Qualcomm

Qualcomm Technologies, Inc.

Qualcomm Aware™ Positioning Service JSON API

Reference Manual

80-42204-1 Rev. AE

October 15, 2024

Revision history

Revision	Date	Description
AE	October 2024	Updated the product name from Qualcomm Terrestrial Positioning Service (TPS) to Qualcomm Aware™ Positioning Service in the entire document.
AD	June 2024	■ Added a note about device MAC address in Section 3.1 *Wi-Fi access points* ■ Added a note about storing API key in Chapter 6 *Token-based registration using a license*
AC	October 2023	Updated the document to conform to current documentation standards. No technical content was changed in this revision.
AB	November 2022	Updated API implementation information.
AA	June 2022	Initial release

Contents

Tables

1 Introduction to Qualcomm Aware™ Positioning Service API

Qualcomm Aware™ Positioning Service API, formerly known as Skyhook Precision Location API, provides access to positioning information derived from analyzing positioning anchors such as Wi-Fi access points and cell IDs in known locations.

Client applications make JSON HTTPS requests to the positioning service API, which include a list of observed positioning anchors available within the range of the client device, and if available, GPS information starting from the positioning service v2.21. Based on these inputs, the positioning service API returns a calculated geographic location, and optionally returns a street address and time zone.

In addition to the derived location, the positioning service API returns location information based on the IP address of the client. While this information is not a reliable source for navigation, it can be used to provide context when more accurate methods are not available, and may be used as the best estimate of location.

API limitations

Opting for the API approach has benefits pertaining to server-side features, but is less robust than client-side implementations. Few SDK or embedded client features such as smoothing between location sources, fast time-to-fix, MAC address deduplication, power optimization, and offline location are not available with the API method.

In addition, an application or device-specific code must be created to enable the collection of Wi-Fi and cell radio scans. For recommendations in these areas to optimize the use of cloud API, contact the Qualcomm Aware™ Positioning Service team at support.tps@qti.qualcomm.com.

Scope

This document provides an overview of the positioning service JSON API and its implementation to determine the geographic location of connected devices. In addition, the document describes the API requests and responses, and provides guidelines to evaluate the API on a large scale.

Technical assistance

For assistance or clarification, write to support.tps@qti.qualcomm.com.

2 Positioning service JSON API implementation

To implement the positioning service API, the device must collect data about the radio signal environment, including Wi-Fi access point MAC addresses, cell tower IDs, and signal strengths. This data is then submitted using the positioning service API. Along with this data, if accessible, include GPS location information to improve location results even when GPS is unavailable. Starting from positioning service v2.21, GPS switching has been added to server-side functionality.

HTTP headers

For API calls, always use DNS instead of hard coded IP addresses as they are not supported by the positioning service and can cause a failure in returning location requests.

If all API calls are forwarded from a centralized server before reaching the positioning service API endpoint, the x-forwarded-for HTTP header must be used with the original public API address of the device making the request.

Consider the following example of an HTTP header:

```
Forwarded: for=192.0.2.60; proto=http; by=203.0.113.43
```

A new optional `Skyhook-Request-Token` HTTP header has been introduced in protocol v2.28. Clients can choose to include an opaque `messageID` in the `Skyhook-Request-Token` header in any type of request, and the API server will echo back the provided value as a `Skyhook-Request-Token` header in the corresponding response.

API key

An evaluation license agreement is required to obtain an API key for the positioning service. For more information on the license agreement, contact the Qualcomm Aware™ Positioning Service team at sales.tps@qti.qualcomm.com.

Unique device ID

With each unique location request, it is recommended to submit a unique device ID or `user` for the end-user device. This enables more accurate location results and impacts the redundancy of how location data is processed and weighted for optimization. If `user` is not provided, reporting and API performance may be impacted.

NOTE Individual device IDs are not stored permanently in Qualcomm's system, but are stored with a rotating hash in accordance with Qualcomm user and commercial privacy policies.

Endpoint URL

The positioning service API endpoint URL of JSON-formatted requests is https://global.skyhook.com/wps2/json/location.

2.1 HTTP authentication methods

The positioning service JSON API request can be authenticated using:

- HTTP headers via:
  - Skyhook authentication key and username
  - Skyhook web token
- HTTP URL parameters

HTTP header-based authentication

- **Skyhook auth-key and username**: Submit a request with `Skyhook-Auth-Key` header containing the API key along with optional `Skyhook-Auth-User` containing the username associated with the key.

  NOTE An evaluation license agreement is required to obtain an API key for the positioning service. For more information on license agreement, contact Qualcomm Aware positioning service team at sales.tps@qti.qualcomm.com.

- **Skyhook web token**: Skyhook supports a token-based registration and authentication model, available on a per request basis. For more information, see Token-based registration using a license or contact Qualcomm Aware positioning service team at support.tps@qti.qualcomm.com.

HTTP URL parameters-based authentication

The positioning service API JSON request body does not contain explicit authentication section. Rather, the API key and user ID are encoded into the request endpoint URL.

```
https://global.skyhookwireless.com/wps2/json/location?
key="<YOUR API KEY>&user=<DEVICE SERIAL NUM, MAC ADDRESS, OR OTHER UNIQUE ID>"
```

3 Positioning service JSON API request

Description

Requests the geographic location of the device based on observed access points, cell towers, and GPS locations using JSON.

Endpoint URL

The positioning service API endpoint URL of JSON-formatted requests is https://global.skyhook.com/wps2/json/location.

HTTP method

POST

HTTP authentication

See HTTP authentication methods.

Syntax

```
{
  "considerIp": "true",
  "hpeConfidence": 68,
  "includeLocationSource": "true",
  "includeBeaconCounts": "true",
  "wifiAccessPoints": [...],
  "cellTowers": [...],
  "gpsLocations": [...],
  "ibcnBleBeacons": [...],
  "eddyStoneBleBeacons": [...],
  "altBleBeacons": [...],
  "streetAddressLookupType": "full"
}
```

Parameters

Parameter	Values	Description
`considerIP`	▪ True (default) ▪ False	(Optional) The positioning service attempts to locate the client based on the IP address of the client.

Parameter	Values	Description
		This parameter is operational only when: ■ Set to `true`. ■ The client location cannot be determined based on provided Wi-Fi access points, cell towers, or GPS locations.
`hpeConfidence`	■ 68 (default) ■ 95	(Optional) Requests reported uncertainty (HPE) percentile change. For example, a value of 95 reports 95% uncertainty. **NOTE** The default 68% confidence is supported in positioning service v2.28 and later. For 95% confidence, set `hpeConfidence` to 95 in the location request.
`includeLocationSource`	■ True ■ False (default)	(Optional) Set to `true` to display the source of the location in the location response.
`includeBeaconCounts`	■ True ■ False (default)	(Optional) Set to `true` to display the beacon counts in the location response.
WifiAccessPoints	–	List of entries corresponding to the Wi-Fi access points scanned by the client.
cellTowers	–	List of entries corresponding to the cell towers scanned by the client. The contents of each entry depend on the type of cell tower: GSM, LTE, UMTS, WCDMA, CDMA, NB-IoT.
gpsLocations	–	List of GPS locations available to the client.
ibcnBleBeacons	–	List of iBeacon Qualcomm® Bluetooth Low Energy (BLE) beacons scanned by the client.
eddyStoneBleBeacons	–	List of Eddystone BLE beacons scanned by the client.
altBleBeacons	–	List of altBeacon BLE beacons scanned by the client.
`streetAddressLookupType`	■ Full ■ Limited	■ Set to `full` to return all address attributes. ■ Set to `limited` to return only city and state attributes. If a value is not specified in the request, the street address lookup attributes will not be returned.

3.1 Wi-Fi access points

Description

Provides a list of entries corresponding to the Wi-Fi access points scanned by the client.

NOTE This list contains the MAC addresses of the surrounding Wi-Fi access points. However, the list does not include the MAC address of the requesting device because it does not provide any value for location.

Syntax

```
{
  "macAddress": "00:0C:41:82:D8:8C",
```

```
  "ssid" : "myNetworkName",
  "signalStrength": -50,
  "age": 0,
  "frequency": 2484,
  "channel": 11,
  "connected": true
}
```

Parameters

Parameter	Values	Description
`macAddress`	–	A string of hexadecimal characters representing a 6-byte MAC address. Pairs of hexadecimal characters are separated by colon (:).
`ssid`	–	(Optional) Service Set Identifier (SSID) of the access point.
`signalStrength`	–	(Optional) Observed signal strength of the access point, in decibels (dBm).
`age`	–	(Optional) Relative age of the measurement, in milliseconds or timestamp.
`frequency`	–	(Optional) Frequency of scanned Wi-Fi access point, in MHz.
`channel`	–	(Optional) Channel number of scanned Wi-Fi access point. **NOTE** Only one frequency or channel should be provided.
`connected`	▪ True ▪ False	(Optional) Indicates whether the client device is currently connected to the Wi-Fi access point.

3.2 Scanned cell tower group

The scanned cell tower group allows the input of observed cell towers. Multiple towers and multiple types of cell towers are allowed.

Optionally, the cell tower list can include neighbors cell measurements (NMRs) in addition to serving cells. Neighbor cells can be provided with the full cell IDs (as serving cell) or partial IDs (labeled in the table as NMRs).

NOTE Partial cell ID must be accompanied by at least one serving cell.

3.2.1 GSM cell tower

Description

Provides a list of entries corresponding to the GSM cell towers scanned by the client.

Syntax

```
{
  "radioType": "gsm",
  "mobileCountryCode": 722,
  "mobileNetworkCode": 310,
  "locationAreaCode": 8022,
```

```
  "cellId": 32508,
  "neighborId": 230,
  "timingAdvance": 0,
  "signalStrength": -50,
  "age": 533,
  "channel": 40,
  "band": "GSM 480",
  "serving": true
}
```

Parameters

Parameter	Values	Description
`radioType`	–	Type of the cell tower.
`mobileCountryCode`	–	Indicates the mobile country code.
`mobileNetworkCode`	–	Indicates the mobile network code.
`locationAreaCode`	–	Indicates the local area code.
`cellId`	–	Indicates the cell ID.
`neighborId`	Range: 0 to 63	(Optional) GSM BSIC composed of a 3-bit Network Color Code (NCC) and a 3-bit Base station Color Code (BCC).
`timingAdvance`	Range: 0 to 63	(Optional) GSM time advance value (TA) when in connected mode. Integer value as per 3GPP Technical Specification 45.00 (GSM).
`signalStrength`	–	Signal strength of the cell, in dBm.
`age`	–	Relative age of the measurement, in milliseconds or timestamp.
`channel`	–	(Optional) 16-bit GSM channel number (ARFCN).
`band`	GSM 480	(Optional) GSM band.
`serving`	▪ False (default) ▪ True	(Optional) Indicates whether the client device is currently associated with the cell tower.

3.2.2 UMTS/WCDMA cell tower

Description

Provides a list of entries corresponding to the UMTS/WCDMA cell towers scanned by the client.

Syntax

```
{
  "radioType": "UMTS",
  "mobileCountryCode": 730,
  "mobileNetworkCode": 2,
  "locationAreaCode": 64300,
  "cellId": 4633170,
  "neighborId": 230,
```

```
  "signalStrength": -50,
  "signalQuality": "-13",
  "channel": 40,
  "serving": true
}
```

Parameters

Parameter	Values	Description
`radioType`	UMTS or WCDMA	Type of the cell tower.
`mobileCountryCode`	–	Indicates the mobile country code.
`mobileNetworkCode`	–	Indicates the mobile network code.
`locationAreaCode`	–	(Optional) Indicates the local area code.
`cellId`	–	UMTS UC_ID (combination of RNDid and Cell_ID) with a total of 28 bits as per 3GPP Technical Specification 25.401. Populating cell ID with 16 bits may result in incorrect tower information.
`neighborId`	Range: 0 to 511	(Optional) UMTS local cell ID (PSC).
`signalStrength`	–	Signal strength of UMTS cell, RSCP in dBm.
`signalQuality`	Range: -24 to 1	(Optional) Signal quality of UMTS cell.
`age`	–	Relative age of the measurement, in milliseconds or timestamp.
`channel`	–	(Optional) 16-bit UMTS channel number (UARFCN).
`serving`	▪ False (default) ▪ True	(Optional) Indicates whether the client device is currently associated with the cell tower.

3.2.3 LTE cell tower

Description

Provides a list of entries corresponding to the LTE cell towers scanned by the client.

Syntax

```
{
  "radioType": "lte",
  "mobileCountryCode": 730,
  "mobileNetworkCode": 2,
  "locationAreaCode": 0,
  "cellId": 20118,
  "neighborId": 230,
  "timingAdvance": 0,
  "signalStrength": -50,
  "signalQuality": "-29",
  "signalToNoiseRatio": "11",
  "channelQualityIndicator": "12",
```

```
  "age": 533,
  "channel": 40,
  "serving": true
}
```

Parameters

Parameter	Values	Description
`radioType`	–	Type of the cell tower.
`mobileCountryCode`	–	Indicates the mobile country code.
`mobileNetworkCode`	–	Indicates the mobile network code.
`locationAreaCode`	–	(Optional) Indicates the LTE tracking area code (TAC).
`cellId`	–	LTE E-CGI (EUCID); 28 bits.
`neighborId`	Range: 0 to 503	(Optional) ID of LTE local cell (PCI).
`timingAdvance`	▪ Default range: 0 to 1282 ▪ Extended range: 0 to 7690	(Optional) Length of time a signal takes to reach the base station from a mobile phone, half of round-trip time (RTT). The units are LTE 16xTs symbols as specified in Section 4.2.3 of 3GPP Technical Specification 36.213 and 36.133 (LTE).
`signalStrength`	–	(Optional) Signal strength of the cell (RSRP) received from the LTE tower, in dBm.
`signalQuality`	Range: -34 to 3	(Optional) Signal quality of the LTE cell.
`signalToNoiseRatio`	Range: -20 to 30	(Optional) Signal-to-noise ratio of the LTE cell.
`channelQualityIndicator`	Range: 0 to 15	(Optional) Indicator of LTE cell channel quality.
`age`	–	Relative age of the measurement, in milliseconds or timestamp.
`channel`	–	(Optional) 18-bit 3GPP channel number (EARFCN).
`serving`	▪ False (default) ▪ True	(Optional) Indicates whether the client device is currently associated with the cell tower.

3.2.4 CDMA cell tower

Description

Provides a list of entries corresponding to the CDMA cell towers scanned by the client.

Syntax

```
{
  "radioType": "cdma",
  "mobileNetworkCode": 2,
  "locationAreaCode": 251,
  "cellId": 65530,
```

```
  "neighborId": 230,
  "timingAdvance": 0,
  "signalStrength": -50,
  "age": 533,
  "channel": 40,
  "band": 1800,
  "serving": true
}
```

Parameters

Parameter	Values	Description
`radioType`	–	Type of the cell tower.
`mobileNetworkCode`	–	Indicates the CDMA system identifier (SID).
`locationAreaCode`	–	Indicates the CDMA network identifier (NID).
`cellId`	–	Indicates the CDMA base system identifier (BSID).
`neighborId`	Range: 0 to 511	(Optional) ID of CDMA local cell (PNCODE).
`timingAdvance`	Range: 0 to 7690	Length of time a signal takes to reach the base station from a mobile phone, half of round-trip time (RTT). The units are Ts symbols as specified in 3GPP Technical Specification 36.133 (LTE).
`signalStrength`	–	(Optional) Signal strength (pilot power) of the cell received from the tower, in dBm.
`age`	–	Relative age of the measurement, in milliseconds or timestamp.
`channel`	–	(Optional) 18-bit CDMA channel number.
`band`	–	(Optional) CDMA band; string.
`serving`	▪ False (default) ▪ True	(Optional) Indicates whether the client device is currently associated with the cell tower.

3.2.5 NB-IoT cell tower

Description

Provides a list of entries corresponding to the NB-IoT cell towers scanned by the client.

Syntax

```
{
  "radioType": "nbiot",
  "mobileCountryCode": 450,
  "mobileNetworkCode": 8,
  "locationAreaCode": 4616,
  "cellId": 7635205,
  "neighborId": 230,
  "timingAdvance": 0,
```

```
  "signalStrength": -50,
  "age": 533,
  "channel": 40,
  "serving": true
}
```

Parameters

Parameter	Values	Description
radioType	–	Type of the cell tower.
mobileCountryCode	–	Indicates the mobile country code.
mobileNetworkCode	–	Indicates the mobile network code.
locationAreaCode	–	(Optional) Indicates the NB-IoT tracking area code (TAC).
cellId	–	NB-IoT cell ID; 28 bits.
neighborId	Range: 0 to 503	(Optional) ID of NB-IoT local cell (NCID).
timingAdvance	▪ Default range: 0 to 1282 ▪ Extended range: 0 to 7690	(Optional) The length of time a signal takes to reach the base station from a mobile phone, half of round-trip time (RTT). The units are LTE 16xTs symbols as specified in Section 4.2.3 of 3GPP Technical Specification 36.213 and 36.133 (LTE).
signalStrength	–	(Optional) Signal strength of the cell (NRSRP) received from the NB-IoT tower, in dBm.
age	–	Relative age of the measurement, in milliseconds or timestamp.
channel	–	(Optional) 18-bit 3GPP channel number (EARFCN).
serving	–	(Optional) Indicates whether the client device is currently associated with the cell tower.

3.2.6 NR tower (5G, New Radio)

Description

Provides a list of entries corresponding to the 5G NR cell towers scanned by the client.

Syntax

```
{
  "radioType": "nr",
  "mobileCountryCode": 310,
  "mobileNetworkCode": 410,
  "locationAreaCode": 16777200,
  "cellId": 6871947673,
  "neighborId": 503,
  "timingAdvance": 0,
  "signalStrength": -44,
  "signalQuality": -29,
  "signalToNoiseRatio": 11,
```

```
  "channelQualityIndicator": 12,
  "signalReferenceType": "csi",
  "age": 533,
  "channel": 412,
  "serving": true
}
```

Parameters

Parameter	Values	Description
`radioType`	–	Type of the cell tower. Use a global value if this field is not populated.
`mobileCountryCode`	–	Indicates the mobile country code.
`mobileNetworkCode`	–	Indicates the mobile network code.
`locationAreaCode`	Range: 0 to 16777215	(Optional) Indicates the NR tracking area code (TAC).
`cellId`	–	5G local cell ID; 36 bits.
`neighborId`	Range: 0 to 1007	(Optional) 5G local cell ID; equivalent to PCI in LTE (range: 0 to 503).
`timingAdvance`	Range: 0 to 3846	(Optional) Length of time a signal takes to reach the base station from a mobile phone, half of round-trip time (RTT). The units Ts symbols as specified in 3GPP Technical Specification 28.212.
`signalStrength`	–	(Optional) Signal strength of the cell (CSI-RSRP) received from the NR tower, in dBm.
`signalQuality`	Range: -34 to 3	(Optional) Signal quality of the NR cell.
`signalToNoiseRatio`	Range: -20 to 30	(Optional) Signal-to-noise ratio of the NR cell.
`channelQualityIndicator`	Range: 0 to 15	(Optional) Indicator of NR cell channel quality.
`signalReferenceType`	Null (default)	(Optional) Type of signal reference. Applicable to 5G NR cells. Enum: [csi, ss]. Defaults to null and not case-sensitive.
`age`	–	Relative age of the measurement, in milliseconds or timestamp.
`channel`	–	(Optional) 24-bit 3GPP channel number (NRARFCN).
`serving`	▪ False (default) ▪ True	(Optional) Indicates whether the client device is currently associated with the cell tower.

3.2.7 Neighbor cell reporting

Neighbor cells improve cell-based accuracy and yield. The positioning service JSON API supports reporting of neighbor cells with partial identity.

The key requirements for neighbor cell reporting are as follows:

- At least one cell must be fully qualified, that is, it must include all nonoptional cell ID elements.
- Neighbor cells with partial identity should include at least `neighborId` and `channel` elements.
- When the serving cell is 4G (LTE), for example, it is common to see neighbor cells from 4G/3G/2G.
- When available, neighbor cells can be reported with full identity.

Sample request

The following example includes a fully qualified serving LTE cell with all cell identifiers specified, an LTE neighbor cell, and a UMTS neighbor cell:

```
{
  "considerIp": "false",
  "cellTowers": [
    {
      "radioType": "lte",
      "mobileCountryCode": 730,
      "mobileNetworkCode": 2,
      "locationAreaCode": 63010,
      "cellId": 20118,
      "neighborId": 230,
      "timingAdvance": 0,
      "signalStrength": -50,
      "age": 533,
      "channel": 5110,
      "serving": true
    },
    {
      "radioType": "lte",
      "neighborId": 423,
      "signalStrength": -99,
      "age": 752,
      "channel": 5110,
      "serving": false
    },
    {
      "radioType": "umts",
      "neighborId": 157,
      "signalStrength": -79,
      "age": 752,
      "channel": 4384,
      "serving": false
    }
  ]
}
```

3.2.8 GPS locations

Description

Provides a list of GPS locations available to the client.

Syntax

```
{
  "fix": 1,
  "latitude": 42.2970025,
  "longitude": -71.2333229,
  "hpe": 9,
  "altitude": 43,
  "speed": 3.5,
  "bearing": -35,
  "speedUncertainty": 2.5,
  "bearingUncertainty": 2,
  "vpe": 10,
  "age": 987,
  "constellations": [
    {
      "name": "GPS",
      "nsat": 3
    },
    {
      "name": "GLONASS",
      "nsat": 3
    },
    {
      "name": "GALILEO",
      "nsat": 3
    }
  ]
}
```

Parameters

Parameter	Values	Description
`fix`	■ 1 (default) – GPS ■ 2 – DGPS ■ 3 – PPS ■ 4 – RTK ■ 5 – FRTK ■ 6 – Estimated	Type of fix.

Parameter	Values	Description
	■ 7 – MIM ■ 8 – SM	
`latitude`	–	Calculated physical geographic location, expressed in decimal degrees using floating point values, using the WGS84 datum.
`longitude`	–	Calculated physical geographic location, expressed in decimal degrees using floating point values, using the WGS84 datum.
`hpe`	–	Horizontal Positioning Error. Estimated horizontal error of the location, in meters, with 68% confidence. **NOTE** The default 68% confidence is supported in positioning service v2.28 and later. For 95% confidence, set `hpeConfidence` to 95 in the location request.
`altitude`	–	(Optional) Altitude above WGS84, in meters.
`speed`	–	(Optional) Speed in meters/sec.
`bearing`	–	(Optional) Bearing, in degrees, from North in counterclockwise (+90 ° is West).
`speedUncertainty`	–	(Optional) Uncertainty of speed, in meters/sec with 68% confidence.
`bearingUncertainty`	–	(Optional) Uncertainty of bearing, in degrees with 68% confidence.
`vpe`	–	(Optional) Vertical Positioning Error. Estimated vertical error of the altitude, in meters, with 68% confidence.
`age`	–	(Optional) Relative age of the measurement, in milliseconds or timestamp.
`constellations`	–	List of constellation types that contributed to the fix result.
`name`	–	(Optional) Name of the constellation of GNSS satellites; possible examples are GPS, GLONASS, Galileo, BeiDou, QZSS, SBAS, UNKNOWN.
`nsat`	–	(Optional) Number of GNSS satellites, an integer value, used in a constellation to compute location.

3.3 Scanned BLE beacon group

The scanned Bluetooth low energy (BLE) beacon group allows the input of observed BLE beacons. Multiple BLE beacons and multiple types of BLE beacons are allowed as input.

3.3.1 IBeacon BLE

Description

Provides a list of iBeacon BLE beacons scanned by the client.

Syntax

```
{
  "uuid": "f7826da6-4fa2-4e98-8024-bc5b71e0893e",
  "major": 33834,
  "minor": 28699,
  "companyId": 235,
  "macAddress": "00:0C:41:A2:DF:52",
  "rssi": -80,
  "age": 10,
  "txPower": -50
}
```

Parameters

Parameter	Values	Description
`uuid`	–	Universally Unique Identifier. Comprises 32 hexadecimal digits, split into 5 groups, separated by hyphens; 16 bytes.
`major`	Range: 0 to 65535	Additional identifying number assigned to iBeacon; unsigned integer.
`minor`	Range: 0 to 65535	Additional identifying number assigned to iBeacon; unsigned integer.
`companyId`	–	(Optional) Identifier provided with iBeacon prefix; integer.
`macAddress`	–	(Optional) MAC address of the device.
`rssi`	–	(Optional) Received signal strength, in dBm. **NOTE** If this value is provided, `txPower` must also be provided for higher accuracy.
`age`	–	(Optional) Relative age of the scan, in milliseconds or timestamp.
`txPower`	Range: -128 to 127	(Optional) Transmission power of the beacon; rx_power at 1 m from BLE beacon (1 byte); signed integer.

3.3.2 Eddystone BLE

Description

Provides a list of Eddystone BLE beacons scanned by the client.

Syntax

```
{
  "namespaceId": "f7826da6bc5b71e0893e",
  "instanceId": "415931775549",
  "macAddress": "00:0C:41:A2:DF:52",
  "rssi": -84,
  "age": 10,
```

```
  "rangingData": -22
}
```

Parameters

Parameter	Values	Description
`namespaceId`	–	10-byte hexadecimal identifier; string.
`instanceId`	–	6-byte hexadecimal identifier; string.
`macAddress`	–	(Optional) MAC address of the device.
`rssi`	–	(Optional) Received signal strength, in dBm. **NOTE** If this value is provided, `txPower` must also be provided for higher accuracy.
`age`	–	(Optional) Relative age of the scan, in milliseconds or timestamp.
`rangingData`	–	(Optional) Configured maximum transmission power, in dBm, emitted by the beacon at 0 m (1 byte).

3.3.3 altBeacon BLE

Description

Provides a list of altBeacon BLE beacons scanned by the client.

Syntax

```
{
  "uuid": "123e4567-e89b-12d3-A456-426655440000",
  "major": 33834,
  "minor": 28699,
  "companyId": 235,
  "macAddress": "00:0C:41:A2:DF:52",
  "rssi": -95,
  "age": 10,
  "refRssi": -50
}
```

Parameters

Parameter	Values	Description
`uuid`	–	Universally Unique Identifier. Comprises 32 hexadecimal digits, split into 5 groups, separated by hyphens; 16 bytes.
`major`	Range: 0 to 65535	Additional identifying number assigned to altBeacon; unsigned integer.
`minor`	Range: 0 to 65535	Additional identifying number assigned to altBeacon; unsigned integer.
`companyId`	–	(Optional) Identifier provided with altBeacon prefix; integer.

Parameter	Values	Description
`macAddress`	–	(Optional) MAC address of the device.
`rssi`	–	(Optional) Received signal strength, in dBm. **NOTE** If this value is provided, `refRssi` must also be provided for higher accuracy.
`age`	–	(Optional) Relative age of the scan, in milliseconds or timestamp.
`refRssi`	Range: -128 to 0	(Optional) Average signal strength received at 1 m from the advertiser; signed 1-byte value.

4 Positioning service JSON API response

Description

Response to the positioning service JSON API request. The response includes latitude, longitude, and HPE (accuracy) determined from the request inputs.

NOTE Qualcomm Technologies, Inc. (QTI) uses multiple methods and sources for positioning IPs. Hence, HPE (accuracy) may not be returned in all responses.

Syntax

Requests that are successful return an HTTP 200 response status and a response body as follows:

```
{
    "location": {
        "lat": 42.297002,
        "lng": -71.233323
    },
    "accuracy": 18.0,
    "nap": 0,
    "ncell": 0,
    "nlac": 0,
    "nsat": 9,
    "nble": 0,
    "source": "gnss",
    "streetAddress": {
            "streetNumber": 64,
            "addressLine": "Farnsworth St",
        "neighborhood": "Beantown",
        "city": "Boston",
        "metro1": "Boston",
        "metro2": "Boston County",
        "postalCode": 02210,
        "county": "Suffolk",
        "region": "North East",
        "stateCode": "MA",
        "stateName": "Massachusetts".
        "countryCode": "US",
        "countryName": "United States"
    }
}
```

Parameters

Parameter		Description
`location`		The position of the device.
`accuracy`		Estimates HPE of the location, in meters, when `hpeConfidence` is set to 68% (default) or 95% in the API request. **NOTE** The default 68% confidence is supported in positioning service v2.28 and later. For 95% confidence, set `hpeConfidence` to 95 in the location request.
`nap`		Number of AP scans used to determine location. A value is returned only if `includeBeaconCounts` is set to `true` in the API request.
`ncell`		Number of cell scans used to determine location. A value is returned only if `includeBeaconCounts` is set to `true` in the API request.
`nlac`		Number of Location Area Codes (LACs) used to determine location. A value is returned only if `includeBeaconCounts` is set to `true` in the API request.
`nsat`		Number of satellites in the GPS location used to determine location. In case of cell fallback, this value is greater than 0. A value is returned only if `includeBeaconCounts` is set to `true` in the API request.
`nble`		Number of BLE scans used to determine location. A value is returned only if `includeBeaconCounts` is set to `true` in the API request.
`source`		Beacon type used to determine to location. A value is returned only if `includeLocationSource` is set to `true` in the API request. Values can be: ■ `unknown` ■ `cell` ■ `wifi` ■ `gnss` ■ `ble` ■ `rfpm` ■ `hybrid` (scans of multiple beacon types used)
`streetAddress`		Street address element. A value is returned only if `streetAddressLookupType` is set to either `full` or `limited` in the API request, and its location can be detected by the RGeo lookup service.
	`streetNumber`	Street number of the address. A value is returned only if `streetAddressLookupType` is set to `full` in the API request.
	`addressLine`	Contains street name, and other address lines such as apartment number, if applicable. A value is returned only if `streetAddressLookupType` is set to `full` in the API request.
	`neighborhood`	Neighborhood of the address.
	`city`	City of the address.
	`metro1`	City of the location.
	`metro2`	City/township of the location.
	`postalCode`	Postal code associated with the address.

Parameter			Description
	country		County in which the address is located.
	region		Region in which the address is located.
	stateCode		State code associated with the address.
	stateName		State in which the address is located.
	countryCode		Country code associated with the address.
venue			Venue element. A value is returned only if the beacons associated with the user's account (linked by authentication key) are relevant to the beacons in the API request that were determined for a specific indoor venue.
	name		Name of the venue.
	conf		Confidence value in detecting the venue, in decimal percentage.
	floor		Floor element. A value is returned only if the beacons in the API request were determined for a specific floor within the venue.
		name	Name of the floor.
		level	Level of the floor, in decimals, such as 0.5 floors may exist.
		conf	Confidence value in detecting the floor within the venue, in decimal percentage.

Sample response

```
{
    "location": {
        "lat": 42.297002,
        "lng": -71.233323
    },
    "accuracy": 18.0,
    "nap": 0,
    "ncell": 0,
    "nlac": 0,
    "nsat": 9,
    "nble": 0,
    "source": "gnss",
    "streetAddress": {
        "streetNumber": 64,
        "addressLine": "Farnsworth St",
        "neighborhood": "Beantown",
        "city": "Boston",
        "metro1": "Boston",
        "metro2": "Boston County",
        "postalCode": 02210,
        "county": "Suffolk",
        "region": "North East",
        "stateCode": "MA",
        "stateName": "Massachusetts".
```

```
            "countryCode": "US",
            "countryName": "United States"
        }
        *"venue": {
            "name": TestLoc
            "conf": 0.9
            "floor":  {
                "name": "Floor 1",
                "level": 1.0,
                "conf": 0.9
            }
        }
    }
```

5 Test sample positioning service JSON API request

To issue a positioning service JSON API request or to test, perform the following steps:

1. Create a file named `location_rq.json` with the following sample JSON request body:

```
{
  "considerIp": "false",
  "hpeConfidence": 95,
  "includeLocationSource": "true",
  "includeBeaconCounts": "true",
  "cellTowers": [
    {
      "radioType": "lte",
      "mobileCountryCode": 311,
      "mobileNetworkCode": 480,
      "locationAreaCode": 25828,
      "cellId": 25919502,
      "neighborId": 210,
      "timingAdvance": 1,
      "signalStrength": -50,
      "age": 533,
      "channel": 1000,
      "serving": true
    }
  ],
  "wifiAccessPoints": [
    {
      "macAddress": "E2:55:7D:91:35:50",
       "ssid": "Skyhook-1400",
      "signalStrength": -50,
      "age": 0,
      "frequency": 2437,
      "connected": true
    }
  ],
  "ibcnBleBeacons": [
    {
      "uuid": "f7826da6-4fa2-4e98-8024-bc5b71e0893e",
```

```
        "major": 58966,
        "minor": 39642,
        "companyId": 666,
        "macAddress": "00:0C:41:A2:DF:52",
        "rssi": -95,
        "age": 1000,
        "txPower": -50
      }
    ]
}
```

2. Submit the file using a cURL command-line tool with the following syntax (all in one line):

```
curl -i -v -H "Content-Type: application/json"
-d@location_rq.json https://global.skyhookwireless.com/wps2/json/location?
key="<YOUR API KEY>&user=<DEVICE SERIAL NUM, MAC ADDRESS, OR OTHER UNIQUE
ID>"
```

5.1 Positioning service JSON API status codes

The positioning service JSON API uses standard HTTP status codes to provide high-level category of error and additional details as to the nature of the error in the response JSON object. You can use these details to provide appropriate feedback for user experience and debugging.

The following table provides a list of observable HTTP status codes for the positioning service JSON API request.

Table 5-1 HTTP status codes for positioning service JSON API

Status code	Response	Description
200	Success	Valid request with location result.
400	Bad Request	Parse error due to invalid JSON.
400	Bad Request	No valid signal data.
401	Unauthorized	–
403	Forbidden	A request is made to a secured resource without authentication.
403	Forbidden	Request is throttled based on licensing terms or usage limits.
404	Not Found	A request is made to a nonexistent endpoint.
404	Not Found	Unable to determine the location with the request made.
500	Internal Server Error	–
503	Service Unavailable	–

5.2 Positioning service JSON API error responses

In the JSON interface, errors are organized in the following structure:

```
{
  "error":
```

```
  {
    "errors":[
      {
        "domain":"<class of error>",
        "reason":"<error text>",
        "message":"<short description of error>"
      }
    ],
    "code":<status code>,
    "message":"<short description of error>"
  }
}
```

Where:

- `domain` provides the classification for the associated error response. Most observed domains are `global`, `geolocation`, `usageLimits`, and `internalServer`.
- `reason` describes the particular error type.
- `message` provides the short description of the error.

Sample error responses

A sample error response for incorrectly formatted JSON is as follows:

```
{
  "error":
  {
    "errors":[
      {
        "domain":"global",
        "reason":"parseError",
        "message":"Parse Error"
      }
    ],
    "code":400,
    "message":"Parse Error"
  }
```

A sample error response for correctly formatted JSON but does not contain valid signal data is as follows:

```
{
  "error":
  {
    "errors":[
      {
        "domain":"geolocation",
        "reason":"invalidRequest",
        "message":"At least one ap/cell/gps required"
      }
```

```
    ],
    "code":400,
    "message":"At least one ap/cell/gps required "
  }
}
```

A sample error response for invalid authentication key is as follows:

```
{
  "error":
  {
    "errors":[
      {
        "domain":"usageLimits",
        "reason":"keyInvalid",
        "message":"Unauthorized"
      }
    ],
    "code":401,
    "message":"Unauthorized"
  }
}
```

A sample error response for valid requests but location results could not be determined is as follows:

```
{
  "error":
  {
    "errors":[
      {
        "domain":"geolocation",
        "reason":"notFound",
        "message":"Unable to determine location with request provided"
      }
    ],
    "code":404,
    "message":" Unable to determine location with request provided "
  }
}
```

6 Token-based registration using a license

The positioning service JSON API also uses a token-based registration model, which:

- Enables devices to generate a unique registration token.
- Provides access to the positioning service solution for a set period using the generated token.

NOTE A license agreement is required for token-based registration of the positioning service JSON API. For more information on the license agreement, contact the Qualcomm Aware™ Positioning Service team at sales.tps@qti.qualcomm.com.

The service period of the generated token is configured by the Qualcomm Aware™ Positioning Service based on the duration or term agreed upon in the license agreement. This team can also configure post term license period on a project-by-project basis. For support, contact the Qualcomm Aware™ Positioning Service team at support.tps@qti.qualcomm.com.

Token registration request

All the fields in the following token registration request are encoded into the HTTP header. There are no registration and authentication elements in the request message.

```
GET /registration-token HTTP/1.1
Host: global.skyhook.com:8080
Skyhook-Auth-Key: YOUR API KEY HERE
Skyhook-Device-Model: Sample-Device-Model
Skyhook-PID: DEVICE SERIAL #, MAC ADDRESS OR OTHER UID HERE
Skyhook-CC: 300 (Optional)
```

NOTE It is recommended not to store the API key directly on the client device.

Table 6-1 Positioning service API token registration request fields

Field	Description
`Skyhook-Auth-Key`	The API key provided for this project.
`Skyhook-Device-Model`	The device model for which this token is being generated.
`Skyhook-PID`	A unique and persistent ID for the device.
`Skyhook-CC`	(Optional) The three-digit Mobile Country Code (MCC) from a cellular service.

Token registration response

```
Skyhook-Auth-Token: SGVyZSdzIHRoYXQgdG9rZW4geW91IGFza2VkIGZvci4=
```

The response provides a registration token, and the subsequent location requests should use this token in their HTTP headers. Store the token securely on the client device for use.

An example of the location request format using the `Skyhook-Auth-Token` header is as follows:

```
curl -H "Content-Type: text/xml" -H "X-Forwarded-For: 127.0.0.1" -H
"Skyhook-Auth-Token: SGVyZSdzIHRoYXQgdG9rZW4geW91IGFza2VkIGZvci4" -d
@location_rq.xml
https://global.skyhookwireless.com/wps2/location
```

Token registration HTTP status codes

The following table provides a list of observable HTTP status codes for positioning service API token registration.

Table 6-2 HTTP status codes for positioning service API token registration

Status code	Response	Description
200	Success	Valid registration request.
200	Success	Retransmission of the token when the request is made from a device that is already registered.
501	Not Extended	Client service has expired and cannot be registered again unless post-term license terms are negotiated. In this case, the client device should not continue to send repeated registration requests. Instead, it is recommended to implement a registration retry method on each restart of the device.
511	Network Authentication Required	Token expired or not provided. The device should send a registration message to receive a valid token before requesting location information.

7 Positioning service JSON API for IoT (Echo)

Echo is a key feature developed specifically for IoT as a mechanism to forward location information of devices in real time directly from the positioning server to the OEM web server. This feature allows mapping and tracing applications to display the location of a device without the need for additional round-trip communication to the device.

Echo implementation

NOTE Contact the Qualcomm Aware™ Positioning Service team at support.tps@qti.qualcomm.com to set up and configure Echo.

The client application provides the URL to the remote server. The service uses HTTP authentication, a username and password generated by Qualcomm, for securing the traffic. After the URL is confirmed, the username and password are relayed to the client application. The URL and credentials are provided to the OEM via a secure means of communication of their choice.

Sample JSON POST message

```
{
   "location":{
      "lat":40.071248,
      "lon":-75.465249
   },
   "accuracy":5,                // comments
   "source":"WIFI",             // comments
   "status":"SUCCESS",             // comments
   "partner_id":”XXXXX”,         //  comments
   "device_id":"4aa472471972"   //  comments
}
```

Parameters

Parameter	Values	Description
`accuracy`	■ 68% (default) ■ 95%	Estimated horizontal positioning error (HPE) of the location, in meters when `hpeConfidence` is set to the value mentioned in the request. **NOTE** The default 68% confidence is supported in positioning service v2.28 and later. For 95% confidence, set `hpeConfidence` to 95 in the location request.
`source`	■ Unknown ■ cell ■ wifi ■ gnss ■ ble ■ rfpm ■ hybrid (scans of multiple beacon types used)	The beacon type that was used to determine the location. This value is populated only when `includeLocationSource = true` in the request.
`status`	■ 200 (successful) ■ 400 (failed)	Status of the response.
`partner_id`	–	API-Advanced Encryption Standard (AES) key created for the project.
`device_id`	–	End-user device identification number provided in the location request.

8 Guidelines to evaluate positioning service JSON API

Before testing the positioning service JSON API on a large scale, it is necessary to evaluate the accuracy and yield of Wi-Fi and cellular ID-based services correctly.

8.1 Guidelines to perform field test and collect data

The best approach for testing the positioning service JSON API is to conduct testing in the field where real-world errors can be determined. Qualcomm has tools and applications to help with field testing, data collection, and analysis.

Consider the following guidelines and recommendations to achieve consistent and reliable test results:

- Test in various morphologies such as dense urban, suburban, deep indoor, and indoor.
  - A mix of 70% indoors and 30% outdoors is recommended to account for devices being indoors most of the day.
- A statistically significant sample size is required to get an accurate picture of the location services.
  - At a minimum, take several thousand samples in a given city. If this is not achievable due to limitations in resources or individual city access, contact the Qualcomm Aware™ Positioning Service team to determine a fair sample size and location criteria for the specific test area.
- When choosing test points, ensure that an accurate ground truth location can be determined and recorded.
  - Use street corners or other recognizable landmarks for testing. Online satellite and vector map services can help to determine ground truth.
- After selecting the test point, ensure that the device is not in motion and record the following information:
  - Time
  - Real location (ground truth), where the test sample is taken via a mapping service
  - Result of location system
- After recording all the information of the test location, compare each location result to the ground truth to determine individual location errors.
  - Compile these individual location errors into a Cumulative Distribution Function (CDF). See Guidelines to analyze test results for more information.

8.2 Guidelines to use GNSS/GPS for ground truth

With recent improvements in GNSS and receiver technologies, GNSS/GPS is often used to determine accurate ground truth. However, using GNSS/GPS for ground truth is not advised in many cases as it can be inaccurate in urban areas due to the urban-canyon effect of multipath. In such cases, it is possible that the GNSS receiver suggests that its accuracy is good, but in reality it is not.

Performing field tests particularly on a large scale may not always be possible due to constraints in resources, time, and cost. However, high-quality GNSS reference points may mostly be available in areas where network location is least valuable such as outdoors in areas with clear view of the sky.

When determining the quality of a location system using GNSS/GPS, consider the following guidelines and recommendations to achieve consistent and reliable test results. These results are used for replaying RF signal environment scans and comparing them to GNSS as a proxy for accurate ground truth.

- Select a bounding box for the test area with enough device traffic and assumed access point/cell coverage to remove signal scan and data processing variability.
  - This bounding box can be in metropolitan areas or suburban environments where network-based locations are most valuable to a system but also have the potential for accurate GNSS reference locations.
- Ensure that the GNSS samples used indicate that the location result is highly accurate.
  - Use GNSS samples where recommended GNSS uncertainty is $\leq$ 10 m and the number of satellites reporting is $\geq$ 8.
  - Remove GNSS samples that indicate that the device is likely in motion. This includes GNSS attributes such as speed < 0.5 m/sec.
- Ensure that there is minimal delta time between the GNSS fix used and the Wi-Fi/network information replayed for that sample.
  - For example, the recommended delta time between the GNSS fix and the corresponding Wi-Fi scan is $\leq$ 1.
- Ensure that the data use for replays is as recent as possible.
  - In general, Wi-Fi access points are moved and reused, and cell identifiers can be rotated or changed by operators. These updated locations are reflected in the most recent database of location system providers.
  - Using the most recent possible data reduces the risk that access points and cell IDs may have either moved or changed during the time between data collection and the replayed requests.

8.3 Guidelines to analyze test results

Comparing results from different location systems can be difficult due to varying algorithms and trade-offs between yield and accuracy. Here, yield is the percentage of time a location is returned and accuracy is the distance to ground truth.

A few common industry standard methods for improving accuracy metrics over yield are as follows:

- Do not include an error value for individual location requests that produced no or poor results.
- Do not include an error value for individual location requests for single access point locations or low number of access points.

Qualcomm implements algorithms and fallback logic to maximize yield and provides the best location solution suitable for all circumstances. Compared to returning a failed location in most cases, this solution provides the best possible location of any sort. OEMs can use the best available location result depending on the use case, and if necessary, filter out locations that are above a certain error estimation value.

Though it is difficult to compare different systems, it is a necessary step to choose a technology. Consider the following guidelines and recommendations to compare systems as equally possible:

- Select a bounding box for the test area with enough device traffic and assumed access point/cell coverage to remove signal scan and data processing variability.
  - This bounding box can be in metropolitan areas or suburban environments where network-based locations are most valuable to a system but also have the potential for accurate GNSS reference locations.
- Ensure that the request or test sample size of a given region is large enough to evaluate the systems appropriately.
  - Though it depends on the available data, time, or bandwidth of the tester, it is recommended to take several thousand samples for each region evaluated.
  - Sample counts should accompany test results to understand the scale at which the solutions were tested in each region.
- Ensure to evaluate both accuracy and yield in a given region.
- To analyze accuracy, calculate the distance to ground truth (error) in each individual location response.
  - Use these values to calculate and plot a Cumulative Distribution Function (CDF) in increments of 10% across the entire set for each test region or in aggregate.
  - For example, an accuracy value of X meters at the 50th percentile indicates 50% of locations in the set are either equal to or less than error X.
  - Include the 67% as this represents 1-sigma or 1 standard deviation from the statistical mean.
- Standardize an error value for each failed location result.
  - It is important to not skew accuracy data in higher percentiles for locations that can only be returned for a given system.
  - Set a large distance for this error value, for example, 100,000 m or greater.

References

Acronyms and terms

Acronym or term	Definition
AES	Advanced encryption standard
BLE	Bluetooth low energy
CDF	Cumulative distribution function
CDMA	Code division multiple access
GPS	Global positioning system
GNSS	Global navigation satellite system
GSM	Global system for mobile
HPE	Horizontal position error
LTE	Long-term evolution
NB-IoT	Narrowband Internet of things
NR	New radio
RTT	Round-trip time
SSID	Service set identifier
UMTS	Universal mobile telecommunications service
WCDMA	Wideband code division multiple access

LEGAL INFORMATION

Your access to and use of this material, along with any documents, software, specifications, reference board files, drawings, diagnostics and other information contained herein (collectively this "Material"), is subject to your (including the corporation or other legal entity you represent, collectively "You" or "Your") acceptance of the terms and conditions ("Terms of Use") set forth below. If You do not agree to these Terms of Use, you may not use this Material and shall immediately destroy any copy thereof.

1) Legal Notice.

This Material is being made available to You solely for Your internal use with those products and service offerings of Qualcomm Technologies, Inc. ("**Qualcomm Technologies**"), its affiliates and/or licensors described in this Material, and shall not be used for any other purposes. If this Material is marked as "**Qualcomm Internal Use Only**", no license is granted to You herein, and You must immediately (a) destroy or return this Material to Qualcomm Technologies, and (b) report Your receipt of this Material to qualcomm.support@qti.qualcomm.com. This Material may not be altered, edited, or modified in any way without Qualcomm Technologies' prior written approval, nor may it be used for any machine learning or artificial intelligence development purpose which results, whether directly or indirectly, in the creation or development of an automated device, program, tool, algorithm, process, methodology, product and/or other output. Unauthorized use or disclosure of this Material or the information contained herein is strictly prohibited, and You agree to indemnify Qualcomm Technologies, its affiliates and licensors for any damages or losses suffered by Qualcomm Technologies, its affiliates and/or licensors for any such unauthorized uses or disclosures of this Material, in whole or part.

Qualcomm Technologies, its affiliates and/or licensors retain all rights and ownership in and to this Material. No license to any trademark, patent, copyright, mask work protection right or any other intellectual property right is either granted or implied by this Material or any information disclosed herein, including, but not limited to, any license to make, use, import or sell any product, service or technology offering embodying any of the information in this Material.

THIS MATERIAL IS BEING PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESSED, IMPLIED, STATUTORY OR OTHERWISE. TO THE MAXIMUM EXTENT PERMITTED BY LAW, QUALCOMM TECHNOLOGIES, ITS AFFILIATES AND/OR LICENSORS SPECIFICALLY DISCLAIM ALL WARRANTIES OF TITLE, MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, SATISFACTORY QUALITY, COMPLETENESS OR ACCURACY, AND ALL WARRANTIES ARISING OUT OF TRADE USAGE OR OUT OF A COURSE OF DEALING OR COURSE OF PERFORMANCE. MOREOVER, NEITHER QUALCOMM TECHNOLOGIES, NOR ANY OF ITS AFFILIATES AND/OR LICENSORS, SHALL BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY EXPENSES, LOSSES, USE, OR ACTIONS HOWSOEVER INCURRED OR UNDERTAKEN BY YOU IN RELIANCE ON THIS MATERIAL.

Certain product kits, tools and other items referenced in this Material may require You to accept additional terms and conditions before accessing or using those items.

Technical data specified in this Material may be subject to U.S. and other applicable export control laws. Transmission contrary to U.S. and any other applicable law is strictly prohibited.

Nothing in this Material is an offer to sell any of the components or devices referenced herein.

This Material is subject to change without further notification.

In the event of a conflict between these Terms of Use and the *Website Terms of Use* on www.qualcomm.com, the *Qualcomm Privacy Policy* referenced on www.qualcomm.com, or other legal statements or notices found on prior pages of the Material, these Terms of Use will control. In the event of a conflict between these Terms of Use and any other agreement (written or click-through, including, without limitation any non-disclosure agreement) executed by You and Qualcomm Technologies or a Qualcomm Technologies affiliate and/or licensor with respect to Your access to and use of this Material, the other agreement will control.

These Terms of Use shall be governed by and construed and enforced in accordance with the laws of the State of California, excluding the U.N. Convention on International Sale of Goods, without regard to conflict of laws principles. Any dispute, claim or controversy arising out of or relating to these Terms of Use, or the breach or validity hereof, shall be adjudicated only by a court of competent jurisdiction in the county of San Diego, State of California, and You hereby consent to the personal jurisdiction of such courts for that purpose.

2) Trademark and Product Attribution Statements.

Qualcomm is a trademark or registered trademark of Qualcomm Incorporated. Arm is a registered trademark of Arm Limited (or its subsidiaries) in the U.S. and/or elsewhere. The Bluetooth® word mark is a registered trademark owned by Bluetooth SIG, Inc. Other product and brand names referenced in this Material may be trademarks or registered trademarks of their respective owners.

Snapdragon and Qualcomm branded products referenced in this Material are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm patented technologies are licensed by Qualcomm Incorporated.