

# GATT Schema

---

## 1 Directory Contents

The following files can be found in the directory:

Filename	Type of File
assigned_numbers.xsd	Schema
attribute.xsd	Schema
attribute_field.xsd	Schema
attribute_uri.xsd	Schema
characteristic.xsd	Schema
characteristic_field.xsd	Schema
characteristic_uri.xsd	Schema
descriptions.xsd	Schema
descriptor.xsd	Schema
descriptor_field.xsd	Schema
descriptor_uri.xsd	Schema
formats.xsd	Schema
profile.xsd	Schema
profile_configuration.xsd	Schema
profile_role.xsd	Schema
profile_uri.xsd	Schema
service.xsd	Schema
service_uri.xsd	Schema
unit_uri.xsd	Schema
uuid.xsd	Schema
service_characteristic.xsd	Schema
service_configuration.xsd	Schema
gatt.sqlite	SQLite database
org.bluetooth.assigned_numbers	Assigned Numbers
org.bluetooth.attribute.gatt.characteristic	Attribute
org.bluetooth.attribute.gatt.include	Attribute
org.bluetooth.attribute.gatt.primary_service	Attribute
org.bluetooth.attribute.gatt.secondary_service	Attribute
org.bluetooth.characteristic.alert_level	Characteristic
org.bluetooth.characteristic.battery_level	Characteristic
org.bluetooth.characteristic.battery_level_state	Characteristic
org.bluetooth.characteristic.battery_state	Characteristic
org.bluetooth.characteristic.blood_pressure_measurement	Characteristic
org.bluetooth.characteristic.boolean	Characteristic
org.bluetooth.characteristic.current_time	Characteristic
org.bluetooth.characteristic.date_time	Characteristic
org.bluetooth.characteristic.day_of_week	Characteristic
org.bluetooth.characteristic.daylight_saving_time	Characteristic

org.bluetooth.characteristic.elevation	Characteristic
org.bluetooth.characteristic.exact_time_100	Characteristic
org.bluetooth.characteristic.exact_time_256	Characteristic
org.bluetooth.characteristic.firmware_revision_string	Characteristic
org.bluetooth.characteristic.gap.appearance	Characteristic
org.bluetooth.characteristic.gap.device_name	Characteristic
org.bluetooth.characteristic.gap.peripheral_preferred_connection_parameters	Characteristic
org.bluetooth.characteristic.gap.peripheral_privacy_flag	Characteristic
org.bluetooth.characteristic.gap.reconnection_address	Characteristic
org.bluetooth.characteristic.gatt.service_changed	Characteristic
org.bluetooth.characteristic.hardware_revision_string	Characteristic
org.bluetooth.characteristic.heart_rate_control_point	Characteristic
org.bluetooth.characteristic.heart_rate_measurement	Characteristic
org.bluetooth.characteristic.heart_rate_sensor_location	Characteristic
org.bluetooth.characteristic.hid_version	Characteristic
org.bluetooth.characteristic.ieee_11073-20601_regulatory_certification_data_list	Characteristic
org.bluetooth.characteristic.intermediate_blood_pressure	Characteristic
org.bluetooth.characteristic.intermediate_temperature	Characteristic
org.bluetooth.characteristic.latitude	Characteristic
org.bluetooth.characteristic.local_time_information	Characteristic
org.bluetooth.characteristic.longitude	Characteristic
org.bluetooth.characteristic.manufacturer_name_string	Characteristic
org.bluetooth.characteristic.measurement_interval	Characteristic
org.bluetooth.characteristic.model_number_string	Characteristic
org.bluetooth.characteristic.position_2d	Characteristic
org.bluetooth.characteristic.position_3d	Characteristic
org.bluetooth.characteristic.product_id	Characteristic
org.bluetooth.characteristic.reference_time_information	Characteristic
org.bluetooth.characteristic.secondary_time_zone	Characteristic
org.bluetooth.characteristic.serial_number_string	Characteristic
org.bluetooth.characteristic.software_revision_string	Characteristic
org.bluetooth.characteristic.system_id	Characteristic
org.bluetooth.characteristic.temperature_celsius	Characteristic
org.bluetooth.characteristic.temperature_fahrenheit	Characteristic
org.bluetooth.characteristic.temperature_measurement	Characteristic
org.bluetooth.characteristic.temperature_type	Characteristic
org.bluetooth.characteristic.time	Characteristic
org.bluetooth.characteristic.time_accuracy	Characteristic
org.bluetooth.characteristic.time_broadcast	Characteristic
org.bluetooth.characteristic.time_source	Characteristic
org.bluetooth.characteristic.time_update_control_point	Characteristic
org.bluetooth.characteristic.time_update_state	Characteristic
org.bluetooth.characteristic.time_with_dst	Characteristic
org.bluetooth.characteristic.time_zone	Characteristic
org.bluetooth.characteristic.tx_power_level	Characteristic
org.bluetooth.characteristic.vendor_id	Characteristic

org.bluetooth.characteristic.vendor_id_source	Characteristic
org.bluetooth.descriptor.gatt.characteristic_aggregate_format	Descriptor
org.bluetooth.descriptor.gatt.characteristic_extended_properties	Descriptor
org.bluetooth.descriptor.gatt.characteristic_presentation_format	Descriptor
org.bluetooth.descriptor.gatt.characteristic_user_description	Descriptor
org.bluetooth.descriptor.gatt.client_characteristic_configuration	Descriptor
org.bluetooth.descriptor.gatt.server_characteristic_configuration	Descriptor
org.bluetooth.descriptor.valid_range	Descriptor
org.bluetooth.service.battery	Service
org.bluetooth.service.current_time	Service
org.bluetooth.service.device_information	Service
org.bluetooth.service.dst_change	Service
org.bluetooth.service.generic_access	Service
org.bluetooth.service.generic_attribute	Service
org.bluetooth.service.health_thermometer	Service
org.bluetooth.service.immediate_alert	Service
org.bluetooth.service.link_loss	Service
org.bluetooth.service.network_availability	Service
org.bluetooth.service.reference_time_update	Service
org.bluetooth.service.tx_power	Service
org.bluetooth.service.watchdog	Service
org.bluetooth.profile.battery	Profile
org.bluetooth.profile.find_me	Profile
org.bluetooth.profile.health_thermometer	Profile
org.bluetooth.profile.proximity	Profile
org.bluetooth.profile.time_update	Profile
loadxml.py	XML to Python
mkdatabase.py	Python to SQLite
sanitycheck.py	Python Sanity Check
update	Shell Script
validate	Shell Script

## 2 Schema Files

assigned_numbers.xsd	Purpose
attribute.xsd	Attribute
attribute_field.xsd	Attribute Field
attribute_uri.xsd	Defined Attribute URIs
characteristic.xsd	Characteristic
characteristic_field.xsd	Characteristic Field
characteristic_uri.xsd	Defined Characteristic URIs
descriptions.xsd	Descriptions
descriptor.xsd	Descriptor
descriptor_field.xsd	Descriptor Field
descriptor_uri.xsd	Descriptor URIs
formats.xsd	Data Formats
service.xsd	Service
service_characteristic.xsd	Service Characteristics

service_configuration.xsd	Service Configurations
service_uri.xsd	Service URIs
profile.xsd	Profile
profile_configuration.xsd	Profile Configurations
profile_role.xsd	Profile Roles
profile_uri.xsd	Profile URIs
unit_uri.xsd	Unit URIs
uuid.xsd	16-bit and 128-bit UUIDs

## 2.1 URIs

A Universal Resource Indicators are used to uniquely identify an individual component within this schema. All Bluetooth SIG URIs start with *org.bluetooth*. If an external organization is wishing to define additional URIs, then these should start with a URI that is owned by that company. For example, if a company has example.com as a registered domain name, then the URI would start with *com.example*.

The next part of the URI identifies the type of this resource. The defined types are:

- attribute
- characteristic
- descriptor
- service
- profile
- unit

In some special cases, where attributes, characteristics or descriptors are defined in a specification that is not standalone. In these cases, the next part of the URI should identify the specification where these are defined. For example, the appearance characteristic is defined in the GAP core specification, and therefore the URI is: *org.bluetooth.characteristic.gap.appearance*. At the moment, only GATT and GAP are allowed specification identifiers.

The last part of the URI is the name of the resource. These should replace any spaces in the specification name with an underscore '\_'. For example, the battery level characteristic defined by the Bluetooth SIG would have the URI: *org.bluetooth.characteristic.battery\_level*. The last part of the name should never include the type again, even if in the specifications they are labeled with that type. For example, the Health Thermometer Service is referenced as «Health Thermometer Service» in the specification, but when used as a URI would be: *org.bluetooth.service.health\_thermometer*.

In addition to attributes, characteristics, descriptors, services and profiles, the schema also defines URIs for units. The format of a unit URI is similar to the above URIs but with an additional part. The unit has both the type of unit and unit. For example, the standard length unit is the metre would have a URI of 'org.bluetooth.unit.length.metre'; additional length units for feet would have the URI of 'org.bluetooth.unit.length.foot'. The units are always singular.

The URIs are utf-8 encoded, and therefore the URI for the unit an ångström is: 'org.bluetooth.unit.length.ångström'. It is however recommended that English is used as the language for all schema based files.

The schema contains a number of [type]\_uri.xsd files that define the valid uris for a given type. These are defined in separate files so that in the individual schema files a characteristic cannot use an attribute or descriptor URI. Therefore the following uri files exist.

- attribute\_uri.xsd
- characteristic\_uri.xsd
- descriptor\_uri.xsd
- service\_uri.xsd
- profile\_uri.xsd
- unit\_uri.xsd

## 2.2 Formats

The schema defines data formats, similar to those defined in GATT section 3.3.3.5.2. There are additional formats defined in addition to those in that section, for example a 40-bit unsigned integer.

The format.xsd schema file contains three separate formats one each for: characteristics, attributes, and descriptors. These are differentiated as defined below:

- characteristic formats define the additional 'uint40' format
- attribute formats define a 'uuid' as a format – this can either be a 16-bit UUID or a 128-bit UUID
- attribute formats do not define a 'struct' format
- descriptor formats do not define a 'struct' format
- descriptor formats define a 'characteristic' format – where the format of this field is determined by the format of the characteristic value that this descriptor is contained

## 2.3 Characteristics

The characteristic.xsd file defines a characteristic, including the characteristic\_uri.xsd file, and the characteristic\_field.xsd file.

A characteristic file must define its URI in the file, and a human readable name. It can optionally have some additional notes – informative text associated with this characteristic. It denotes if it has been adopted by the Bluetooth SIG. Finally there are one or more fields.

The URI in this characteristic file must match the filename. The human readable name must match the same name in the assigned numbers document. These are checks cannot be performed by any xml validation tool, and therefore are performed by an additional sanitycheck.py program.

### 2.3.1 Characteristic Field

Each field of a characteristic is defined using a <field> element. This is a complex schema and therefore is contained in its own xsd file. It could be possible to combine this file with the characteristic.xsd file, but the easy of readability of these files was considered more important than just reducing the number of xsd files.

A field always has a fieldname, and optionally some informative notes. The field then has a choice of either having a <format> element and other associated information to define the format of this field, or it can have <reference> element that uses the fields in the referenced characteristic in this characteristic. This referencing allows two characteristics that have very similar formats, for example the «Temperature Measurement» and «Intermediate Temperature» characteristics have the same format; the fields are defined in the «Temperature Measurement» characteristic, and the «Intermediate Temperature» just references the fields.

The format of a field defines how many octets this field occupies, and the format of the data within this field. The rest of the elements then further define the meaning of this field's value. This can include either a decimal or binary exponent, a unit, a minimum and maximum value, a description value, one or more enumerated values, or a bit-field.

Enumerated values are key : value pairs that define a string based upon a given key. There are no true and false values, and therefore the key of '0' is defined as false, and the key of '1' is defined as true. A enumeration key : value pair may also define additional field elements such that optional fields can be included in a characteristic depending on the value of an enumerated value.

Bit-fields are similar to enumerations except that each set of bits within a bit-field, defined by the bit index and the size of this set of bits. The bits themselves can be further defined as taking the "value" of this field, being reserved, or being an enumeration. Enumerations can also reference additional bit-fields and fields.

## 2.4 Attributes & Descriptors

Attributes and descriptors are very similar to characteristics except that they include different attribute field, descriptor field, and URI xsd files. This distinction allows the slightly different semantics of an attribute or a descriptor to be enforced.

## 2.5 Services

A service has the same URI and human readable names that are required by characteristics, attributes and descriptors. The same requirements for sanity checking are performed. Services can also have additional informative notes associated with them. Services also have the same adopted element. After the basic information about a service, one or more characteristics can be defined. Following this, one or more configurations can be defined.

Service characteristics define which characteristics are used by the service, along with their requirements, descriptor requirements, characteristic and descriptor properties, and characteristic security requirements.

Service configurations allow one or more characteristics to be given additional semantics around requirements. For example, if one characteristic exists, another characteristic must also be used.

## 2.6 Profiles

At the moment, the profile schema is still a work in progress, as the representation of characteristics and services has taken a higher priority.

## 2.7 Descriptions

The GATT specification defines a number of “descriptions” that can be associated with a characteristic. The descriptions.xsd schema file enumerates the allowable descriptions that can then be used by characteristics.

## 2.8 Assigned Numbers

The assignednumbers.xsd file defines the schema for the assigned numbers document. This includes a mapping for each of the service, unit, attribute, descriptor, and characteristic URIs to both a defined uuid, a defined name, and whether this assigned number has been adopted by the Bluetooth SIG. For units, it also defines the reference, symbol, and a way to express this value using SI units. These three are all meant to be human readable elements.

# 3 Work Flow

There are two shell scripts exposed: validate and update.

The validate shell script exposed with the schema to validate the schema and associated files. This validate script performs an xmllint on each of the types of file with the associated schema. This implicitly ensures that the URI types, as defined by the filename, are using the correct schema.

Upon a successful xmllint of each of these files, a sanitycheck.py is performed. This loads in all the files into memory and then performs a sanity check. This validates that the assigned numbers match the individual files. It also validates that each assigned number has an associated individual file.

The update shell script first performs a validation of the information using the above validate shell script. It then runs the mkdatabase.py that takes all the information in the files and represents this in an SQLite database. This generates the gatt.sqlite database file. This is primarily done to aid the generation of webpages by using the SQLite database rather than the raw xml files. Finally, a copy of these schema files is made on a server that can then deliver web pages based on this database.

## 3.1 Tools Required

The following tools are used as part of this system:

- xmllint – this is a program that comes as part of xml-lib2 and is distributed free with all good operating systems.

- python – this is a programming language that is used to process the xml files, load them into memory, convert them into an SQLite database, and sanity-check the files.
- SQLite – this is an open source database that is highly optimized for simple databases. The key difference is that the database is linked into the program that needs to access the data rather than as a standalone database. It is therefore very quick for small datasets.
- rsync – a file coping program that synchronizes files between a source and destination machine. It performs comparisons of the files over the network before copying any files, therefore only copying files that have changed.

## 4 Example Files

### 4.1 GATT Characteristic Attribute

```

1 : <?xml version="1.0" encoding="UTF-8"?>
2 :
3 : <attribute xsi:noNamespaceSchemaLocation="attribute.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4 :     <uri>org.bluetooth.attribute.gatt.characteristic</uri>
5 :     <name>Characteristic</name>
6 :     <adopted>true</adopted>
7 :     <field>
8 :         <fieldname>Characteristic Properties</fieldname>
9 :         <format>uint8</format>
10 :         <bitfield>
11 :             <bit index="0">
12 :                 <enumeration key="1" value="Broadcast"/>
13 :             </bit>
14 :             <bit index="1">
15 :                 <enumeration key="1" value="Read"/>
16 :             </bit>
17 :             <bit index="2">
18 :                 <enumeration key="1" value="Write Without Response"/>
19 :             </bit>
20 :             <bit index="3">
21 :                 <enumeration key="1" value="Write"/>
22 :             </bit>
23 :             <bit index="4">
24 :                 <enumeration key="1" value="Notify"/>
25 :             </bit>
26 :             <bit index="5">
27 :                 <enumeration key="1" value="Indicate"/>
28 :             </bit>
29 :             <bit index="6">
30 :                 <enumeration key="1" value="Authenticated Signed Writes"/>
31 :             </bit>
32 :             <bit index="7">
33 :                 <enumeration key="1" value="Extended Properties"/>
34 :             </bit>
35 :         </bitfield>
36 :     </field>
37 :     <field>
38 :         <fieldname>Characteristic Value Handle</fieldname>
39 :         <format>uint16</format>
40 :     </field>
41 :     <field>
42 :         <fieldname>Characteristic UUID</fieldname>
43 :         <format>uuid</format>
44 :     </field>
45 : </attribute>

```



Line 1 defines that this is an xml formatted file, based on version 1.0 of the xml specifications, using a UTF-8 character encoding.

Line 3 starts the definition of this attributes, using the defined schemas.

Line 4 defines the URI for this attribute. This is the GATT Characteristic attribute, as defined by GATT Section 3.3.1.

Line 5 defines that the name is «Characteristic».

Line 6 says that this attribute has been adopted.

Line 7 to 36, 37 to 40 and 41 to 44 define three fields.

Line 8 defines that the first field of this characteristic is called “Characteristic Properties”.

Line 9 defines that the Characteristic Properties is an unsigned 8 bit value.

Line 10 defines that the value of this field is a bitfield.

Line 11 and 12 defines that bit 0 that has the value 1 has the label “Broadcast”.

Line 14 and 15 defines that bit 1 that has the value 1 has the label “Read”.

Line 17 and 18 defines that bit 2 that has the value 1 has the label “Write Without Response”.

Line 20 and 21 defines that bit 3 that has the value 1 has the label “Write”.

Line 23 and 24 defines that bit 4 that has the value 1 has the label “Notify”.

Line 26 and 27 defines that bit 5 that has the value 1 has the label “Indicate”.

Line 29 and 30 defines that bit 6 that has the value 1 has the label “Authenticated Signed Writes”.

Line 32 and 33 defines that bit 7 that has the value 1 has the label “Extended Properties”.

Line 38 defines that the second field is called “Characteristic Value Handle”.

Line 39 defines that the Characteristic Value Handle is an unsigned 16-bit value.

Line 42 defines that the last field is called “Characteristic UUID”.

Line 43 defines that the format of this field is a uuid; either a 16-bit uuid or a 128-bit uuid.

## 4.2 Tx Power Level Characteristic

```
1 : <?xml version="1.0" encoding="UTF-8"?>
2 :
3 : <characteristic xsi:noNamespaceSchemaLocation="characteristic.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4 :     <uri>org.bluetooth.characteristic.tx_power_level</uri>
5 :     <name>Tx Power Level</name>
```

```

6 :      <adopted>false</adopted>
7 :      <field>
8 :          <fieldname>Tx Power</fieldname>
9 :          <format>sint8</format>
10 :          <unit>org.bluetooth.unit.logarithmic_radio_quantity.decibel</unit>
11 :      </field>
12 : </characteristic>

```

This characteristic is very similar to the example above, except that line 10 defines that the Tx Power field is represented using the logarithmic radio quantity of decibel unit.

### 4.3 Latitude Characteristic

```

1 : <?xml version="1.0" encoding="UTF-8"?>
2 :
3 : <characteristic xsi:noNamespaceSchemaLocation="characteristic.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4 :     <uri>org.bluetooth.characteristic.latitude</uri>
5 :     <name>Latitude</name>
6 :     <adopted>false</adopted>
7 :     <field>
8 :         <fieldname>Latitude</fieldname>
9 :         <notes>Latitude in degrees north</notes>
10 :         <format>sint32</format>
11 :         <decimalexponent>-7</decimalexponent>
12 :         <unit>org.bluetooth.unit.plane_angle.degree</unit>
13 :         <minimum>-899999999</minimum>
14 :         <maximum>900000000</maximum>
15 :         <description>Latitude</description>
16 :     </field>
17 : </characteristic>

```

This characteristic shows the use of notes (line 9), decimal exponent (line 11), minimum and maximum value ranges (line 13 and 14) and description elements (line 15).

This characteristic therefore has a single field named latitude with notes stating that this is “Latitude in degrees north”. It is encoded as a signed 32 bit integer with a decimal exponent of  $10^{-7}$ . The value uses the plane angle unit of degrees. Valid values range from -89.9999999 to 90.0000000 after the exponent has been applied. And the description field says that this is a Latitude.

### 4.4 Position 2D Characteristic

```

1 : <?xml version="1.0" encoding="UTF-8"?>
2 :
3 : <characteristic xsi:noNamespaceSchemaLocation="characteristic.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4 :     <uri>org.bluetooth.characteristic.position_2d</uri>
5 :     <name>Position 2D</name>
6 :     <adopted>false</adopted>
7 :     <field>
8 :         <fieldname>Latitude</fieldname>
9 :         <reference>
10 :             <type>org.bluetooth.characteristic.latitude</type>
11 :         </reference>
12 :     </field>
13 :     <field>
14 :         <fieldname>Longitude</fieldname>
15 :         <reference>
16 :             <type>org.bluetooth.characteristic.longitude</type>
17 :         </reference>
18 :     </field>

```

```
19 : </characteristic>
```

This characteristic shows how two characteristic formats can be combined into a single characteristic. In this case, a two-dimensional position on the earth can be encoded using two fields, one that is called Latitude using the format of the latitude characteristic (shown above), and another that is called Longitude using the format of the longitude characteristic.

## 4.5 Position 3D

```
1 : <?xml version="1.0" encoding="UTF-8"?>
2 :
3 : <characteristic xsi:noNamespaceSchemaLocation="characteristic.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4 :   <uri>org.bluetooth.characteristic.position_3d</uri>
5 :   <name>Position 3D</name>
6 :   <adopted>false</adopted>
7 :   <field>
8 :     <fieldname>Position</fieldname>
9 :     <reference>
10 :       <type>org.bluetooth.characteristic.position_2d</type>
11 :     </reference>
12 :   </field>
13 :   <field>
14 :     <fieldname>Elevation</fieldname>
15 :     <reference>
16 :       <type>org.bluetooth.characteristic.elevation</type>
17 :     </reference>
18 :   </field>
19 : </characteristic>
```

This characteristic shows that a 3D position can then be created that is a combination of the Position field, referencing the 2d-position characteristic shown above, as an Element field that references the elevation characteristic. Using this method, complex characteristics can be built out of a combination of very simple characteristics.

## 4.6 Temperature Measurement Characteristic

```
1 : <?xml version="1.0" encoding="UTF-8"?>
2 :
3 : <characteristic xsi:noNamespaceSchemaLocation="characteristic.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4 :   <uri>org.bluetooth.characteristic.temperature_measurement</uri>
5 :   <name>Temperature Measurement</name>
6 :   <adopted>false</adopted>
7 :   <field>
8 :     <fieldname>Flags</fieldname>
9 :     <format>uint8</format>
10 :    <bitfield>
11 :      <bit index="0" size="1">
12 :        <enumeration key="0" value="Celsius">
13 :          <field>
14 :            <fieldname>Temperature Measurement Value (Celsius)</fieldname>
15 :            <format>FLOAT</format>
16 :            <unit>org.bluetooth.unit.thermodynamic_temperature.degree_celsius</unit>
17 :            <enumeration key="8388606" value="+ Infinity"/>
18 :            <enumeration key="8388607" value="Not A Number"/>
19 :            <enumeration key="8388608" value="Not At This Resolution"/>
20 :            <enumeration key="8388609" value="Reserved For Future Use"/>
21 :            <enumeration key="8388610" value="- Infinity"/>
22 :          </field>
23 :        </enumeration>
```

```

24 :      <enumeration key="1" value="Fahrenheit">
25 :          <field>
26 :              <fieldname>Temperature Measurement Value (Fahrenheit)</fieldname>
27 :              <format>FLOAT</format>
28 :              <unit>org.bluetooth.unit.thermodynamic_temperature.degree_fahrenheit</unit>
29 :              <enumeration key="8388606" value="+ Infinity"/>
30 :              <enumeration key="8388607" value="Not A Number"/>
31 :              <enumeration key="8388608" value="Not At This Resolution"/>
32 :              <enumeration key="8388609" value="Reserved For Future Use"/>
33 :              <enumeration key="8388610" value="- Infinity"/>
34 :          </field>
35 :      </enumeration>
36 :  </bit>
37 :  <bit index="1" size="1">
38 :      <enumeration key="0" value="No Time Stamp"/>
39 :      <enumeration key="1" value="Time Stamp present">
40 :          <field>
41 :              <fieldname>Time Stamp</fieldname>
42 :              <reference>
43 :                  <type>org.bluetooth.characteristic.date_time</type>
44 :              </reference>
45 :          </field>
46 :      </enumeration>
47 :  </bit>
48 :  <bit index="2" size="1">
49 :      <enumeration key="0" value="No Temperature Type"/>
50 :      <enumeration key="1" value="Temperature Type present">
51 :          <field>
52 :              <fieldname>Temperature Type</fieldname>
53 :              <reference>
54 :                  <type>org.bluetooth.characteristic.temperature_type</type>
55 :              </reference>
56 :          </field>
57 :      </enumeration>
58 :  </bit>
59 :  </bitfield>
60 : </field>
61 : </characteristic>

```

Probably the most complex characteristic today is the «Temperature Measurement» characteristic from the medical working group. The characteristic itself has just one field (line 7) “Flags”. The Flags value is an unsigned 8-bit bitfield that defines three bits (line 11, 37, 48). The first bit 0 is one bit in size and defines two enumerated values: 0 = Celsius, 1 = Fahrenheit. For each of these enumerations, an additional field is defined that is a FLOAT in the correct unit. This bit 0 is therefore used to determine if the next 32 bits is a floating point value in Celsius or Fahrenheit.

The next bit (line 37 to 47) defines that the 2<sup>nd</sup> bit in the Flags field defines another single bit value with two enumerations. The first enumeration (line 38) just equates the value of this bit = 0 to the label “No Time Stamp”. The second enumeration (line 39) equates the value of this bit = 1 to the label “Time Stamp present” and then defines an additional field for the time stamp, reusing the date time characteristic.

The last defined bit (line 48 to 58) defines that the 3<sup>rd</sup> bit in the Flags field defines another single bit value with another two enumerations. This is similar to above except it references the temperature type characteristic.

## 4.7 Immediate Alert Service

```
1 : <?xml version="1.0" encoding="UTF-8"?>
2 :
3 : <service xsi:noNamespaceSchemaLocation="service.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance">
4 :   <uri>org.bluetooth.service.immediate_alert</uri>
5 :   <name>Immediate Alert Service</name>
6 :   <adopted>false</adopted>
7 :   <characteristic>
8 :     <type>org.bluetooth.characteristic.alert_level</type>
9 :     <requirements>mandatory</requirements>
10 :    <property>write</property>
11 :  </characteristic>
12 : </service>
```

The Immediate Alert Service defines just a single characteristic (line 7), alert level (line 8), that is mandatory (line 9) with write properties (line 10).

## 4.8 Link Loss Service

```
1 : <?xml version="1.0" encoding="UTF-8"?>
2 :
3 : <service xsi:noNamespaceSchemaLocation="service.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance">
4 :   <uri>org.bluetooth.service.link_loss</uri>
5 :   <name>Link Loss Service</name>
6 :   <adopted>false</adopted>
7 :   <characteristic>
8 :     <type>org.bluetooth.characteristic.alert_level</type>
9 :     <requirements>mandatory</requirements>
10 :    <property>read</property>
11 :    <property>write</property>
12 :  </characteristic>
13 : </service>
```

The Link Loss Service is similar to above; it defines the same characteristic (line 7 to 8) but this time the characteristic has two properties: read and write (line 10 & 11).

## 4.9 Device Information Service

```
1 : <?xml version="1.0" encoding="UTF-8"?>
2 :
3 : <service xsi:noNamespaceSchemaLocation="service.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance">
4 :   <uri>org.bluetooth.service.device_information</uri>
5 :   <name>Device Information Service</name>
6 :   <adopted>false</adopted>
7 :   <characteristic>
8 :     <type>org.bluetooth.characteristic.manufacturer_name_string</type>
9 :     <requirements>conditional</requirements>
10 :    <property>read</property>
11 :  </characteristic>
12 :  <characteristic>
13 :    <type>org.bluetooth.characteristic.model_number_string</type>
14 :    <requirements>conditional</requirements>
15 :    <property>read</property>
16 :  </characteristic>
17 :  <characteristic>
18 :    <type>org.bluetooth.characteristic.serial_number_string</type>
19 :    <requirements>conditional</requirements>
20 :    <property>read</property>
21 :  </characteristic>
```

```

22 :     <characteristic>
23 :         <type>org.bluetooth.characteristic.hardware_revision_string</type>
24 :         <requirements>conditional</requirements>
25 :         <property>read</property>
26 :     </characteristic>
27 :     <characteristic>
28 :         <type>org.bluetooth.characteristic.firmware_revision_string</type>
29 :         <requirements>conditional</requirements>
30 :         <property>read</property>
31 :     </characteristic>
32 :     <characteristic>
33 :         <type>org.bluetooth.characteristic.software_revision_string</type>
34 :         <requirements>conditional</requirements>
35 :         <property>read</property>
36 :     </characteristic>
37 :     <characteristic>
38 :         <type>org.bluetooth.characteristic.system_id</type>
39 :         <requirements>conditional</requirements>
40 :         <property>read</property>
41 :     </characteristic>
42 :     <characteristic>
43 :         <type>org.bluetooth.characteristic.ieee_11073-20601_regulatory_certification_data_list</type>
44 :         <requirements>conditional</requirements>
45 :         <property>read</property>
46 :     </characteristic>
47 :
48 :     <configuration type="must_support_one_of">
49 :         <type>org.bluetooth.characteristic.manufacturer_name_string</type>
50 :         <type>org.bluetooth.characteristic.model_number_string</type>
51 :         <type>org.bluetooth.characteristic.serial_number_string</type>
52 :         <type>org.bluetooth.characteristic.hardware_revision_string</type>
53 :         <type>org.bluetooth.characteristic.firmware_revision_string</type>
54 :         <type>org.bluetooth.characteristic.software_revision_string</type>
55 :         <type>org.bluetooth.characteristic.system_id</type>
56 :         <type>org.bluetooth.characteristic.ieee_11073-20601_regulatory_certification_data_list</type>
57 :     </configuration>
58 : </service>

```

The Device Information service shows an example of a configuration. In this service, all the characteristics have conditional requirements (line 9, 14, 19, 24, 29, 34, 39, 44). These conditional requirements are defined in the configuration (line 48) that states that a device must support one of the list of characteristics.