SEARAN

dotstack GATT database XML file format

Table of Contents

1 Overview

The general format of an XML file used for defining the GATT database is as the following:

```
<gatt xmlns="http://www.searanllc.com/gatt" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation=http://www.searanllc.com/gatt gatt.xsd
xmlns:att="http://www.searanllc.com/att">

<!-- Service 1 -->
<service>
        <characteristics>
                <!-- Characteristic 1 -->
                <characteristic>
                        <value value-length="variable">
                        </value>
                        <extended-properties />
                        <user-description />
                        <client-configuration />
                        <server-configuration />
                        <presentation-format />

                </characteristic>
                .
                .
                .
                <!-- Characteristic N -->
                <characteristic>
                        <value value-length="variable">
                        </value>
                        <extended-properties />
                        <user-description />
                        <client-configuration />
                        <server-configuration />
                        <presentation-format />
                </characteristic>
        </characteristics>
</service>
.
.
.
```

```
<!-- Service N -->
<service>
        …
</service>

</gatt>
```

As you can see the files consists of service definitions with their characteristics. You can have as many services as you like.

2 Service Definition

The <service> tag defines a service. It has two attributes - type and id. The type attribute can be "primary" or "secondary". The "id" attribute can be a 16-bit hexadecimal number (0x1111), a 128-bit hexadecimal number (0x11112222333344445555666677778888), a standard 128-bit UUID representation (123e4567-e89b-12d3-a456-426655440000) or a predefined string constant that corresponds to a Bluetooth SIG assigned service ID:

GATT_SERVICE_UUID_ALERT_NOTIFICATION_SERVICE
GATT_SERVICE_UUID_BATTERY_SERVICE
GATT_SERVICE_UUID_BLOOD_PRESSURE
GATT_SERVICE_UUID_CURRENT_TIME_SERVICE
GATT_SERVICE_UUID_DEVICE_INFORMATION
GATT_SERVICE_UUID_GENERIC_ACCESS
GATT_SERVICE_UUID_GENERIC_ATTRIBUTE
GATT_SERVICE_UUID_HEALTH_THERMOMETER
GATT_SERVICE_UUID_HEART_RATE
GATT_SERVICE_UUID_HUMAN_INTERFACE_DEVICE
GATT_SERVICE_UUID_IMMEDIATE_ALERT
GATT_SERVICE_UUID_LINK_LOSS
GATT_SERVICE_UUID_NEXT_DST_CHANGE_SERVICE
GATT_SERVICE_UUID_PHONE_ALERT_STATUS_SERVICE
GATT_SERVICE_UUID_REFERENCE_TIME_UPDATE_SERVICE
GATT_SERVICE_UUID_SCAN_PARAMETERS
GATT_SERVICE_UUID_TX_POWER

3 Characteristic Definition

The <service> tag can have zero or more <characteristic> tags. Each <characteristic> tag defines one characteristic, its value and optionally characteristic descriptors. The <characteristic> tag has the following attributes:

id - characteristic's ID. It has the same format as the service id except the predefined string constant should be chosen from the following list:

GATT_CHAR_UUID_ALERT_CATEGORY_ID
GATT_CHAR_UUID_ALERT_LEVEL
GATT_CHAR_UUID_ALERT_NOTIFICATION_CONTROL_POINT
GATT_CHAR_UUID_ALERT_STATUS
GATT_CHAR_UUID_APPEARANCE
GATT_CHAR_UUID_BATTERY_LEVEL
GATT_CHAR_UUID_BLOOD_PRESSURE_FEATURE
GATT_CHAR_UUID_BLOOD_PRESSURE_MEASUREMENT
GATT_CHAR_UUID_BODY_SENSOR_LOCATION
GATT_CHAR_UUID_BOOT_KEYBOARD_INPUT_REPORT
GATT_CHAR_UUID_BOOT_KEYBOARD_OUTPUT_REPORT
GATT_CHAR_UUID_BOOT_MOUSE_INPUT_REPORT
GATT_CHAR_UUID_CURRENT_TIME
GATT_CHAR_UUID_DATE_TIME
GATT_CHAR_UUID_DAY_DATE_TIME
GATT_CHAR_UUID_DAY_OF_WEEK"
GATT_CHAR_UUID_DEVICE_NAME"
GATT_CHAR_UUID_DST_OFFSET"
GATT_CHAR_UUID_EXACT_TIME_256"
GATT_CHAR_UUID_FIRMWARE_REVISION_STRING"
GATT_CHAR_UUID_HARDWARE_REVISION_STRING"
GATT_CHAR_UUID_HEART_RATE_CONTROL_POINT"
GATT_CHAR_UUID_HEART_RATE_MEASUREMENT
GATT_CHAR_UUID_HID_CONTROL_POINT
GATT_CHAR_UUID_HID_INFORMATION
GATT_CHAR_UUID_IEEE_11073_20601_REGULATORY_CERTIFICATION_DATA_LIST
GATT_CHAR_UUID_INTERMEDIATE_CUFF_PRESSURE
GATT_CHAR_UUID_INTERMEDIATE_TEMPERATURE
GATT_CHAR_UUID_LOCAL_TIME_INFORMATION
GATT_CHAR_UUID_MANUFACTURER_NAME_STRING
GATT_CHAR_UUID_MEASUREMENT_INTERVAL
GATT_CHAR_UUID_MODEL_NUMBER_STRING

GATT_CHAR_UUID_NEW_ALERT
GATT_CHAR_UUID_PERIPHERAL_PREFERRED_CONNECTION_PARAMETERS
GATT_CHAR_UUID_PERIPHERAL_PRIVACY_FLAG
GATT_CHAR_UUID_PROTOCOL_MODE
GATT_CHAR_UUID_RECONNECTION_ADDRESS
GATT_CHAR_UUID_REFERENCE_TIME_INFORMATION
GATT_CHAR_UUID_REPORT_MAP
GATT_CHAR_UUID_RINGER_CONTROL_POINT
GATT_CHAR_UUID_RINGER_SETTING
GATT_CHAR_UUID_SCAN_INTERVAL_WINDOW
GATT_CHAR_UUID_SCAN_REFRESH
GATT_CHAR_UUID_SERIAL_NUMBER_STRING
GATT_CHAR_UUID_SERVICE_CHANGED
GATT_CHAR_UUID_SOFTWARE_REVISION_STRING
GATT_CHAR_UUID_SUPPORTED_NEW_ALERT_CATEGORY
GATT_CHAR_UUID_SUPPORTED_UNREAD_ALERT_CATEGORY
GATT_CHAR_UUID_SYSTEM_ID
GATT_CHAR_UUID_TEMPERATURE_MEASUREMENT
GATT_CHAR_UUID_TEMPERATURE_TYPE
GATT_CHAR_UUID_TIME_ACCURACY
GATT_CHAR_UUID_TIME_SOURCE
GATT_CHAR_UUID_TIME_UPDATE_CONTROL_POINT
GATT_CHAR_UUID_TIME_UPDATE_STATE
GATT_CHAR_UUID_TIME_WITH_DST
GATT_CHAR_UUID_TIME_ZONE
GATT_CHAR_UUID_TX_POWER_LEVEL
GATT_CHAR_UUID_UNREAD_ALERT_STATUS

broadcast - defines whether broadcast of the Characteristic Value using Server Characteristic Configuration Descriptor is permitted. Possible values: true/false, yes/no, 1|0

read - defines whether read of the Characteristic Value is permitted. Possible values: true/false, yes/no, 1|0.

write-without-response - defines whether writes of the Characteristic Value without response is permitted. Possible values: true/false, yes/no, 1|0.

write - defines whether writes of the Characteristic Value with response is permitted. Possible values: true/false, yes/no, 1|0.

notify - defines whether notifications of a Characteristic Value without acknowledgement are permitted. Possible values: true/false, yes/no, 1|0.

indicate - defines whether indications of a Characteristic Value with acknowledgement are permitted. Possible values: true/false, yes/no, 1|0.

authenticated-signed-write - defines whether signed writes to the Characteristic Value are permitted. Possible values: true/false, yes/no, 1|0.

3.1 Characteristic Value Definition

Each <characteristic> tag has to have exactly one <value> tag which defines the characteristic's value. This tag has the following attributes:

access - defines what opeartions (read or write) are permitted on the value. Possible values: read, write, readwrite. This attribute is optional. If not specified only the read operation is permitted.

authorization - defines whether authorization is required when accessing the characteristic's value. Possible values: read (authorization is required when a central tries to read the value), write (authorization is required when a central tries to write the value), readwrite (authorization is required when a central tries to read or write the value), not-required (authorization is not required). This attribute is optional. If not specified authorization is not required.

authentication - defines whether authentication is required when accessing the characteristic's value. Possible values: read (authentication is required when a central tries to read the value), write (authentication is required when a central tries to write the value), readwrite (authentication is required when a central tries to read or write the value), not-required (authentication is not required). This attribute is optional. If not specified authentication is not required.

min-enc-key-size - minimum encryption size a central should support in order to get access to a characteristic value that requires authentication. This attribute is optional and should only be used if "authentication" is not "not-required". Possible values are from 7 to 16.

storage - defines where the characteristic's value is stored. Possible values: ram, rom, flash. This attribute is optional. If not specified the value is stored in ram.

value-length - defines whether the characteristic's value is of fixed or variable length. Possible values: fixe, variable. This attribute is optional. If not specified the length of the value is fixed.

3.1.1 Allocation storage for Characteristic Value

The space required to store the value of the characteristic is defined with the following children tags of the <value> tag:

<att:uint8 value="<hex or dec number>" /> - reserves one byte for the value. The value should match the following patterns: 0x[\dA-Fa-f]{1,2} or \d{1,3}.

<att:uint16 value="hex or dec number" /> - reserves two bytes for the value. The value should match the following patterns: 0x[\dA-Fa-f]{1,4} or \d{1,5}.

<att:uint32 value="hex or dec number" /> - reserves four bytes for the value. The value should match the following patterns: 0x[\dA-Fa-f]{1,8} or \d{1,10}.

<att:uint64 value="hex or dec number" /> - reserves eight bytes for the value. The value should match the following patterns: 0x[\dA-Fa-f]{1,16} or \d{1,20}.

<att:uint128 value="hex or dec number" /> - reserves sicteen byte for the value. The value should match the following patterns: 0x[\dA-Fa-f]{1,32} or \d{1,39}.

<att:int8 value="hex or sugned dec number" > - reserves one byte for the value. The value should match the following patterns: 0x[\dA-Fa-f]{1,2} or -?\d{1,3}.

<att:int16 value="hex or signed dec number" > - reserves two bytes for the value. The value should match the following patterns: 0x[\dA-Fa-f]{1,4} or -?\d{1,5}.

<att:int32 value="hex or signed dec number" > - reserves four bytes for the value. The value should match the following patterns: 0x[\dA-Fa-f]{1,8} or -?\d{1,10}.

<att:int64 value="hex or signed dec number" > - reserves eight bytes for the value. The value should match the following patterns: 0x[\dA-Fa-f]{1,16} or -?\d{1,19}.

<att:int128 value="hex or signed dec number" > - reserves sixteen bytes for the value. The value should match the following patterns: 0x[\dA-Fa-f]{1,32} or -?\d{1,39}.

<att:bool value="boolean value"> - reserves one byte for the value. The value should match the following patterns: (t|T)(r|R)(u|U)(e|E), (f|F)(a|A)(l|L)(s|S)(e|E), (y|Y)(e|E)(s|S), (n|N)(o|O), (0|1).

<att:float32 value="float value"> - reserves four bytes for the value. The value should match the following pattern: -?\d+(\.\d*)?.

<att:float64 value="float value"> - reserves eight bytes for the value. The value should match the

following pattern: -?\d+(\.\d*)?.

<att:sfloat value="float value"> - reserves two bytes for the value. The value should match the following pattern: -?\d+(\.\d*)?.

<att:float value="float value"> - reserves four bytes for the value. The value should match the following pattern: -?\d+(\.\d*)?.

<att:string value="string value" length="value length"> - reserves len("string value") or "value length" bytes for the characteristic's value. if "length" attribute is not specified the len("string value") bytes are reserved. if "length" attribute is specified reserves the "length" bytes even if "string value" is longer.

<att:uuid16> - reserves two bytes for the value. The value should match the following pattern: 0x[\dA-Fa-f]{1,4}.

<att:uuid32> - reserves four bytes for the value. The value should match the following pattern: 0x[\dA-Fa-f]{1,8}.

<att:uuid128> - reserves sixteen bytes for the value. The value should match the following patterns: 0x[\dA-Fa-f]{1,32}, [\dA-Fa-f]{8}-[\dA-Fa-f]{4}-[\dA-Fa-f]{4}-[\dA-Fa-f]{4}-[\dA-Fa-f]{12}, ^\{[\dA-Fa-f]{8}-[\dA-Fa-f]{4}-[\dA-Fa-f]{4}-[\dA-Fa-f]{4}-[\dA-Fa-f]{12}\}$.

<att:byte-array value="array of bytes" length="array length"> - reserves "array length" bytes for the characteristic's value. The value should match the following patterns: (0x[\dA-Fa-f]{1,4}\s)*(0x[\dA-Fa-f]{1,4}){0,1} or (\d{1,3}\s)*(\d{1,3}){0,1}. I.e., the array of bytes can specified as the following:

Hexadecimal notation - 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F 0x10

Decimal notation - 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

There can be any number in any order of the above tags. They do not have any special meaning to the dotstack. They just simply reserve space for storing the characteristic's value. What is stored in the reserved space is profile specific as each profile defines its own set of characteristics and their values.

3.2 Characteristic Descriptors Definition

Each <characteristic> can have zero or more optional descriptors. The available descriptor are:

<extended-properties> - defines additional Characteristic Properties.

<user-description> - defines a UTF-8 string of variable size that is a user textual description of the Characteristic Value.

<client-configuration> - defines how the characteristic may be configured by a specific client.

<server-configuration> - defines how the characteristic may be configured for the server.

<presentation-format> - defines the format of the Characteristic Value.

<custom-descriptor> - custom descriptor that can be defined by an application for its own use.

There can only at maximum one instance of each descriptor.

3.2.1 Extended Properties Descriptor

This descriptor has the following attributes:

reliable-write - defines whether reliable writes of the Characteristic Value are permitted. Possible values: true|false, yes|no, 0|1.

writable-auxiliaries - defines whether permits to the User Description characteristic descriptor are permitted. Possible values: true|false, yes|no, 0|1.

3.2.2 User Description Descriptor

This descriptor has the following attributes:

value - Characteristic User Description UTF-8 String.

3.2.3 Client Configuration Descriptor

This descriptor has the following attributes:

notification - defines whether the Characteristic Value shall be notified. Possible values: true|false, yes|no, 0|1.

indication - defines whether the Characteristic Value shall be indicated. Possible values: true|false, yes|no, 0|1.

3.2.4 Server Configuration Descriptor

This descriptor has the following attributes:

broadcast - defines whether the Characteristic Value shall be broadcast when the server is in the broadcast procedure if advertising data resources are available. This value can only be set if the characteristic's property has the broadcast bit set. Possible values: true|false, yes|no, 0|1.

3.2.5 Presentation Format Descriptor

This descriptor has the following attributes:

format - format of the value of this characteristic. The possible values are:

Boolean
2bit
nibble
uint8
uint12
uint16
uint24
uint32
uint48
uint64
uint128
sint8
sint12
sint16
sint24
sint32
sint48
sint64
sint128

float32
float64
SFLOAT
FLOAT
duint16
utf8s
utf16s
struct

exponent - exponent field to determine how the value of this

characteristic is further formatted. This is a one byte hexadecimal or decimal number.

unit - the unit of this characteristic.

name-space - this attribute is used to identify the organization that is responsible for defining the enumerations for the "description" attribute. The only allowed values is 1 (Bluetooth SIG Assigned Numbers)

description - this is an enumerated value from the organization identified by the "name-space" attribute.

3.2.6 Custom Descriptor

This descriptor has the following attributes:

read - defines whether reads of the descriptor value are permitted.

write - defines whether writes to the descriptor value are permitted.

authorization - defines whether authorization is required when accessing the descriptor's value. Possible values: read (authorization is required when a central tries to read the value), write (authorization is required when a central tries to write the value), readwrite (authorization is required when a central tries to read or write the value), not-required (authorization is not required). This attribute is optional. If not specified authorization is not required.

authentication - defines whether authentication is required when accessing the descriptor's value. Possible values: read (authentication is required when a central tries to read the value), write (authentication is required when a central tries to write the value), readwrite (authentication is required when a central tries to read or write the value), not-required (authentication is not required). This attribute is optional. If not specified authentication is not required.

min-enc-key-size - minimum encryption size a client should support in order to get access to the descriptor's value that requires authentication. This attribute is optional and should only be used if "authentication" is not "not-required". Possible values are from 7 to 16.

4 Additional Information

For farther information on GATT database refer to Core Version 4.1 (https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=282159), Volume 3, Part G: Generic Attribute Profile.