATT Services 章节，如下图

GATT Services

Services are collections of characteristics and relationships to other services that encapsulate the behavior of part of a device.

All Service Assigned Numbers values on this page are normative. All other materials contained on this page is informative only. Authoritative compliance information is contained in the applicable *Bluetooth*® specification.

Name	Uniform Type Identifier	Assigned Number	Specification
Generic Access	org.bluetooth.service.generic_access	0×1800	GCD
Alert Notification Service	org.bluetooth.service.alert_notification	0×1811	GCD
Automation IO	org.bluetooth.service.automation_io	0×1815	GCD
Battery Service	org.bluetooth.service.battery_service	0×180F	GCD
Blood Pressure	org.bluetooth.service.blood_pressure	0×1810	GCD
Body Composition	org.bluetooth.service.body_composition	0×181B	GCD

CHARACTERISTIC 是包含一个 Characteristic 声明、Characteristic 属性、特性值的描述，例如例子中的 2A00，代表 Device Name。

Viewer

Last Modified: 2013-05-29

Name: Device Name

Type: org.bluetooth.characteristic.gap.device_name Download / View

Assigned Number: 0×2A00

READ | DYNAMIC，描述2A00的属性代表 Device Name 可读可变，可用属性的特性值可参考下图，填写时字母要全部大写。

支持的通用属性的特性值关键字有：

READ，WRITE_WITHOUT_RESPONSE，WRITE，NOTIFY，INDICATE，

AUTHENTICATED_SIGNED_WRITE，EXTENDED_PROPERTIES，官网的说明如下：

Service Characteristics

Overview	Properties	Security	Descriptors
Name: Device Name **Type:** org.bluetooth.characteristic.gap.device_name **Requirement:** Mandatory	Property / Requirement Read — Mandatory Write — Optional WriteWithoutResponse — Excluded SignedWrite — Excluded Notify — Excluded Indicate — Excluded WritableAuxiliaries — Excluded Broadcast — Excluded ExtendedProperties	None	None

常用属性的特性值使用注意：

(1)WRITE、WRITE_WITHOUT_RESPONSE 是 CLIENT 端（GATT 主机角色）向 SERVER 端（GATT 从机角色）执行的发送数据操作。而 NOTIFY 和 INDICATE 是 SERVER 端向 CLIENT 执行的发送数据操作。操作是以 handle 的方式标识。

(2)WRITE、INDICATE 的操作是需要对方响应回复命令，多用于数据交互带流控和可靠的传输方式。

(3)WRITE_WITHOUT_RESPONSE 、INDICATE 是不需要对方响应回复，多用于数据快速传输的方式。

另外增加私有的属性的特性值关键字有 DYNAMIC，AUTHENTICATION_REQUIRED，分别代表意思如下：

DYNAMIC

---数据可变处理，当有 READ，WRITE，WRITE_WITHOUT_RESPONSE，会产生对应的回调函数 read_callback 和 write_callback 处理，执行获取长度，填入对应的数据等操作。

AUTHENTICATION_REQUIRED

---需要配对加密认证标记，代表 CLIENT 端操作该属性的读写必需要经过配对加密后才能被允许，否则操作失败。SERVER 端可以使用该关键字，指示 CLIENT 端需要发起配对加密流程（SERVER 端常用的请求加密方式）。声明

定义示例如下：

```
1.  //PnP ID
2.  CHARACTERISTIC,  2a50, READ | AUTHENTICATION_REQUIRED | DYNAMIC,
```

另外通用和私有特性值可并列使用，例如 ae10有 READ，WRITE，DYNAMIC 属性那么填写

```
18. CHARACTERISTIC,  ae10, READ | WRITE | DYNAMIC,
```

注意：CHARACTERISTIC 中有 READ,WRITE,WRITE WITHOUT RESPONSE 属性，都必须添加 DYNAMIC 属性；不然无法应答对方读写数据的操作，除非默认数据长度为0。

本例子的 ae30 、ae3a 都是是自定义的 Service，内容的填写可参考以上的介绍，客户在开发的过程中可根据开发需求填写 profile。 标准的服务和属性对应 UUID（例如:服务1800和属性2A00）都可以在官网查询到。官网没有定义的 UUID，则认为是非标准的，可以由客户自定义使用。

1.2.3 Characteristic 的 Descriptors

上述已提到过 CHARACTERISTIC 如果有 NOTIFY，INDICATE，profile 工具结构会默认增加 GATT Descriptors 中的 Client Characteristic Configuration(UUID 2902,简称 CCC)，但 CCC 是需要 CLIENT 端通过 WRITE 的 Enable 操作开关，才能执行使用服务中的 NOTIFY 或 INDICATE 通知动作，向 CLIENT 端发送数据。

另外 profile 工具还支持 Descriptors 的关键字有 REPORT_REFERENCE 和 CHARACTERISTIC_USER_DESCRIPTION:

REPORT_REFERENCE

---UUID 是2908，多用于 HID 的 profile 定义，指定改，参考 hogp 的配置定义，其示例用法和官网的说明如下：

```
1.  //report1
2.  CHARACTERISTIC,  2a4d, READ | WRITE | NOTIFY | DYNAMIC,
3.  //report_id + report_type:1-input,2-ouput,3-feature
4.  REPORT_REFERENCE,01,01
```

Report Reference	org.bluetooth.descriptor.report_reference	0x2908	GSS

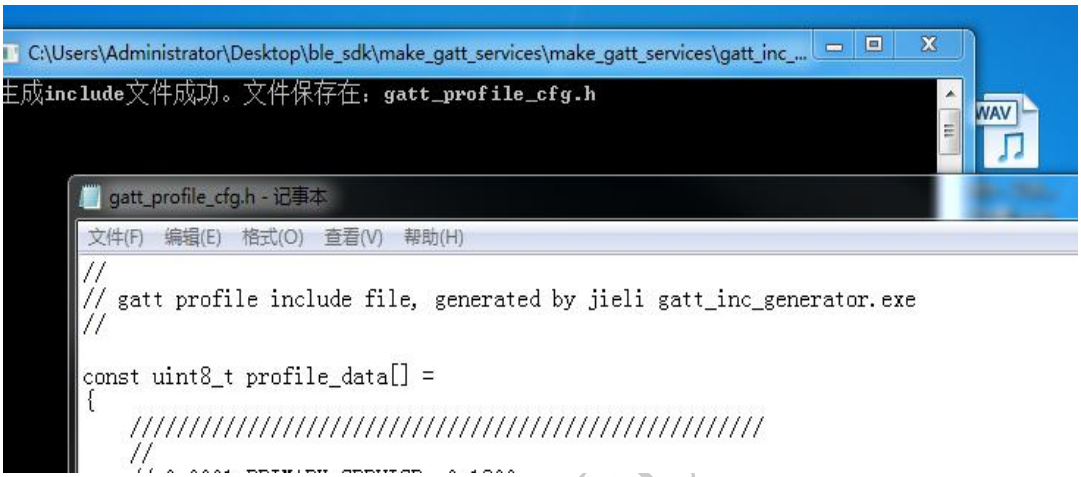
CHARACTERISTIC_USER_DESCRIPTION

---UUID 是 2091，用于增加对 Characteristic 属性的用户描述，数据格式是字符串，属性定义如下：

```
1.  CHARACTERISTIC,  ff03, NOTIFY,
2.  CHARACTERISTIC_USER_DESCRIPTION, READ |  DYNAMIC,
```

1.2.4 Profile 的使用

1、填写完成后点击 gatt_inc_generator.exe 生成 gatt_profile_cfg.h，如下图：



2、将 gatt_profile_cfg.h 内容复制到自己的工程代码中替换原来的 profile_data 和对应的 handle，例如使用 trans_data 例子，将内容复制到 ble_trans_profile.h 中替换。

```
1.  //
2.  // characteristics <--> handles
3.  //
4.  #define ATT_CHARACTERISTIC_2a00_01_VALUE_HANDLE 0x0003
5.  #define ATT_CHARACTERISTIC_ae01_01_VALUE_HANDLE 0x0006
6.  #define ATT_CHARACTERISTIC_ae02_01_VALUE_HANDLE 0x0008
7.  #define ATT_CHARACTERISTIC_ae02_01_CLIENT_CONFIGURATION_HANDLE 0x0009
8.  #define ATT_CHARACTERISTIC_ae03_01_VALUE_HANDLE 0x000b
9.  #define ATT_CHARACTERISTIC_ae04_01_VALUE_HANDLE 0x000d
10. #define ATT_CHARACTERISTIC_ae04_01_CLIENT_CONFIGURATION_HANDLE 0x000e
11. #define ATT_CHARACTERISTIC_ae05_01_VALUE_HANDLE 0x0010
12. #define ATT_CHARACTERISTIC_ae05_01_CLIENT_CONFIGURATION_HANDLE 0x0011
```

```
#define ATT_CHARACTERISTIC_ae10_01_VALUE_HANDLE 0x0013
#define ATT_CHARACTERISTIC_ae3b_01_VALUE_HANDLE 0x0016
#define ATT_CHARACTERISTIC_ae3c_01_VALUE_HANDLE 0x0018
#define ATT_CHARACTERISTIC_ae3c_01_CLIENT_CONFIGURATION_HANDLE 0x0019
```

3、将生成的 handle 对应到程序，执行相应的操作，例如生成

将带有 READ 特性值的属性 Characteristic 对应的 handle 放到 att_read_callback 函数，例如：

ATT_CHARACTERISTIC_2a00_01_VALUE_HANDLE

```
case ATT_CHARACTERISTIC_2a00_01_VALUE_HANDLE: {
        char *gap_name = ble_comm_get_gap_name();
        att_value_len = strlen(gap_name);

        if ((offset >= att_value_len) || (offset + buffer_size) > att_value_len) {
            break;
        }

        if (buffer) {
            memcpy(buffer, &gap_name[offset], buffer_size);
            att_value_len = buffer_size;
            log_info("\n------read gap_name: %s\n", gap_name);
        }
    }
    break;
```

4、将带有 WRITE 特性值的属性 Characteristic 对应的 handle 放到 att_write_callback 函数中处理，例如：

ATT_CHARACTERISTIC_ae01_01_VALUE_HANDLE

```
case ATT_CHARACTERISTIC_ae01_01_VALUE_HANDLE:
        log_info("\n-ae01_rx(%d):", buffer_size);
        put_buf(buffer, buffer_size);
        //收发测试，自动发送收到的数据;for test
        if (ble_comm_att_check_send(connection_handle, buffer_size)) {
            log_info("-loop send1\n");
```

```
            ble_comm_att_send_data(connection_handle, ATT_CHARACTERISTIC_ae02_01_VALUE_HANDLE, buf
fer, buffer_size, ATT_OP_AUTO_READ_CCC);
        }
        break;
```

5、带有 NOTIFY 或者 INDICATE 属性的 handle 可以通过通知的方式发送数据到 CLIENT 端（前提需要 CLIENT 端使能 handle 的 CCC 使能值），发送前先检测缓存是否能填入，再执行发送操作，示例如下：

ATT_CHARACTERISTIC_ae02_01_VALUE_HANDLE

```
if (ble_comm_att_check_send(connection_handle, buffer_size)) {
            log_info("-loop send1\n");
            ble_comm_att_send_data(connection_handle, ATT_CHARACTERISTIC_ae02_01_VALUE_HANDLE, buf
fer, buffer_size, ATT_OP_AUTO_READ_CCC);
 }
```