



make_gatt_services

make_gatt_services 工具使用 用户手册

Rev 1.04

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

版权所有 2021 杰理科技有限公司未经许可，禁止转载




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



1.1 目录结构

 gatt_inc_generator.exe	2018-06-08 13:41	应用程序
 gatt_profile.cfg	2018-01-06 10:21	CFG 文件

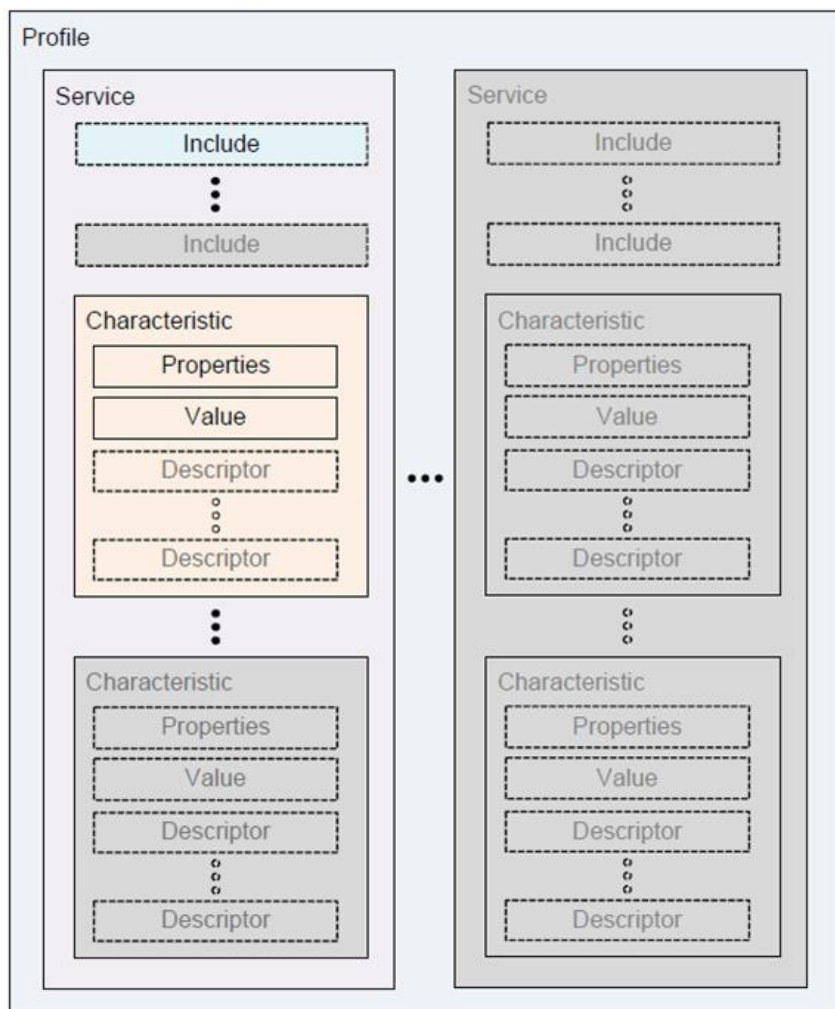
注意两个文件必须在同一级目录下才可使用。

1.2 使用说明

make_gatt_services 主要用于生成 gatt services profile, GATT 指定了数据交互的结构(Structure)

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, 如下图:

GATT Profile Hierarchy





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

1、打开 gatt_profile.cfg 文件如下图, 是 le_server_module.c 例子的 GATT Attribute 生成例子:

```
gatt_profile.cfg x
1 PRIMARY_SERVICE, 0x1800
2 CHARACTERISTIC, 2A00, READ | DYNAMIC,
3
4 PRIMARY_SERVICE, FF00
5 CHARACTERISTIC, FF01, READ | DYNAMIC,
6 CHARACTERISTIC, FF02, READ | WRITE | DYNAMIC,
7 CHARACTERISTIC, FF03, NOTIFY,
8
9
```

图中第 1、2 行代表一个 service。第 4、5、6、7 代表一个 service, 每行的开始是 GATT Declarations, 支持的值有:

PRIMARY_SERVICE, SECONDARY_SERVICE, INCLUDE_SERVICE, CHARACTERISTIC。

其定义可参考蓝牙官网

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, 例如本例子中的 0x1800, 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	0x1800	GCD
Alert Notification Service	org.bluetooth.service.alert_notification	0x1811	GCD
Automation IO	org.bluetooth.service.automation_io	0x1815	GCD
Battery Service	org.bluetooth.service.battery_service	0x180F	GCD
Blood Pressure	org.bluetooth.service.blood_pressure	0x1810	GCD
Body Composition	org.bluetooth.service.body_composition	0x181B	GCD

第2行 CHARACTERISTIC 是包含一个 Characteristic 声明、Characteristic 属性、值、值的描述，例如本例子中的 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: 0x2A00

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

支持的通用属性值有：

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:			None	None
Device Name	Property	Requirement		
Type:	Read	Mandatory		
org.bluetooth.characteristic.gap.device_name	Write	Optional		
Requirement:	WriteWithoutResponse	Excluded		
Mandatory	SignedWrite	Excluded		
	Notify	Excluded		
	Indicate	Excluded		
	WritableAuxiliaries	Excluded		
	Broadcast	Excluded		
	ExtendedProperties			

另外增加私有的属性值有 DYNAMIC, AUTHENTICATION REQUIRED, 分别代表意思如下:

DYNAMIC---数据可变，若有 READ，WRITE，WRITE WITHOUT RESPONSE，会产生的回调函数处理。

AUTHENTICATION REQUIRED---代表属性的读写操作需要配对加密后才能成功操作，要求主机加密。

通用和私有属性可支持并列使用，例如 FF02 有 READ, WRITE, DYNAMIC 属性那么填写

CHARACTERISTIC, FF02, READ | WRITE | DYNAMIC,

注意: CHARACTERISTIC 中有 READ, WRITE, WRITE WITHOUT RESPONSE 属性, 则必须添加 DYNAMIC 属性。

本例子的第4~7行是自定义的 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 属性具体内容可上蓝牙官网查询对应的 UUID。

CHARACTERISTIC 如果有 NOTIFY, INDICATE, profile 结构会默认增加 GATT Descriptors 中的

Client Characteristic Configuration, 需要主机 enable 操作才能执行 NOTIFY 和 IDICATE 动作。

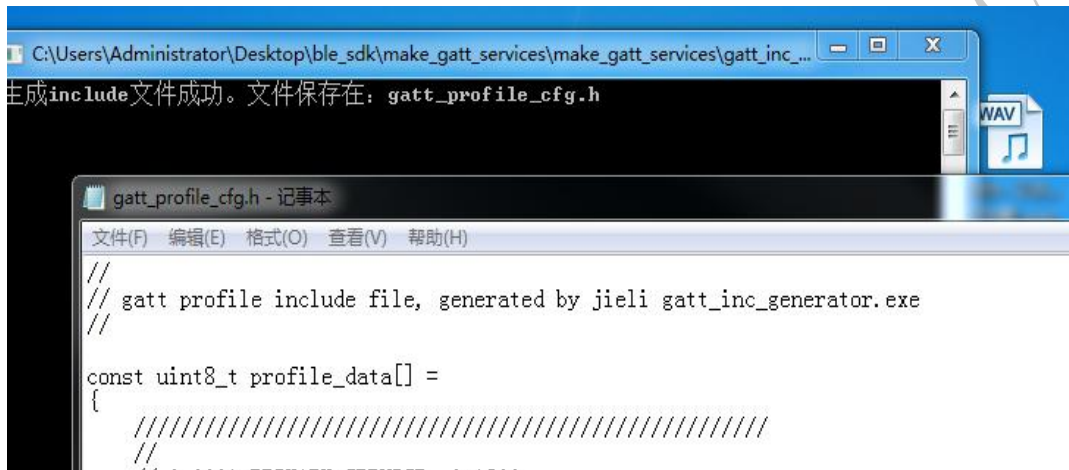
另外还支持 Descriptors 的关键字有 REPORT_REFERENCE；多用于 HID 的 profile 定义，其示例用法和官网的说明如下：



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

Report Reference	org.bluetooth.descr	0x2908	GSS
	iptor.report_refere		
	nce		

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



3、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 le_server_module.c 时，将内容复制到 le_server_module.h 中替换原来的 profile_data 和对应的 handle。

4、将生成的 handle 号对应到程序，例如生成

```
//  
// list mapping between characteristics and handles  
//  
#define ATT_CHARACTERISTIC_GAP_DEVICE_NAME_01_VALUE_HANDLE 0x0003  
#define ATT_CHARACTERISTIC_FF01_01_VALUE_HANDLE 0x0006  
#define ATT_CHARACTERISTIC_FF02_01_VALUE_HANDLE 0x0008  
#define ATT_CHARACTERISTIC_FF03_01_VALUE_HANDLE 0x000a  
#define ATT_CHARACTERISTIC_FF03_01_CLIENT_CONFIGURATION_HANDLE 0x000b
```



带有 read 属性的放到 att_read_callback 函数，例如

ATT_CHARACTERISTIC_GAP_DEVICE_NAME_01_VALUE_HANDLE

```
switch (handle) {
case ATT_CHARACTERISTIC_GAP_DEVICE_NAME_01_VALUE_HANDLE:
|   att_value_len = strlen(gap_device_name);

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

|   if (buffer) {
|       if (att_value_len > buffer_size) {
|           att_value_len = buffer_size;
|       }
|       memcpy(buffer, gap_device_name, att_value_len);
|       log_info("\n-----read gap_name:");
|   }
|   break;
```

将带有 write 属性的放到 att_write_callback 函数中，例如

ATT_CHARACTERISTIC_FF02_01_VALUE_HANDLE

```
case ATT_CHARACTERISTIC_FF02_01_VALUE_HANDLE:
|   if ((offset >= TEST_BUF_SIZE) || (offset + buffer_size) > TEST_BUF_SIZE) {
|       result = -1;
|   } else {
|       log_info("\n-----write test_buf:");
|       memcpy(&test_buf[offset], buffer, buffer_size);
|   }
|   break;
```

将带有 NOTIFY 或者 indication 属性的 handle 通过传参到发送数据的函数中，示例如下：

ATT_CHARACTERISTIC_FF03_01_VALUE_HANDLE

```
//收发测试，自动发送收到的数据;for test
if (app_send_user_data_check(buffer_size)) {
    app_send_user_data(ATT_CHARACTERISTIC_FF03_01_VALUE_HANDLE, buffer, buffer_size, ATT_OP_AUTO_READ_CCC);
}
```




5、若用户想使用自定义的 128bit UUID，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,

6、增加可用关键字 GATT_HANDLE_BEGIN，可指定起始分配的 HANDLE 值；使用示例如下：

!!! 注意：整个 profile 分配生成的 handle 不能出现重复

1. //set handle start 0x0080
2. GATT_HANDLE_BEGIN, 0080
- 3.
4. PRIMARY_SERVICE, ae00
5. CHARACTERISTIC, ae01, WRITE_WITHOUT_RESPONSE | DYNAMIC,
6. CHARACTERISTIC, ae02, NOTIFY,